Autonomous Underwater Data Muling using Wireless Optical Communication and Agile AUV Control

by

Marek Wojciech Doniec

Vordiplom, Universität Karlsruhe (TU) (2004)
M.S., Yale University (2007)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2013

Author ....
Department of Electrical Engineering and Computer Science
December 12, 2012

Certified by ....
Daniela Rus
Professor
Thesis Supervisor

Accepted by ....
Leslie A. Kolodziejski
Chairman, Department Committee on Graduate Students

Autonomous Underwater Data Muling using Wireless Optical Communication and Agile AUV Control

by

Marek Wojciech Doniec

Submitted to the Department of Electrical Engineering and Computer Science
on December 12, 2012, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Underwater exploration and surveillance currently relies on subsea cables and tethers to relay data back to the user. The cause for this is that water heavily absorbs most electromagnetic signals, preventing effective radio communication over large distances, and that underwater communication with acoustic signals affords only bit rates on the the order of Kilobits per second.

In this thesis we present a novel design and implementation for an underwater data muling system. This system allows for automatic collection of underwater datasets without the need to physically connect to or move the sensors by using mobile robots to travel to the sensors and download the data using wireless optical communication to bring it back to the base station.

The system consists of two parts. The first part is a modular and adaptive robot for underwater locomotion in six degrees of freedom. We present a hardware design as well as control algorithms to allow for in-situ deployment without the need for manual configuration of the parameter space. To achieve this we designed a highly parameterizable controller and methods and algorithms for automatically estimating all parameters of this controller. The second part of the data mulling system is a novel high-bandwidth optical underwater communication device. This device allows for transfer of high-fidelity data, such as high-definition video and audio, images, and sensor logs. Finally we present algorithms to control the robots path in order to maximize data rates as it communicates with a sensor while using only the signal strength as a measurement.

All components and algorithms of the system have been implemented and tested in the real world to demonstrate the validity of our claims.

Thesis Supervisor: Daniela Rus

Title: Professor

Acknowledgments

First and foremost, I would like to thank my thesis advisor, Daniela Rus, who has provided me with the perfect environment to follow my passions, build my knowledge base and refine my skills. She was at all times a source of inspiration, encouragement, and feedback, as well as a tireless supporter and advocate of my work. Without her, this thesis and the rich and exciting path I have taken at MIT would have not been possible. Thank you, Daniela!

In addition to Daniela, there are several other faculty to whom I am thankful. I would like to thank John Leonard, Russ Tedrake and Mandar Chitre for serving as my thesis committee. I have had many interesting and fruitful discussions with them and their feedback and support have been invaluable to my research and to this thesis. I am very grateful for their support. Mandar has also been a long time collaborator and a great host to my many visits to Singapore, which I have truly enjoyed! John has been a continued advocate of mine and I have always felt welcome in his office. He was a much appreciated and never-ending source of pointers to papers and ways to improve my work. Finally, Russ, together with Gerald Sussman, served as my research qualification committee and provided me with feedback and encouragement early on, which I truly appreciate. I am further very grateful to Randall Davis who has provided support far beyond the duties of an academic advisor. Finally, I would like to thank the many more MIT faculty that I have interacted with and that have made my experience here rich and exciting.

I would like to thank Shigeo Hirose and Edwardo Fukushima for welcoming me into their lab in Tokyo in December 2009. Your robots are a true inspiration! I would like to thank Oussama Khatib for a great visit to Stanford. I would like to thank Rodney Brooks for encouraging me to follow my dream to build cool robots.

I owe many thanks to my fellow lab-mates at the Distributed Robotics Lab and the Computer Science and Artificial Intelligence Lab, many of whom have become close friends. Iuliu Vasilescu and Carrick Detweiler have been my mentors in the art of underwater robot building. I have had many great discussions with and have learned many skills from Elizabeth Basha, Kyle Gilpin, Brian Julian, Mac Schwager, and Marsette Vona. Anqi Xu and Michael Angermann have helped me during the final stages of my thesis and their help is greatly appreciated. Stephanie Gil and Sejoon Lim were my roommates in Singapore and have been incredible companions during this journey and beyond. I would like to thank Ron Wiken and The Infrastructure Group (TIG) for making CSAIL as efficient as it is. I would like to thank our lab assistants Kathy Bates, Marcia Davidson, and Mieke Moran. Finally, I would like to thank Alyona Michel for her support and for proofreading parts of this document. I am extremely grateful to these and many more people who have made my experience at MIT as amazing as it was!

Overall, I have been very fortunate to make many new friends during my PhD, all of whom have been an invaluable enrichment to my life. I am thankful for their friendship. It is impossible to list all of them, but there are four that have been a large constant during my life in Cambridge: Andrew Correa, David Glenn, Sertac Karamann, and Matt Walter.

Finally, I would like to thank my parents, Maria and Marek, and my brother, Piotr. My parents encouraged me to dream big and taught me the tools to turn those dreams into reality. Piotr is my best friend, and I admire him in many ways.

Support for this research has been provided in part by the MURI Antidote grant N00014-09-1-1031 and NSF grants IIS-1133224 and IIS-1117178. I am grateful for this support.

The sea is everything. It covers seven tenths of the terrestrial globe. Its breath is pure and healthy. It is an immense desert, where man is never lonely, for he feels life stirring on all sides. The sea is only the embodiment of a supernatural and wonderful existence. It is nothing but love and emotion; it is the Living Infinite.

CAPTAIN NEMO,
JULES VERNE'S 20,000 LEAGUES UNDER THE SEA

Contents

List of Figures

Chapter 1

Introduction

1.1 Vision

More than 70% of our planet is covered by water. It is widely believed that the underwater world holds ideas and resources that will fuel much of the next generation of science and business. This is evidenced by the hundreds of millions of dollars in funding dedicated towards the development of underwater observatories [30]. The concept of an underwater observatory, suggested no later than 1870 by Jules Verne [122], has recently become a reality with cabled networks as NEPUTE [15], VENUS [16], and MARS [66]. All of these systems provide researchers with the ability to install and connect with long term instrumentation at fixed locations in the water. However, currently these systems are limited in scope to deployments at or near designated instrumentation sites.

We envision underwater observatories in which robots and in-situ static and mobile sensors collect information about the underwater environment and deliver this information to remote users with little or no spatial constraints, much like a mobile web-cam delivers remote data to users on the ground. Such observatories would act mostly without human intervention, allowing for easy and safe collection of data. They could be deployed to monitor corral reefs like the Great Barrier Reef, deep ocean environments, such as the Marina Trench, or surveil marine areas important to national security, such as coastal lines.

However, today's underwater operations are fraught with difficulty due to the absence of an easy way to collect and monitor data at high data rates on the order of Megabits per second and with comprehensive area-coverage. Although underwater sensors exist, they lack much of the functionality needed in such an underwater observatory:

- The position of sensors is fixed resulting in a limited spatial scope for data collection with individual sensors;
- Collecting the data is subject to very long delays which is prohibitive to live data streams such as video feeds;
- Individual sensors are unable to perform operations that require cooperation, such as tracking relative movement and locating events.

What is required is a low-cost, versatile, high-quality, and easily deployable platform for underwater observatories that will

- provide agile mobile nodes that can be controlled in 6 degrees of freedom (DOF), much like a webcam that can be repositioned remotely;
- provide and interface with sensors that collect high quality data, such as real color images and video;
- communicate data from the observatory to the user at a fast data rate and with minimal latency;
- communicate commands from the user to the observatory;
- automate data collection and scale-up in time and space, and
- be easy to use.

It is our goal is to develop the technologies that enable users to set up and interact with underwater observatories. We believe that an effective method for large-scale data transfers (which are required when uploading the data collected by an underwater sensor network) is optical data muling. In optical data muling, a robot equipped with an optical modem visits each node of the sensor network and uploads its data while hovering within optical communication range. Previous work has described an underwater sensor network system capable of uni-directional optical data muling using GPS to find the sensor node [116]. However, the data transfer performance of the entire system was limited by the optical modem's maximum data rate of 320 Kbps.

In this thesis, we propose developing a new generation of an underwater data collection system, whose performance is a significant improvement over the state of the art, and several applications of underwater communication. This system is also based on a data muling approach and consists of two major components. The first is an optical communications platform whose performance in both range and data rates exceeds that of state-of-the-art systems. The second is a highly maneuverable, versatile, and easily deployable underwater robot capable of mobile data collection and autonomous travel to a sensor site for optical data download.

1.2 Challenges

There are many challenges which must be overcome in order to achieve underwater data muling. The most important challenges include communication, estimation, and controls.

When trying to achieve underwater data muling, the first problem is a *communication* problem. In order to successfully data mule, communication links with high data transfer rates of 1 Mbps and above must be established. Such data rates are necessary if we want to transfer high fidelity data, such as pictures, audio and video recordings. Attaining data rates of 1 Mbps and above is difficult even at short ranges below 50 m which are required in order to transfer data from a sensor node to the robot performing data muling. Conventional high-speed wireless communication relies on radio frequencies and does not work underwater at distances greater than a few meters. This is due to the strong absorption of electromagnetic radiation by water across the majority of the frequency spectrum. Currently, the de facto standard for wireless underwater communication is the use of the acoustic channel. However, acoustic communication allows for only data transfer rates on the order of a few Kilobits per second that do not permit the download of large data files [70].

The second problem is an *estimation* problem. In order to establish a high-speed communication link, we need to come within range of the sensor node from which we intend to data mule. This requires localization of the underwater sensor node relative to the data muling robot. When distances between the robot and the nodes are large, and their locations are unknown, positioning the data muling robot within optical communication range is challenging. However, this is usually the case in large open environments, such as the ocean, where the deployment of large-scale locations networks still remains a major challenge. Rather than focusing on the development and use of such a network, we instead try to solve the problem with equipment provided solely via the sensor node and robot. Finally, once we have localized the sensor node, we need to continuously update our position estimates in order to maintain a position that is within communication range.

The third problem is a *controls* problem. The data muling robot has to approach the sensor node and subsequently maintain position within communication range. This is especially challenging as optical communication requires a direct line of sight between the transmitter and receiver in most cases. In addition, the robot is required to potentially transport sensors to and from the field of deployment. This wide range of tasks can be achieved with a modular robot whose thrusters can be easily repositioned, added, or removed from the system to ensure that the system can deliver and position the payload with the precision required by the task. Such a modular robot requires a robot configuration and control system that is very adaptive.

In summary, the design of an autonomous and efficient data muling system poses many problems, with this thesis addressing the following:

- Efficient control of a hovering underwater robot with n degrees of freedom (n thrusters) on a curve in the Special Euclidean Group SE^3. The robot is reconfigurable in the field (thrusters can be change location, be added or removed).

This makes it difficult to compute a valid vehicle dynamics model. Thus, emphasis is placed on controls without the existence of a vehicle dynamics model.

- Design of a device and algorithms for high bandwidth, point-to-point optical communication. This device forms the communication link for our system.
- Integration of robot control and optical communication to allow for autonomous and efficient data muling.
- Localization of the robot based on the optical link signal strength if an external localization is not available or too coarse in space or time.

1.3 Our Approach

In order to overcome the challenges presented above, we needed to focus on the development of hardware as well as algorithms.

To address the problem of high-data-rate underwater *communication*, we developed an underwater wireless communication device called AquaOptical, which uses visible light to transmit data at rates of up to 8 Mbps. We have investigated ways to maintain these data rates while maximizing our communication range. In particular, we have focused on minimizing the requirement for pointing of the transmitter and receiver at each other. This simplifies establishing and maintaining communication links. Further, we focused on a device that is practical, i.e., that can be constructed and deployed at a reasonable cost. This requires a careful choice of components, attention to form factors and power consumption, and the development of low-complexity communication algorithms that can be implemented within the hardware constraints of the system. In order to allow easy experimentation with different communication codes and protocols, we developed the optical modem to work as a "software defined radio" [40, 82]. This also allows the system to adapt to the environment by easily changing its coding.

To address the *estimation* problem of locating the sensor, we focused on two aspects. First, we developed a method for reliable measurement of the optical signal strength arriving at the receiver and a model to predict this signal strength. We showed that if we are within optical communication range, then the signal strength measurements can be used to estimate the location of the sensor node. We use two methods to integrate these measurements into a location estimate: (1) gradient descent and (2) a particle filter. The latter can be integrated with the signal strength model to significantly improve estimation performance. The second aspect of our solution focuses on combining the estimation (based on optical signal strength) with acoustic ranging. The integration of an acoustic signal is necessary not only because it extends our location estimation capability beyond the short range of optical communication, it can even improve the overall estimation performance. Also, it allows for high-level message passing, such as turning the optical communication device on and off and thus saving power on the sensor.

We tackle the *controls* problem through the development of an underwater robot system, whose dynamic properties can be adjusted in real time, and the algorithms required to control this system. Specifically, we developed modular underwater robots whose thruster configuration can be adjusted on the fly, by adding, removing, or repositioning thrusters. We focus on the key challenge to develop a robot control system capable of adapting to changing robot dynamics in situ, without human reprogramming. In order to do so, we develop an algorithm for controlling a robot with arbitrary but known thruster configurations. The algorithm allows for full control of the robot's position and attitude in the special euclidean group SE^3, provided that this is possible with the current thruster configuration. SE^3 is the group of all three translational plus all three rotational degrees of freedom. We then present an algorithm for in-situ estimation of previously unknown thruster configurations of modular underwater robots.

Finally, we combine these three aspects of our solution to create an autonomous underwater data muling system that can retrieve data from sensor nodes of previously unknown location. The final algorithm has three phases. In the first phase, acoustic communication is used to bring the data muling robot within some close range of the desired sensor where it can detect the optical signal. In the second phase, the robot

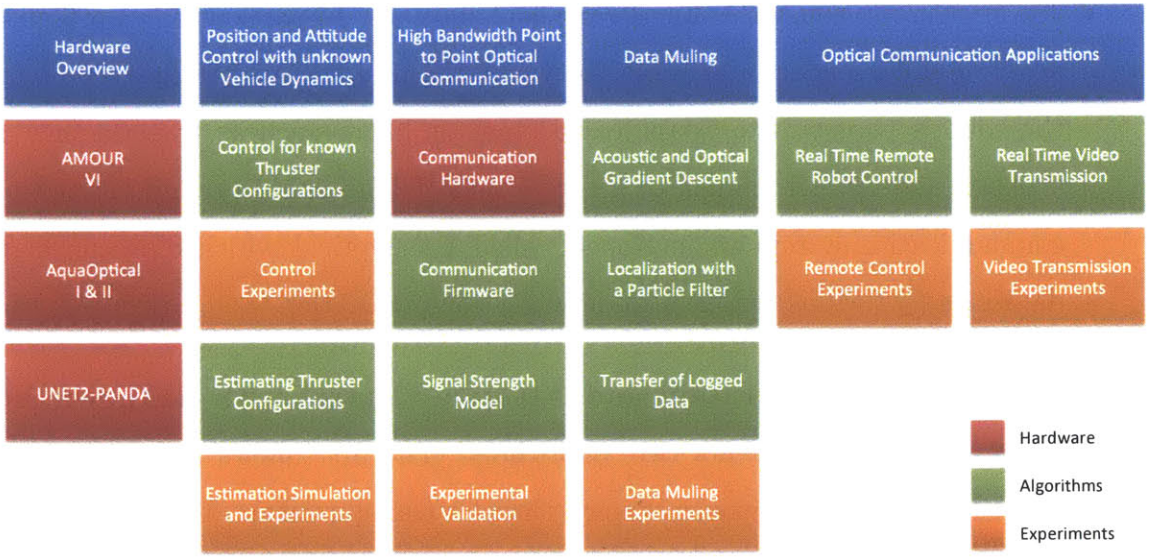

Hardware Overview
Position and Attitude Control with unknown Vehicle Dynamics
High Bandwidth Point to Point Optical Communication
Data Muling
Optical Communication Applications
AMOUR VI
Control for known Thruster Configurations
Communication Hardware
Acoustic and Optical Gradient Descent
Real Time Remote Robot Control
Real Time Video Transmission
AquaOptical I & II
Control Experiments
Communication Firmware
Localization with a Particle Filter
Remote Control Experiments
Video Transmission Experiments
UNET2-PANDA
Estimating Thruster Configurations
Signal Strength Model
Transfer of Logged Data
Estimation Simulation and Experiments
Experimental Validation
Data Muling Experiments
🟥 Hardware
🟩 Algorithms
🟧 Experiments

Figure 1-1: Thesis tree.

does a local search using the optical signal strength to precisely locate the sensor and position itself within communication range. In the third phase the robot uses optical communication to collect the data from the sensor. In practice, phases two and three overlap once the signal strength becomes strong enough to transmit data.

1.4 Thesis Contributions

In this document, we aim to examine the following thesis:

Underwater Data Retrieval can be effectively achieved using Data Muling with Combined Underwater Optical and Acoustic Communication.

To do so, we focus on the development of high-data-rate underwater optical communication systems and their integration with existing acoustic communication systems. We show that together with appropriate communication, estimation, and control algorithms, they can be used to achieve autonomous underwater data retrieval at unprecedented data rates.

The main contributions of this thesis are illustrated in Figure 1-1. They can be grouped into different parts:

- The first group comprises **hardware** contributions. In particular, we present a novel modular underwater robot for which the user can arbitrarily reconfigure the robot's thrusters to adapt to the task at hand (such as transporting an acoustic and optical modem or a sensor node). We further present the optical communication system we developed, called AquaOptical, which allows for high-data-rate ($\geq$1 Mbps) wireless underwater communication.

- The second group, **Position and Attitude Control with unknown Vehicle Dynamics** presents two algorithms. The first allows for control of a robot in $\mathcal{SE}^3$ with previously known thruster configurations. The second algorithm estimates these thruster configurations. Together these algorithms allow the user to assemble the robot, autonomously estimate its control parameters (thruster configuration), and then control it in all three translational and all three rotational degrees of freedom.

- The next group, **High Bandwidth Point-to-Point Optical Communication**, presents the algorithms used by AquaOptical. We present the low-level codes used, as well as algorithms to detect and decode data packets. Further, we present a method for reliable estimation of the received signal strength and a model for predicting this signal strength.

- The fourth group is **Data Muling**. It presents algorithms that combine the previous contributions to achieve data muling. In particular, we present an algorithm that locates the sensor node using gradient descent on the acoustic and optical signal. The second algorithm presented in this group improved localization of the sensor node through the use of a particle filter. Finally, we present algorithms for robust transfer of data from the sensor node to the robot, while minimizing power consumption on the sensor node.

- In the final group, **Optical Communication Applications**, we present two additional uses of our underwater system. The first is real-time remote robot control, in which we use AquaOptical to remotely control our underwater robot.

The second is real-time video transmission that allows us to close the loop between the user and the robot.

All algorithms and systems presented in the thesis are verified experimentally. In summary, this thesis contributes the following:

- The concept of, and the algorithms to implement, data muling using combined acoustic and optical communication.
- Design of a device and algorithms for high-bandwidth point-to-point optical communication.
- An end-to-end model of signal strength in underwater optical communication, along with an experimental evaluation of the model using the AquaOptical II modems.
- Design of a modular and reconfigurable thruster system for an underwater robot which allows the robot to adapt to different use cases.
- An algorithm for efficient control of a reconfigurable hovering underwater robot with n degrees of freedom (n thrusters) on a curve in the Special Euclidean Group SE^3.
- An algorithm for estimating thruster configurations of a reconfigurable hovering underwater robot with n degrees of freedom (n thrusters).

1.5 Thesis Outline

The remainder of this document is organized as follows. In Chapter 2, we present existing research related to this thesis. In particular, we present related works in the fields of underwater optical communications, localization in the context of data muling, and vehicles controls. In Chapter 3 we present the underwater robot AMOUR VI and AquaOptical II communication hardware developed in the context of this thesis as well as the underwater sensor network PANDA, which has been integrated into our data muling system. Chapter 4 presents the control algorithms developed to achieve six degree of freedom control for arbitrary configurations of AMOUR VI. Chapter 5 presents a model for estimating optical signal strength for underwater transmission and the model is experimentally validated using AquaOptical II. In Chapter 6 we present the algorithms developed for estimating a single nodes location based on acoustic ranging and optical signal strength measurements. We further present experimental data muling results. Chapters 7 and 8 present additional use cases of AquaOptical II for data transfer and remote robot operation. Specifically, Chapter 7 presents a system for wireless real-time remote control of AMOUR VI and Chapter 8 presents algorithms for wireless real-time transmission of video using, both using AquaOptical II. Finally, Chapter 9 discusses the lessons learned from developing our system and presents an outlook into possible future research directions related to this thesis.

Chapter 2

Related Work

This Chapter highlights prior works related to the algorithms and systems presented in this thesis. First we address prior art in underwater wireless communication in Section 2.1. Section 2.2 focuses on works related to wireless underwater video streaming. existing research related to data muling and in particular the location estimation problem is reviewed in Section 2.3. Finally, in Section 2.4 we give an overview of AUV control literature.

2.1 Underwater Optical Communication

Underwater optical communication has received a great deal of interest in the recent years despite still being a very young field of research. Before visible light was used for underwater communications it was in use for communications through air as reviewed by Smyth *et al.* [104]. The use of light emitting diodes (LEDs) for wireless over the air communication of audio signals was reported in by Pang *et al.* [88].

Bales *et al.* proposed the use of optical communication underwater and computed that data rates of 10 Mbps should be possible over distances of 20 m with off the shelf components [14]. An early stage underwater communication system reported by Tsuchida *et al.* uses optical communication to transmit measurement data from non-tethered crayfish in a laboratory tank [108]. This was quickly followed by early underwater optical digital communication systems. Tivey *et al.* , Schill *et* *al.* and Vasilescu *et al.* proposed optical communication for swarms of underwater vehicles [107, 96, 119]. Dunabin *et al.* proposed the use of optical communication for data muling where an underwater vehicle approaches a sensor and downloads data using a short range but high speed optical link [48]. In this thesis we expand on this idea and provide a system that is capable of data muling in open water environments at scales of kilometers. Optical remote control of underwater vehicles is demonstrated by Farr *et al.* at large ocean depths where no ambient light is present [55]. Our system allows remote control of underwater vehicles in environments with ambient light present.

Many of the systems implemented today have not been analyzed methodically to determine factors that influence their transmission ranges. Often ranges were measured during field experiments and the best range was reported. Some in-depth characterizations of transmission ranges have been performed in laboratories with large modem setups, but they are not field-ready. Chancey *et al.* measured the performance of their 10 Mbps system with the transmitter and receiver aimed at each other in a water tank at a distance of 4.6 m [22]. Hanson *et al.* demonstrated a laser based system capable of data rates of 1 Gbps over a 2 m path in a tank with 36 dB extinction (absorbents were added to the water) [65]. Their system's performance is impressive, but it is based on a bulky, fragile, and power inefficient laser setup that is not practical in the context of a small underwater robot.

Jaruwatanadilok *et al.* proposed a channel model for optical underwater communication that uses vector radiative transfer theory [68]. Smart *et al.* explored the theory of optical transmission in water and suggested possible optical modem designs [103]. Giles *et al.* gave a model for computing achievable transmission range for a given setup when pointing both devices at each other. They considered both light source power and detector sensitivity as well as extinction through the water [61]. Gabriel *et al.* proposed a model for simulating channel impulse response [58]. They used Monte Carlo simulations of their model to show that channel time dispersion is negligible in most practical cases, even for data rates of 1 Gbps over a distance of 50 m with water turbidity equivalent to 7.5 e-folding lengths over 50 m. Kedar *et al.* proposed a model for non-line-of-sight (NLOS) underwater optical communication [69]. In this thesis we present an optical signal strength model based on physical analysis of all components in the communication path.

The reminder of Section 2.1 gives an overview of current and past optical modem developments organized by different research groups.

2.1.1 Woods Hole Oceanographic Institution

Farr *et al.* estimate the transmit power required to maintain a bit error rate (BER) of 10^{-6} for various data rates and distances [53] [54]. Their model combines absorption and scattering losses in water into a single constant attenuation loss e^{-cR} for a distance R. This is Beer's law. They assume an e-folding length of 36 m. For a system using LEDs and a photomultiplier tube (PMT), they compute a required transmit power of 300 mW for a data rate of 10 Mbps at 100 m with a 13.4 cm large receiver aperture and 90 degree transmitter beam. If the receiver aperture is reduced to 5 cm and the transmitter beam to 10 degrees, the required power for the same data rate and distance drops to 6.5 mW. They argue that multi-path and signal spreading has no significant effect on data rates below 10 Mbps. Further, they propose that background noise caused by bioluminescence and Cherenkov radiation is low enough that it does not interfere with the signal. They create a test setup with 6 potted LEDs transmitting in a 2π sr hemisphere and a bi-alkali PMT receiver with quantum efficiency of 20 % and a gain of 10^7, driven at 400 to 500 V. They test their system at the Woods Hole docks over a distance of 10 m by sending 1, 5, and 10 MHz square pulse trains and record SNR values of 30 to 35 db. According to the numbers reported by Farr *et al.* their 90 degree transmitter system is theoretically capable of transmitting 33 Million bits/Joule radiant power over a distance of 100 m. However, the number reported during future deployments of their system is lower at 40,000 bits/Joule.

Pontbriand *et al.* use a 17 inch diameter glass sphere to improve the field of view[92]. The sphere houses a PMT, and they use an external power source and feed the output signal to an external digitizer. Their system communicates at rates up to 5 Mbps and distances up to 200 m at an e-folding length of 40 m. In turbid water with an e-folding length of 0.8 m, they achieve 1 Mbps. Their tests are performed in Bermuda and at the Woods Hole dock.

In [52] and [55], Farr *et al.* describe the Optical Telemetry System (OTS) with which they download 20 MB of data over a 5 Mbps link at a range of 80 m. They report communication at a data rate of 10 Mpbs at a distance of 108 M, 5 Mbps at distances of up to 128 m, and 1 Mbps at distances of up to 138 m. Additionally, they provide optical signal power measurements at various distances. For a source power of 15-20 W they report incident power of approximately $2 \cdot 10^{-7}$ W at 50 m, 10^{-8} W at 100 m, and $2 \cdot 10^{-9}$ W at 150 m.

Camilli *et al.* [21] discuss the application of optical communication to ocean observatories and claim that it allows for 10 Mbps with ten or more e-folding scales resulting in communication distances of 100-200 m. They categorize optical communication as permitting data rates on the order of 1 Mbps at distances close to 100 m at an energy expenditure of 40,000 bits/Joule.

In comparison, our AquaOptical II system utilizes an Avalanche Photodiode because of its increased dynamic range [71] and quantum efficiency [1]. It operates over distances of 50 m at a data rate of 4 Mbps with an energy expenditure of 400,000 bits/Joule radiant power. This results in 120,000 bits/Joule when including the total combined power consumption at the transmitter and receiver. Further, the system can cope with ambient light allowing operation near the water surface.

2.1.2 North Carolina

In [22], Chancey uses an amplifier to drive a 1 W LED with a 10BaseT Ethernet signal and a photodiode detector on the receiving side that converts the incoming light pulses back to the 10BaseT range. His design is based on an existing free space optical communication implementation called RONJA. He tests his system in a water tank over a distance of 4.6 m and extrapolates that the potential link distance is 16 m.

In [32], Cox uses a 405 nm laser for transmission and the miniature Hamamatsu R7400U PMT in conjunction with an AD8015 variable gain amplifier (VGA) on the receiving side. His system is tested using a water tank with reliable communication at 500 Kbps.

Using the same laser diode and a PNZ334 photodiode and a PC for post processing, Cox *et al.* show that the use of forward error correcting (FEC) codes achieves a BER of 10^{-4} at SNR levels 8 dB lower than without FEC [34]. They use on-off keying (OOK) in a return-to-zero (RZ) coding to transmit data at a rate of 500 kbps through water in a 3.66 m long tank. They vary the SNR by changing the attenuation coefficient of the water through the addition of Maalox. Their uncoded stream achieves approximately a BER of 0.5 at 6.5 dB SNR, 10^{-1} at 8.5 dB, 10^{-2} at 12 dB, 10^{-3} at 16 dB, and 10^{-4} at 19 dB. When using a (255,129) Reed-Solomon (RS) code, they achieve a BER of 10^{-4} at SNR 11 dB.

Simpson *et al.* show through simulation that, for a BER of 10^{-6}, the (255,223) RS code provides an SNR improvement of 4 dB and the (255,129) RS code provides an SNR improvement of 6 dB [99]. They transmit data through a water tank using a 3 W LED (CREE XR7090) at 470 nm and 525 nm. They use two setups: in one they transmit over a length of 3 m through turbid water (4+ e-folding lengths) and in the other over a length of 7.7 m through less turbid water (1-2 attenuation lengths). A PMT is used to receive the first (weak) signal and a photodiode to receive the second (stronger) signal. Their equipment is battery powered, but the signal is digitized at 100 MSPS (20 times oversampled) using external equipment. They use a matched filter to maximize SNR and early-late gate symbol synchronization. Their system achieves 5 Mbps throughput.

In [101], Simpson *et al.* propose a two-transmitter two-receiver spatial diversity system. They use a 200 mW, 405 nm laser together with a beam splitter to generate two parallel optical beams separated by 10.2 cm. Two 25.4 mm lenses focus the light onto two Thorlabs PDA36A amplified photo detectors. They investigate the use of equal gain recombination and maximum recombination for the two received signals. They transmit through a water tank into which they inject bubbles to generate interruptions of the optical beam. They show that their setup, using a data rate of 500 Kbps, reduces fading-induced errors by a factor of 10 when compared to a single

200 mW transmitter, single detector system.

Another novel approach is proposed by Cox *et al.* in [33]. They designed a MEMS Fabry-Perot modulated reflector that is used to modulate an existing optical beam. The reflector can be attached to a node or vehicle, and a querying node or vehicle shines an optical beam on it and receives the reflected and modulated signal. They use a 20 mW, 532 nm laser and a Thorlabs PDA36A amplified photodetector at the querying node. They demonstrate data rates of 500 Kbps to 1 Mbps using quadrature phase shift keying (QPSK) through a 7.7 m long tank corresponding to 2.5 e-folding lengths. Their system uses little power for the reflector but requires the beam to travel twice the transmission distance.

In [35], Cox *et al.* implement a complete duplex communication system based on 410 nm, 30 mW laser diodes and a Thorlabs PDA36A amplified photodiode receiver with a 51 mm collection lens. They use two Ettus Research USRP signal acquisition boards together with GNU radio software to generate the transmitted signal and to decode the incoming signal. Tests are conducted in a 3.66 m long tank, and different attenuation coefficients are achieved with the addition of Maalox to the water. When using 1 Mbps binary phase shift keying (BPSK) they measure the BER to be 0.2 at a SNR of 4 dB, 10^{-2} at 7 dB, 10^{-3} at 9 dB, and 10^{-4} at 10.5 dB. A duplex networking test is performed using 1 Mbps Gaussian minimum shift keying (GMSK) modulation. They achieve round trip latencies of below 10 ms at up to 5.3 e-folding lengths.

With AquaOptical II we implemented a two-layered coding scheme that can be made invisible to the user. We use a reconfigurable Reed-Solomon inner code with a system Luby Transform (LT-code) outer coder. We chose to utilize LED light sources instead of lasers in order to maintain eye-safety for human operators and to reduce pointing requirements. Because of its large aperture of 7 cm and non-focused beam AquaOptical II implicitly implements a spatial diversity system that achieves low bit-errors due to air bubbles and other scattering agents in the water.

2.1.3 Naval Air Warfare Center (NAVAIR), Patuxent River, MD

Cochenour *et al.* use laser modulation with a 70 MHz radio frequency (RF) carrier and BPSK to achieve error free communication at a data rate of 1 Mbps through highly turbid water [29]. They use a frequency-doubled diode-pumped solid-state laser with a wavelength of 532 nm and an output power of 3 W. This continuous wave (CW) optical signal is modulated using an electro-optic modulator. The optical power transmitted into the water is approximately 500 mW after losses through the optical modular and other optical elements. The detector is a PMT with a 51 mm aperture, corresponding to 18 degrees field of view. The received RF signal is bandpass filtered at 70 MHz and amplified by a 30 dB low noise amplifier (LNA) and a 50 dB variable gain amplifier (VGA). An IQ demodulator extracts the I and Q channels, which are lowpass filtered and digitized using a National Instruments DAQ card. The system is tested in a 3.6 m water tank with the turbidity of the water controlled through the addition of Maalox. A pseudorandom 1024-bit sequence is used to measure bit errors. The transmitted signal is recovered error free even at a turbidity that is equivalent to 90 e-folding lengths at 3.6 m. The phase error for the BPSK never exceeds 3 degrees. However, the authors point out that this surprisingly good result might be due to the small geometry of the tank.

In [28] this work is extended to quadrature phase shift keying (QPSK), 8 phase shift keying (8-PSK), 16 quadrature amplitude modulation (16-QAM), and 32-QAM. Tests are conducted again in a 3.6 m water tank with water turbidity adjusted to 8.4 e-folding lengths for the length of the tank. Data is sent at 10^6 symbols per second, resulting in data rates from 1 Mbps for BPSK to 5 Mbps for 32-QAM. Two transmission powers are used in the tests. With a signal power of 750 mW entering the water, all modulation schemes yield BER of 10^{-5} or better. When a signal power of 75 mW is used, BPSK yields BER 0, QPSK yields BER 10^{-3}, 8-PSK yields BER greater than 0.1, and 16-QAM and 32-QAM fail to establish a link.

In [27] Cochenour *et al.* investigate the use of a diffuser to de-collimate the laser beam. The goal is to allow communication with less pointing accuracy or even non-line-of-sight (NLOS) links. Measurements are taken by sending 10 ns laser pulses at 100 MHz and using a PMT with 1 GHz bandwidth. The pulse shapes are examined for a collimated laser beam and for a laser beam that has passed a diffuser creating a beam with a 20 degree field of view. The authors conclude that the NLOS link requires at least 30 dB more transmission power to achieve a similar signal level at the receiver.

Through the use of LEDs the light emitted from AquaOptical II is inherently de-collimated. In comparison to the above work the LEDs have a lower bandwidth which prohibits RF modulation. However, we utilize Barker-codes for packet detection which separates the signal from the background noise and Manchester coding for bits which is related to binary phase shift keying (BPSK). The use of LEDs allows for smaller and cheaper devices, increase eye-safety, and reduces aiming requirements.

2.1.4 University of Genoa

Anguita *et al.* implement a transmitter that uses 12 LEDs arranged on a circle to transmit omnidirectionally in the plane [4], [5], [8], and [7]. In two different implementations they use two different LEDs. The first is a Ledman LL1503PLBL1-301 with optical power 15 mW and a 30 degree field of view (50 % power angle). The second is an Evergreen Electronic DLE-038-045 with optical power 20 mW and a 30 degree field of view. They propose an 18 LED omnidirectional transmitter design in the shape of an octahedron. Their receiver consists of a Hamamatsu S5971 APD with an area of 1 mm^2 and a sensitivity of 0.2 A/W at 470 nm. The output signal is fed through a 10 KHz to 20 MHz bandpass filter, an amplifier with automatic gain control (AGC) based on LT1006, and a comparator to digitize the incoming signal using thresholding. The optical physical layer (PHY) is implemented inside a field programmable gate array (FPGA) on a Digilent Spartan 3 development board. They test their system with a square wave at distances of up to 2 m and report the voltage swing after the AGC. They show that their AGC allows for pulse detection from distances of a few cm to 10 m in air. Using 4- and 16-pulse-position-modulation (PPM) they report a data rate of 100 Kbps over a distance of 1.8 m in a clean water tank.

The analog front end of AquaOptical II digitizes the received signal at a resolution of 12 bits at 32 MSPS as opposed to the threshold filter described above. This allows us to implement a wide variety of available coding schemes for the physical layer, which is also housed inside an FPGA. An additional ARM processor is used by AquaOptical II to implement the MAC layer.

2.1.5 Commercial

Ambalux Corporation manufactures a commercial optical modem with a data rate of 10 Mbps [3]. They claim transmission distances of up to 40 m. Their modem interfaces over 10BaseT Ethernet and allows TCP/IP and UDP communication. Power is provided externally, with the transmitter consuming 36 W and the receiver 7.2 W. The units are rated to a depth of 60 m. Ambalux Corporations claims that higher bandwidths up to 1 Gbps and larger operating depths up to 6000 m can be achieved. However, no such device is currently available to the public.

Baiden *et al.* used an Ambalux Corporation modem in [11] and [13]. Their tests were performed at Long Lake in Sudbury, Ontario and in the pool. They achieved 9.69 Mbps over a distance of 11 m. In the pool they establish an 8.2 Mbps link at distances up to 21 m. Further, they present a novel spherical transmitter and receiver capable of transmitting at data rates of 20 Mbps over a distance of 10-11 m. The same spherical transmitter / receiver pair achieves a data rate of 115 Kbps at distances of 15 m in the field.

The devices developed by Baiden *et al.* are being commercialized by Pengiun Automated [90].

In comparison, AquaOptical II consumes 24 W at the transmitter and 10 W at the receiver when operating at full throughput over distances of 50 m. We currently allow for UDP connections over a 10BaseT Ethernet connection. Because of its reconfigurable nature, AquaOptical II can be easily adapted to use TCP/IP or other protocols.

2.1.6 Other

Hanson *et al.* demonstrated a laser based system capable of error free data rates of 1 GBit/s over a 2 m path in a tank with 36 dB extinction [65]. Maalox absorbents were added to the water. This corresponds to 8.3 e-folding lengths. Their system uses a 7 mW, 532 nm laser generated from a 1064 nm source that is optically pumped using a ytterbium-doped fiber amplifier and frequency-doubled with a periodically poled lithium niobate (PPLN) crystal. They use an avalanche photodiode as a receiver.

Chen *et al.* use a 532 nm yttrium aluminum garnet (YAG) laser to generate 10 ns wide, 200 mJ pulses [24]. This corresponds to a 2 MW laser and as a result they can only generate pulses at 50 Hz. They use 256-PPM with bin size of 10 ns. They successfully transmit data through a 50 m long water tank.

Hagem *et al.* implement a low cost and short range unidirectional optical modem [63]. They use frequency shift keying (FSK) at a data rate of 2.4 Kbps with modulation frequencies of 10 and 12 KHz. Their implementation consists of a 520 nm, 1.5 mW photodiode with a 70 degree field of view and an optical detector with a sensitivity of 0.3 A/W. They achieve error free communication at distances up to 1.1 m underwater without bubbles and up to 0.7 m with bubbles.

Lu *et al.* use a RL5-G13008 Super-Green LED, with a wavelength of 520 nm and a maximum power of 120 mW, and a Silonex SLD-BG2A photodiode together with a band-gap filter to reject infrared wavelengths [77]. The transmitter had a field of view of 90 degrees, and the receiver a field of view of 120 degrees. They use a 32-bit Gold sequence as a packet preamble for clock synchronization. They tested their system in a pool at night to avoid ambient light. The system detected 80 % of the transmitted packets at a communication distance of 10 m. Over 95 % of the packets were detected at 6 m and below. The BER was approximately $1.8 \cdot 10^{-3}$ at 2 m, $3.5 \cdot 10^{-2}$ at 6 m, and 0.5 at 10 m.

Brundage implements a unidirectional communication system using a 465 nm, 4.8 W Titan blue lighting engine (NT-52B1-0469) and a PC10-6B photodiode manufactured by Pacific Silicon Sensor [20]. The light source has a 22 degree field of view, and the system needs to be aimed. The light source is driven by a computer using a MOSFET, and the detector uses a simple comparator setup to threshold the incoming signal. The author does not specify the protocol. The system achieves error free communication at 1 Mbps at distances up to 13 m. Up to 3 Mbps were possible with unspecified amount of bit errors.

Schill *et al.* designed a small optical communication transceiver for a swarm of submersible robots [96]. They use an IrDA physical layer based on the MCP2120 IrDA encoder/decoder chip. Their receiver uses a SLD-70 BG2A photodiode together with a MAX3120 for amplification and filtering. They examine three different Luxeon III LEDs for transmission: blue (460 nm, 733 mW), cyan (490 nm, 560 mW), and green (520 nm, 165 mW). They use a byte-oriented UART serial interface at 57600 bps. They test their system in air and achieve reliable communication at 1.71 m for 460 nm, 2.02 m for 490 nm, and 1.49 m for 520 nm. All three wavelengths establish a link when tested in a pool at ranges up to 1.7 m.

2.1.7 MIT / AquaOptical

Vasilescu *et al.* design and implement an optical modem capable of optical data multing [116, 117, 48, 38]. They use a Luxeon 5 LXHL-PM02 LED with 532 nm, 700 mW radiant power. The receiver consists of the PIN photodiode PDB-C156 with a surface area of 8 mm^2. They implement digital pulse interval modulation (DPIM) with 2 bits per pulse at an average data rate of 320 Kbps. The communication range of their system is 2.2 m point to point with a cone of 30 degrees. Using a collimating lens they demonstrate error free communication through clear water over a distance of 6.4 m and communication with 4 % packet loss over 7 m.

Two early versions of AquaOptical are presented in [43]. The first is a miniaturized version capable of error free communication in ocean water at data rates of 4 Mbps over a distance of 2.2 m (470 nm) and 2.4 m (532 nm). It uses a Luxeon V LED at either 470 nm (480 mW) or 532 nm (700 mW) and a PDB-C156 photodiode. The second version of AquaOptical presented is a transmitter and receiver pair using an array of 6 such LEDs to increase transmit power and an avalanche photodiode to increase receiver sensitivity. The APD is an Advanced Photonix 5 mm^2 area Silicon APD coupled with a low noise amplifier (LNA) (part number SD197-70-74-661). We evaluate the modem in the Singapore Harbor at a data rate of 600 Kbps with error-free communication at a distance of 7 m and BER of 0.05 at 8 m. The water quality was estimated by divers to yield about 5 e-folding lengths at 8 m. In [41] we demonstrate the use of AuqaOptical for AUV teleoperation.

In [42] we present a bi-directional version, called AquaOptical II, which uses 18 Luxeon Rebel LEDs and the same APD as the previous version. The LEDs transmit at a wavelength of 470 nm with up to 1.126 W of radiant flux each, resulting in a total transmit power of 20 W. An FPGA is used for PHY processing, and an ARM7TDMI processor handles the medium access control (MAC) and provides an external 10 Mbps Ethernet interface. We demonstrate communication at a data rate of 4 Mbps at a distance of up to 50 m in a pool. In a laboratory setting transmitting through air, 32 Mbps are achieved using non-return to zero (NRZ) amplitude modulation with 2 bits per symbol. In [46], we use AquaOptical II in a data muling application where the robot estimates its position relative to a sensor node based on measured optical signal strength and simultaneously downloads payload data.

2.2 Underwater Video Streaming

Real-time transmission of live video is an essential component in a wide range of underwater applications, including the tele-operation of robotic vehicles and remote monitoring of underwater sensor stations. Conventionally, high quality underwater video streaming is implemented using tethered solutions [47, 51]. In such robot telecontrol setups however, the tether restricts locomotion range, is prone to becoming tangled, and often requires human divers to wrangle the cable and prevent harming the underwater environment.

On the other hand, wireless communication methods face a different slew of obstacles in the aquatic domain, where for instance conventional electromagnetic, i.e. radio frequency (RF) based solutions are rendered infeasible due to heavy absorption of electromagnetic waves by water. Extensive research efforts have thus focused on transmitting video payloads via alternative communication channels such as acoustic technologies [89]. Unfortunately, the acoustic channel imposes inherent bandwidth restrictions and thus limits communication throughput to be on the order of 100 Kbps [70, 2]. Consequently, acoustic-based video streaming solutions are restricted to very low resolutions and slow frame rates, which makes them impractical for dexterous tele-manipulation tasks.

In addition to advances in underwater acoustic communication technologies, a number of groups have recently begun exploring optical free-space communication as an alternative wireless transmission medium for aquatic settings. Cox *et al.* studied digital modulation over optical channels at data rates of 1 Mbps in a highly controlled test environment [36]. Simpson *et al.* investigated the use of (255,223) Reed-Solomon error correcting codes with a base transmission rate of 5 Mbps, and achieved communication distances of 7 m [100]. Anguita *et al.* presented an optical communication system with custom Physical and Media Access Control (MAC) network layers inspired by the 802.11 and 802.15.4 protocols. Their optical modem solution is deployed on a Field Programmable Gate Array (FPGA), and has a potential communication distance of up to 4 m [6]. Baiden *et al.* implemented an optical communication system with an omni-directional transmitter and receiver capable of achieving 1.5 Mbps throughput at 11 m distance in clear water environments [12]. Farr *et al.* conducted preliminary deployment of video transmission using an optical link at a data rate of 1 Mbps over a distance of 15 m [51].

In this thesis we present a novel system, operating via a free-space optical communication channel, that achieves wireless, real-time, and high-quality video delivery for underwater applications.

2.3 Data Muling

Previous work has looked at the theoretical performance of data muling [97] and the optimization of the path taken between nodes [75]. In both cases the locations of the nodes are assumed to be known. Data muling with an underwater robot has been previously shown in [48]. The nodes were found using GPS to find the coarse location and a spiral search that looked for a valid optical signal.

Moore *et al.* developed an algorithm for robust distributed sensor network localization [83]. However their system requires nodes to be at least partially within communication range of each other and does not apply to localization of a single node relative to a moving robot. Detweiler *et al.* presented an acoustic localization algorithm for the case that multiple (at least 3) acoustic beacons are available [37]. In [87] Olson *et al.* present an Extended Kalman Filter approach to localizing a moving vehicle using range-only measurements to a group of beacons. They use particle filters to initialize the beacons location. Both of these solutions require the presence of at least 3 sensor modes to localize a target and are thus impractical when attempting to data mule from a single isolated sensor node. In [113] Vander Hook *et al.* localize a single radio beacon located beneath the ice sheet of a lake with a mobile robot moving on top of the ice. Their approach is to measure points at which the signal is lost by the mobile robot and thus triangulate the sensor. Their system requires GPS localization of the mobile robot, which is not available underwater. A method for homing to a single beacon underwater using acoustic ranging based on an Extended Kalman Filter with a fixed robot maneuver for initialization is presented and evaluated in simulation in [109]. However, this system is not tested experimentally.

Looking at short-range communications for data muling, Benson *et al.* present a high-frequency acoustic network that offers range and bandwidth performance between conventional acoustic and optical rates [19].

In this thesis we present two approaches to localizing a single sensor node for the purpose of data muling. The first approach is based on gradient descent using acoustic ranging and the second approach uses a particle filter to fuse acoustic ranging and optical signal strength information. We experimentally validate both methods and demonstrate data muling over distances of up to 250 m with a with data files of up to 200 MB.

2.4 Control of Underwater Vehicles

Today a wide range of underwater vehicle designs and control architectures exist [124]. Most underwater vehicles are torpedo shaped with only one or two thrusters. These vehicles use control surfaces to change direction and depth while they move forward. Traditional pitch/roll/yaw controllers work well on these types of configurations. However, they are often not designed to hover, as they require constant movement for the control surfaces to have an effect. In contrast, our robot, AMOUR VI, is a high degree of freedom robot that can accommodate up to 8 thrusters. It is based on AMOUR V [115] with the added modularity and the ability to reconfigure its thrusters. Other robots in this class of robots include the University of Hawaii ODIN-III robot [26], the CSIRO Starbug robot [49], the Bluefin Robotics HAUV [110], and the MIT AUV Lab Reef Explorer [94]. These robots all share the ability to hover on the spot, provided that some form of localization is available to them.

The use of rotation matrices for robot control is explained in detail in [102] and rigid body attitude control using rotation matrices is highlighted in [23]. The controller described in this thesis is most closely related to work by Hanai *et al.* [64]. They propose a similar geometric solution to robot control. The desired translational and rotational forces are used to determine the force for each thruster using a least squares method. Their work assumes known thruster parameters and requires that the thrust vs. voltage curve to compute desired output values is known. Further, the controller does not have a derivative component leading to large oscillations and often needs more than 20 seconds to settle. The control algorithm presented in this thesis does not assume calibrated thrusters and adds derivative control to produce stable control loops.

Ghabcheloo *et al.* uses a similar vehicle orientation model to steer a group of robots to maintain formation. This work is done in simulation and does not address individual robots or thrusters [60]. Oh *et al.* describes how to compute paths for homing and docking using a similar vehicle orientation representation [86].

Fossen *et al.* present a survey of methods for control allocation of over-actuated marine vessels [57]. The systems are not under-actuated and this work focuses on the two dimensional case where depth, pitch, and roll are not considered. However the model could be extended to the three dimensional case.

Lee *et al.* uses Euler angles to create a 6-DOF controller for a simulated underwater vehicle. A genetic algorithm tunes the controllers [74]. This controller is subject to gimbal lock, however, their approach to tuning the individual controllers may be advantageous. Zhao *et al.* proposes an adaptive controller which does not require tuning by a human [125]. They emphasize the importance of being able to maintain a stable controller even when vehicle dynamics change.

2.4.1 Adaptive Vehicle Control

The AUV control algorithm developed in this thesis provides attitude and position control for modular robots and can handle arbitrary thruster configurations. However, the algorithm requires knowledge of the thrusters' locations. The estimation algorithm in this thesis determines the robot's inverse thruster model automatically effectively learning the robot's thruster geometry and dynamics.

Prior work on learning the thruster configuration of a robot uses neural nets to create a mapping between commands and thruster outputs. Van de Ven *et al.* give an overview of neural network control of underwater vehicles and simulation studies [112]. Gua *et al.* and Ishii *et al.* learn yaw control [62, 67] and Farrell *et al.* and Kodogiannis *et al.* learn depth control [56, 73]. They all verify their controllers in simulation as well as through experiments. Further, Lorentz *et al.* present an online adaptive single input single output neural net controller for the heave motion of the ODIN underwater vehicle [76]. They experimentally verify their controller in the pool. Gaskett *et al.* present a model-free reinforcement learning system for their underwater vehicle Kambara[59]. They evaluate their model in simulation with a two-thruster AUV (one left, one right) in the plane.

In this prior work, the neural nets attempt to simultaneously learn a model of the vehicle and a controller. The learning often takes a long time and the resulting controllers show slow response times of multiple seconds. In contrast to these works we assume the existence of a stable control algorithm and learn an inverse model of the thrusters that is then used by the controller. The approach is related to a procedure presented by Van de Ven *et al.* that uses neural nets to identify the main hydrodynamic parameters of an underwater vehicle model, assuming a priori knowledge of an inverse model of the vehicles propulsion system [111]. However, instead we determine the inverse thruster model and we do so analytically without the use of a neural net.

Chapter 3

Hardware and Systems for Underwater Observatories

This chapter describes the hardware and systems implemented in this thesis for the purpose of establishing an underwater observatory. In this thesis we designed and implemented the following systems:

- The hovering underwater robot AMOUR VI which serves as the mobile platform. It has the unique property of a fully modular thruster prolusion system that allows it to be adapted to different payloads by the user.
- The AquaOptical II high bandwidth optical modem serving as a communication link.
- We integrated our two systems with the PANDA underwater sensor node system developed by the Acoustic Research Lab (ARL) at the National University of Singapore (NUS).

These systems were developed in order to allow for agile control needed when trying to interact close up with the underwater environment, such as a coral reef, or when trying to maintain accurate position above a sensor for data muling. Further, we required a device with high data rate ($\geq$1 Mbps) for establishing a viable short distance communications link for data muling. The acoustic system was necessary to allow localization and robot travel to the node from a distance at which optical communication is not possible. The rest of this section describes each of these systems in detail.

3.1 Hovering Underwater Robot AMOUR VI

In this section we present the design and operation of the sixth iteration of AMOUR (Autonomous Modular Optical Underwater Robot), an AUV tailored for operations in coastal waters and reef environments. The previous versions of AMOUR were presented in [115, 114, 118]. Its applications include marine biology studies, environmental, pollution and port security monitoring. The main design requirements for our robot have been:

1. a small, light-weight package that can be carried by a single person and fit in a suitcase
2. low-cost modular configuration that can be adapted easily to specific needs
3. ability to hover and, more specifically, ability to be controlled in six degrees of freedom if desired
4. ability to interact with an underwater sensor network and
5. ability to carry standard marine instruments over medium-range missions (10 km).

We considered the option of using and adapting an existing underwater robot design, but decided that this path would not easily satisfy our design requirements. We opted to design a new robot system from scratch in order to address the requirements in an organic integrated fashion, as opposed to an add-on approach. The sixth version of AMOUR provides the following improvements over previous versions:

1. Emphasis was placed on the modularity of the robot. In particular we wanted the ability to mount varying amounts of thrusters in many different spatial

configurations to the robot. This would allow to adapt the robot to new sensors and payloads and also to replace thrusters in the field, should they be damaged.

2. The remaining parts of the robot were designed modular as well: The battery, computation, and power hub are all individual components often only connect with a power and communication bus. They can all be easily exchanged.

3. All components of AMOUR VI are designed to be interchangeable between multiple instances of the robot. We build a total of 3 complete robots with 5 thrusters each.

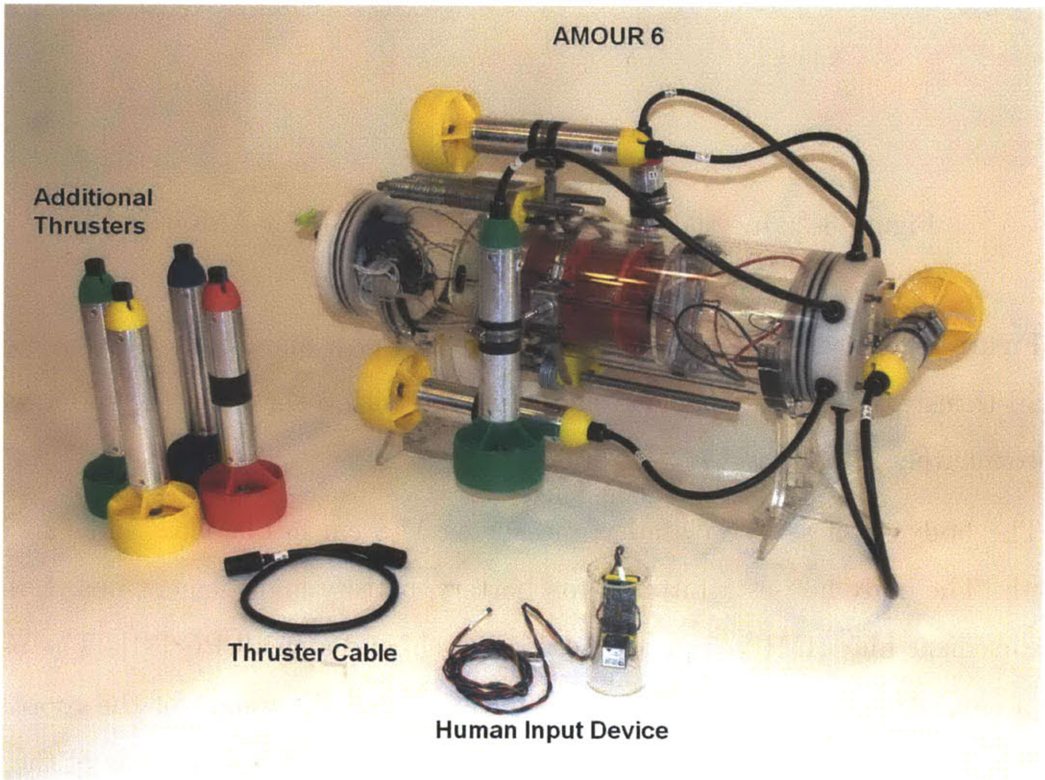


Figure 3-1: The robot assembly with 10 thrusters and the human input device (cylindrical object in the front). Five thrusters are attached in the same configuration as in Figure 4-2 and a sixth thruster is attached to the front of the robot to make it fully holonomic. The other four thrusters are for replacement. The left side of the robot contains the IMU and the controllers. The battery is visible in the middle (red). The right side contains the power electronics. Below the top thruster and at the bottom of the robot the external ballast is visible that is necessary to weight down the robot. The robot is over 3kg buoyant without any payload attached.

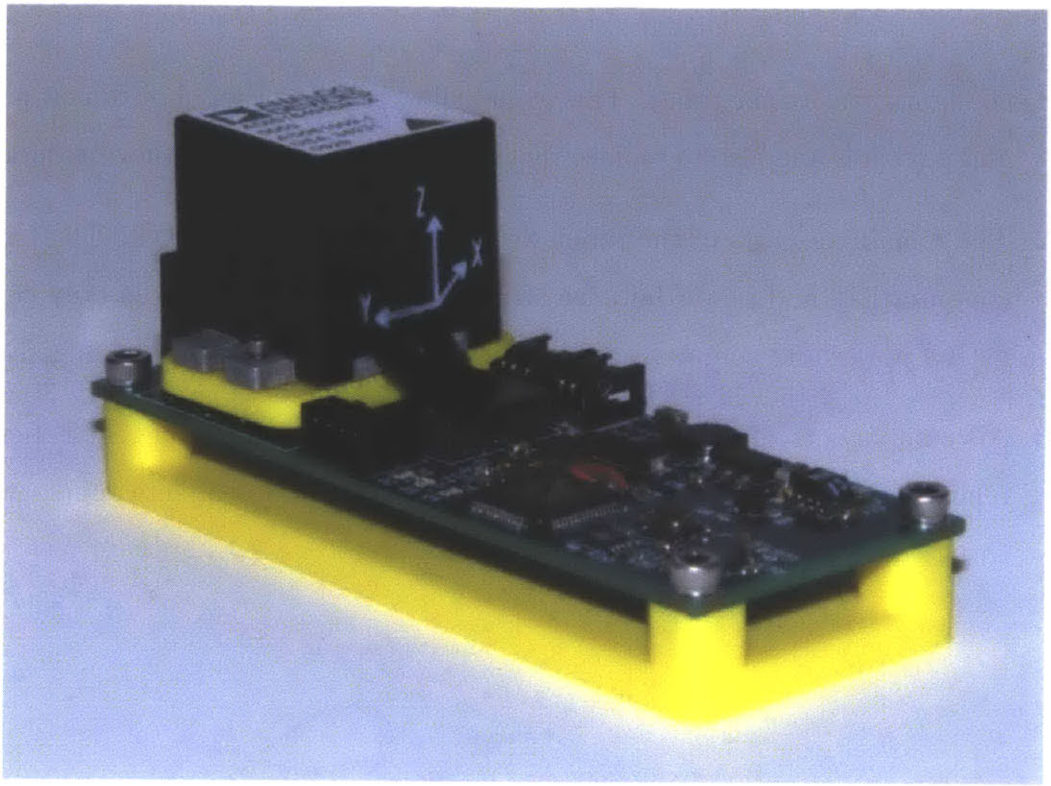


Figure 3-2: Inertial Measurement Unit (IMU) for AMOUR VI

Figure 3-1 shows AMOUR VI. It is composed of a main body and up to 8 external thrusters. Without payload and equipped with 5 thrusters the robot weights approximately 17.5 kg and is 3 kg buoyant.

The body is an acrylic cylinder, measuring 17.8 cm in diameter and 72.4 cm in length. The body houses a Lithium-Ion battery, battery management board, inertial measurement unit (IMU), 5 port ethernet switch, and a small PC [31]. The battery has a capacity of 645 Wh and amounts for a third of the weight of the robot. The battery is actuated by an electric motor and it can travel along the axis of the robot to shift the robot's center of mass [115]. At each end, the body has a two 10 cm empty section that can be used for additional dry payloads. The battery management board distributes the power, measures the robot's current consumption and tracks the precise state of the battery using a charge counter. The IMU is shown in Figure 3-2. It estimates the pose and the depth of the robot at a rate of 400 Hz by fusing the raw data from 10 sensors (one pressure sensor, 3 magnetic field sensors, 3 accelerometers

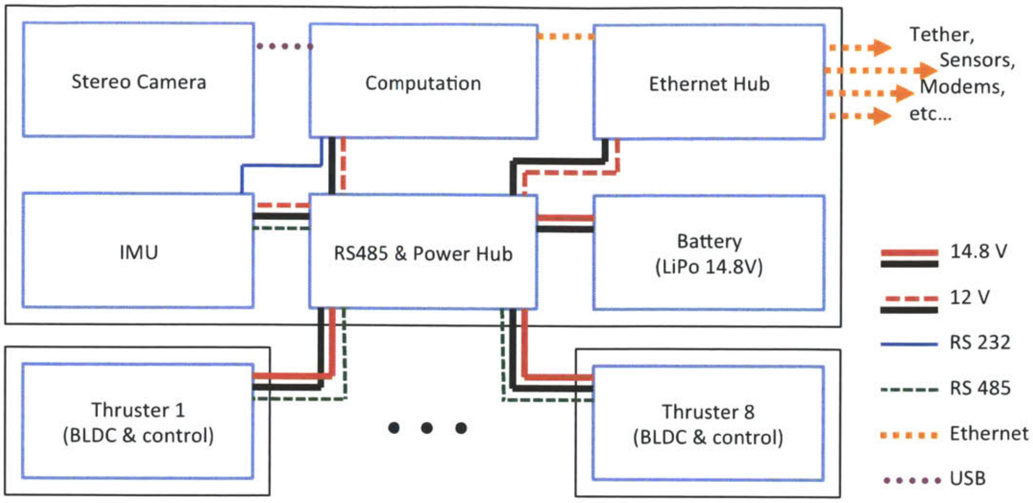


Figure 3-3: System Overview of AMOUR VI showing modularity of thrusters. The blue boxes represent individual components of the system. The black boxes represent individually sealed components of the system, which are easily interchangeable in the field. The separation of components into hermetically sealed enclosures guarantees that if one part of the system is damages or fails otherwise, the other parts will not be affected by the water leak.

and 3 gyroscopes). An additional sensor board [17] can be used for (1) reading of GPS and marine analog and digital sensors, (2) data logging and (3) radio communications. The PC is used for high level control and mission planning, data logging and online data processing of the sensory payload.

3.1.1 Modular and Reconfigurable Thrusters

The thrusters are designed in house. Each thruster is composed of a motor controller and a 600 W geared brushless DC motor driving a Kort nozzle propeller with a diameter of 9.4 cm. The motor and the electronics are housed in an aluminum cylinder, with a diameter of 4.8 cm and a length of 25.4 cm. Each thruster can generate up to 4 kg of static thrust. The thrusters receive commands via a RS485 bus.

Figure 3-3 shows the system layout of AMOUR VI. AMOUR's configuration is very modular. The thrusters can be attached along the robot's body or to the robot's end caps. Each thruster is connected electrically to one of the 8 ports fitted on

AMOUR's bottom cap. Replacing a thruster or changing the thruster configuration can be done in less than 5 minutes. AMOUR is designed to carry internal and external payloads. AMOUR's control algorithms and high power thrusters allow precise and fast manipulation of payloads of size similar to its own.

Two sample configurations of AMOUR VI are shown in Figures 3-4 and 3-5. The first shows AMOUR VI with 5 thrusters that allow for yaw, pitch, roll, surge (forward/backward) and heave (up/down) movements. Since only 5 thrusters are mounted, one degree of freedom (sway - left/right) is missing. As a result the controller presented in this thesis will ignore sway components of any command issued as they cannot be implemented for this thruster configuration. The configuration in Figure 3-5 shows AMOUR VI caring an underwater real color imaging sensor [121]. To make up for the resulting bulky shape of the robot the assembly is configured with a total of 8 thrusters such that all 6 degrees of freedom are available for control.

3.1.2 Human Input Device

We built a miniature model of AMOUR fitted with an IMU, to be used as a human input device (Figure 3-1). The operator can use this device to command the robot's pose quickly and intuitively. When in use, the IMU inside the model reports its orientation, which can be transmitted to AMOUR's controller at a rate of 200 Hz. We used this device to evaluate the robot's response to complex motions. In the future we envision researchers using this device to remote control the robot for video shooting, manipulation, and data gathering.

3.2 High Speed Optical Modem AquaOptical

AquaOptical II is a bidirectional, high data-rate, long-range, underwater optical communication system. Two AquaOptical II modems can be seen in Figure 3-6. In its default configuration AquaOptical II can communicate over distances of 50 m at data

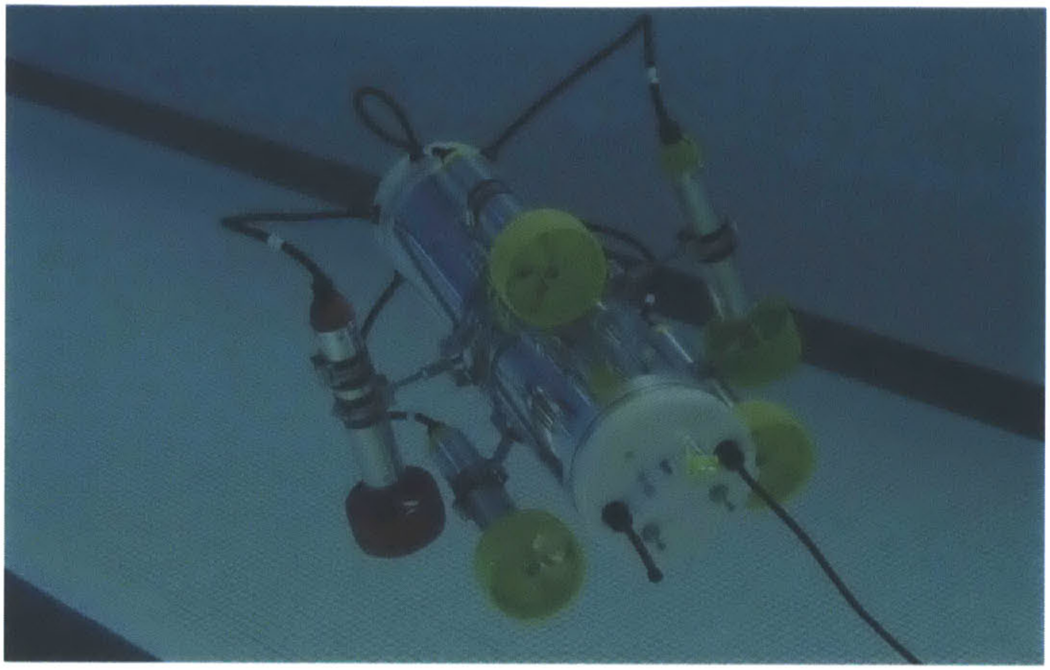


Figure 3-4: AMOUR VI configured with 5 thrusters in the water. This configuration allows the robot to perform yaw, pitch, roll, surge (forward/backward), and heave (up/down) maneuvers. Because a degree of freedom is missing, sway (left/right) maneuvers cannot be executed.

rates of 4 Mbps. However, the system can be operated in a "software defined radio" (SDR) mode [40, 82]. In SDR mode, each AquaOptical II modem can be programmed to transmit user defined waveforms and record the received waveforms for detailed analysis. This allows for the use of many different modulation schemes. We describe the hardware and software architecture we developed for these goals. AquaOptical II is based on the uni-directional version AquaOptical I, which can be seen in Figure 3-7. The major differences are the ability for bi-directional communication, software defined radio principle, and improved power management.

AquaOptical Micro is a miniature version of AquaOptical that was designed as an early case study to be fitted into underwater sensor nodes called AquaNode [120]. It can communicate with AquaOptical II and is described at the end of this section.

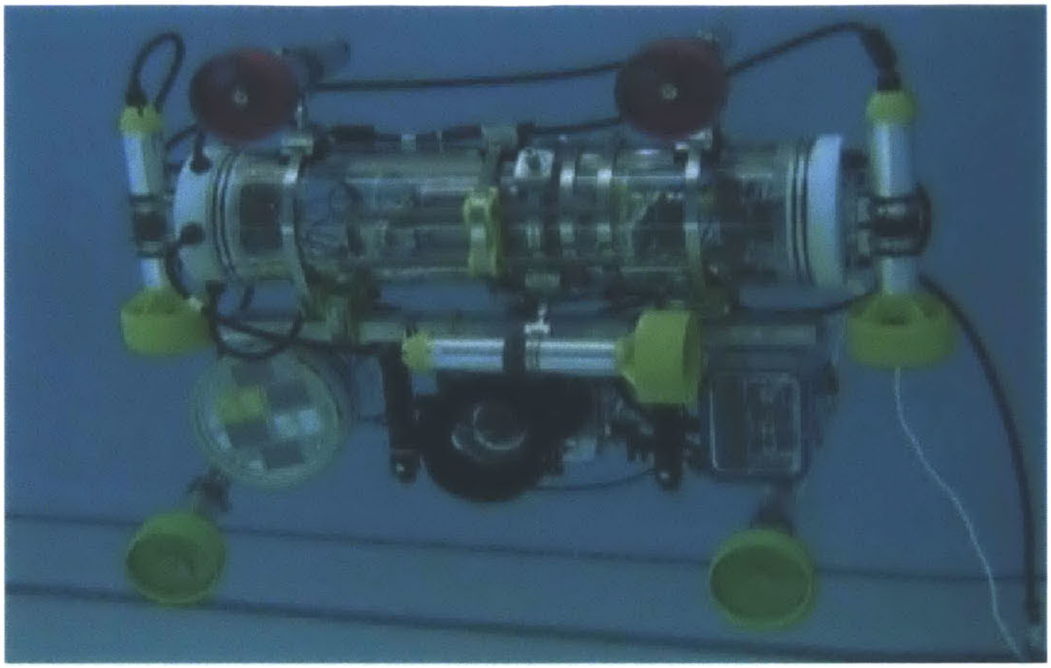


Figure 3-5: AMOUR VI configured with 8 thrusters to carry a real color underwater imaging device. Movements in all 6 degrees of freedom are possible with this configuration.

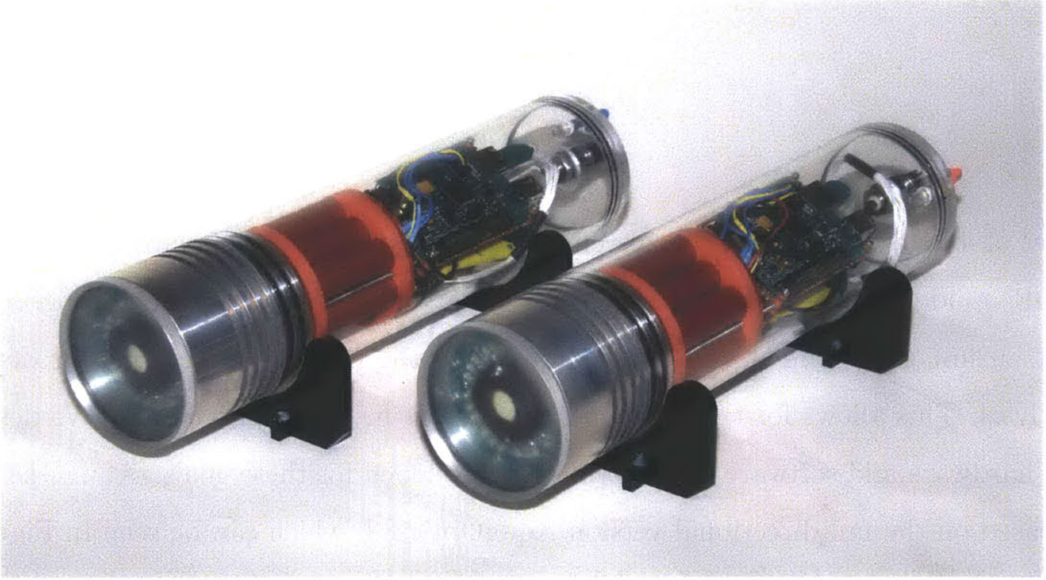


Figure 3-6: Two AquaOptical II modems.

3.2.1 AquaOptical II

A system overview of the AquaOptical II modem is shown in Fig. 8-2. This network interface can be decomposed into three different sections: (1) a transducer comprising

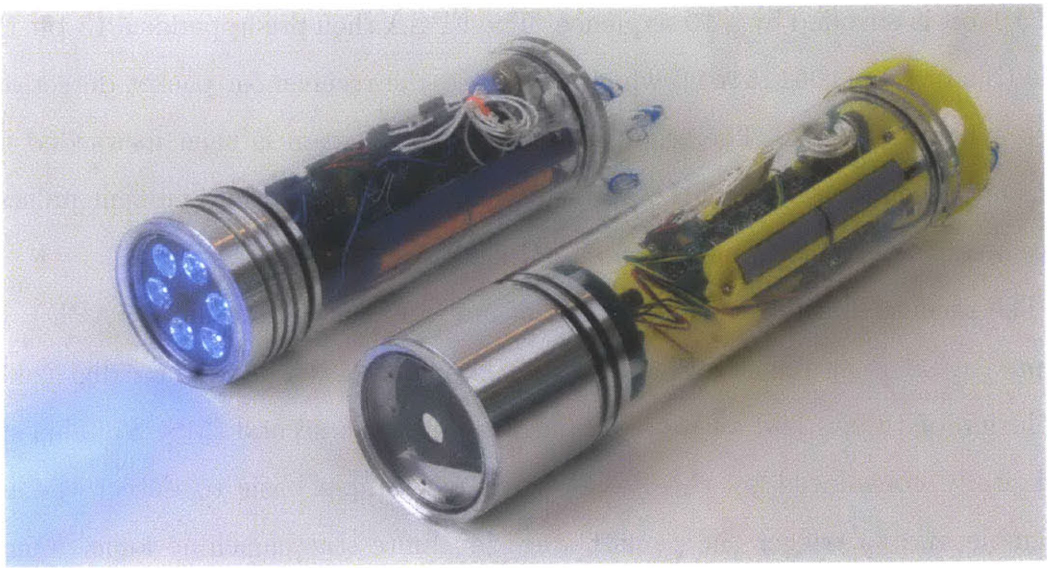


Figure 3-7: AquaOptical I modems. The transmitter is on the left and the receiver is on the right.

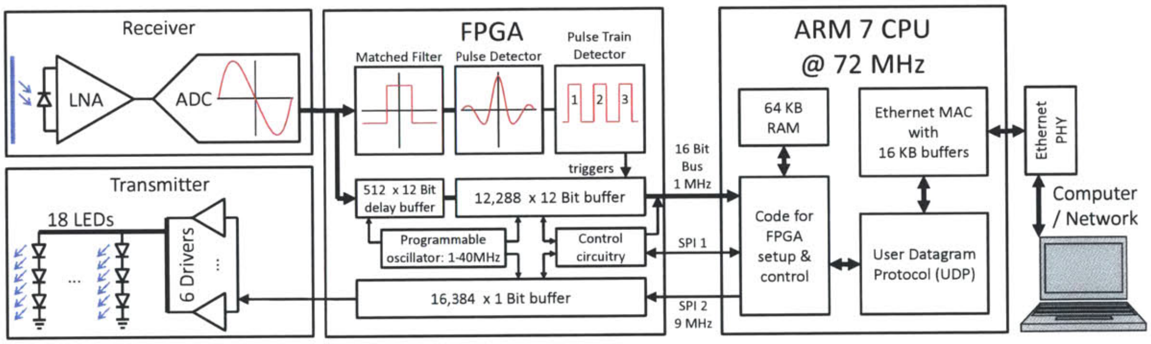


Figure 3-8: Overview of the AquaOptical II hardware and software. The hardware consists of a transmitter and a receiver module, and FPGA, and a CPU. The hardware is described in detail in section 3.2.3. The software modules are located inside the FPGA and CPU and are described in section 3.2.4.

the transmitter and receiver hardware, (2) an FPGA that implements the physical and link layers (as defined by the OSI network model), and (3) an ARM 7 CPU that handles the MAC layer and Ethernet communication.

The AquaOptical II modems receive and transmit payload data using the minimalistic User Datagram Protocol (UDP). The ARM CPU forwards all UDP packets received through the Ethernet interface of AquaOptical II to the FPGA and triggers a `packet transmit` signal. The FPGA then converts the raw bit stream into a transmit bit stream using Manchester coding, where a `0`-bit is encoded by a `01` sequence and a 1-bit is encoded by a `10` sequence. The FPGA then pre-appends a 13-bit Barker code (binary: `111100110101`), which is used by the receiver for packet detection and clock synchronization. The resulting transmit bit stream is then forwarded to the transmitter hardware at 8 MHz, and subsequently converted into current pulses that drive a series of Light Emission Diodes (LED).

The receiver hardware is comprised of an Avalanche photo-diode (APD), whose output signal is amplified by a low noise amplifier (LNA) and then digitized at a resolution of 12 bits and a rate of 32 mega-samples per second (MSPS). This stream is digitally convoluted by the FPGA on a per-sample basis to detect the packet preamble and to trigger the `packet decoder` finite state machine logic. Once the Barker code preamble is identified, a step function matched filter (with 8 samples width, to match an established maximum throughput rate of 4 Mbps) is used to decode the Manchester symbols, where the sign of the filter output is sampled into a binary value. The first 16 bits decoded this way are used to determine the length of the packet, and the remainder of the payload is subsequently stored in a buffer. Once the expected number of bits are sampled, the ARM CPU forwards the contents of the buffer as a UDP packet over Ethernet to the end user.

3.2.2 Software defined radio approach

We designed this version of AquaOptical II specifically with a software defined radio approach in mind [40, 82]. This means that we can generate arbitrary transmission waveforms within the constraints of the LED drive circuitry and that the received waveform is digitized and stored as a whole, with little or no pre-processing. The software defined radio design allows us to test different encoding and decoding algorithms and to switch these on the fly in the field. Since such an approach generates significant amounts of data (up to 60 MB/sec for this receiver) it can limit the throughput of the modem (while the base bandwidth stays unaffected). However, we can directly implement encoding schemes such as Digital Pulse Interval Modulation (DPIM) inside the FPGA located in the modem, as demonstrated in the previous version of AquaOptical, which offered less computational power. We designed AquaOptical II

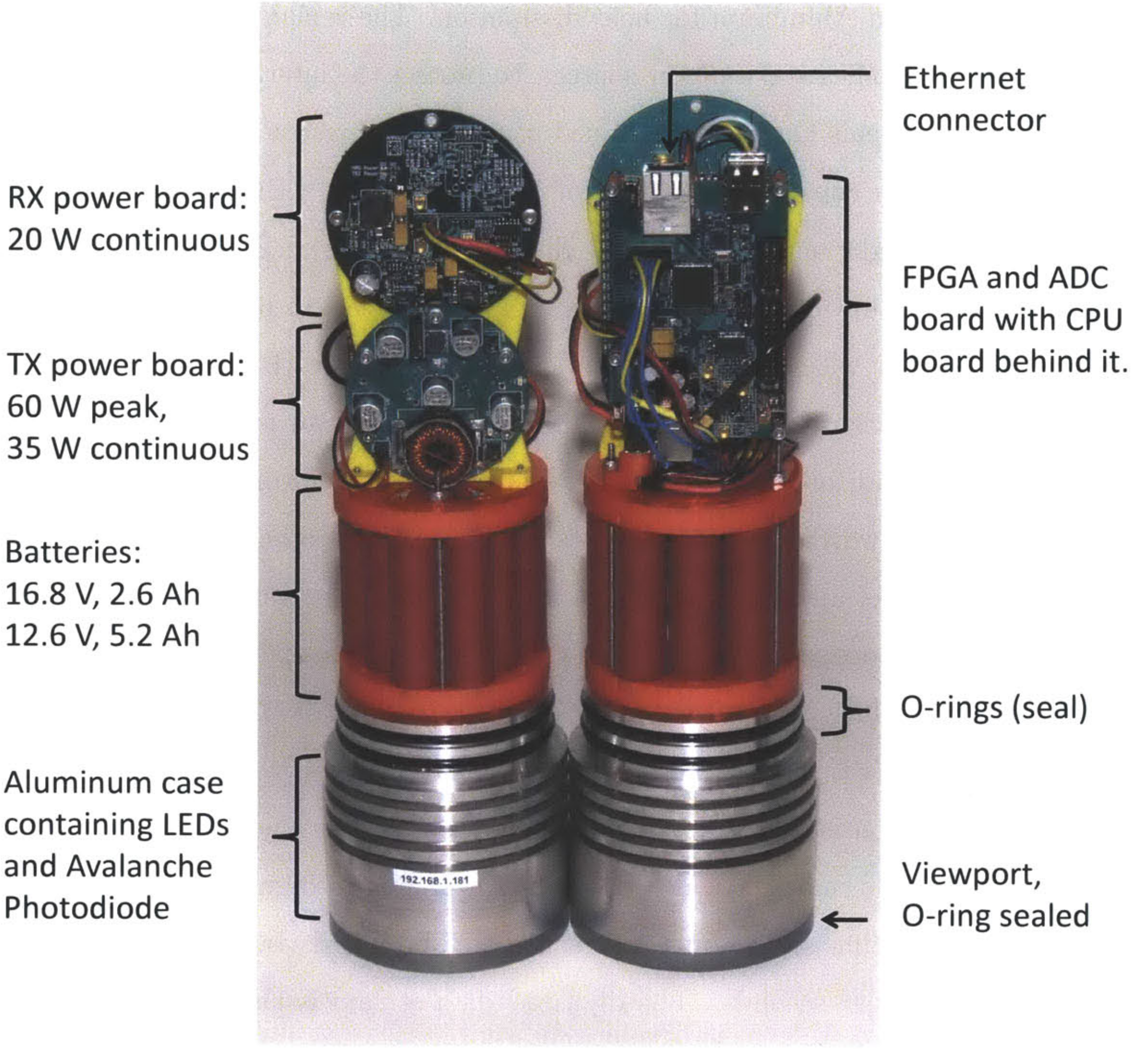

Ethernet connector
RX power board: 20 W continuous
FPGA and ADC board with CPU board behind it.
TX power board: 60 W peak, 35 W continuous
Batteries: 16.8 V, 2.6 Ah 12.6 V, 5.2 Ah
O-rings (seal)
Aluminum case containing LEDs and Avalanche Photodiode
192.168.1.181
Viewport, O-ring sealed

Figure 3-9: Two AquaOptical II modems with the acrylic housing removed.

for theoretical operation of up to 10 Mbps in the case that the encoder and decoder are located entirely inside the FPGA.

3.2.3 Hardware

Figure 3-9 shows the hardware components of AquaOptical II and Figure 7-2 gives presents a system overview. AquaOptical II is designed as a bidirectional communication device where each unit can both send and received data. Each modem is contained inside a water-tight tube of 8 cm diameter and 35 cm length and weights 2000 g. The transmitter and receiver are both housed at the front part of the tube, which is made from Aluminum for heat dissipation. The remaining part of the tube contains the electronics and a power source. To protect the internal components from water damage all seals have been designed using two o-rings each.

The transmitter consists of an array of 18 Luxeon Rebel LEDs that emit 470 nm light. The total light output in radiant flux is 10 W[1]. We measured the LEDs operational range at up to 5 MHz with a minimum pulse length of 100 ns. An ARM7 processor, together with a Field-Programmable Gate Array (FPGA), and an additional drive stage are used to control the LEDs. The processor is used to encode user data to be sent and forwards a raw discrete (on/off) waveform to the FPGA. The FPGA buffers the waveform while it waits for the medium to be free (no light pulses detected for a preset amount of time). It then outputs the waveform onto the LEDs. Waveforms can be output at a rate of up to 40 Mega samples per second allowing for a fine-grained resolution of pulse positions within the signal. The waveforms buffer is written by the CPU using a Serial Peripheral Interface (SPI) running at 9 MHz.

The receiver consists of an Avalanche Photo-diode, which includes a low-noise amplifier and is thermoelectrically cooled. An analog-to-digital converter (ADC) converts the resulting signal into a stream of 12-bit words by sampling at up to 40 Mega samples per seconds. The digitized data stream is then analyzed the the same FPGA used for transmission. The FPGA uses a matched filter to detect single pulses. If a train of a preset number of pulses is detected, the FPGA records up to $12,288$ samples into a buffer. This buffer is then transmitted to the CPU processor for decoding using a 16-bit parallel bus running at 1 MHz.

3.2.4 Software

The AquaOptical II software is divided into two parts: (1) The software inside the FPGA which controls the transmission and sampling timing as well as packet detection and (2) the software inside the CPU which provides an interface to the user through the User Datagram Protocol (UDP).

[1] Each LED is driven at 600 mA and outputs approximately $35lm$ at 470nm [80]. $18 LEDs \cdot 35lm/LED \cdot 16\ mW/lm \approx 10\ W$

The software inside the FPGA forms the heart of the system. It handles the following tasks:

1. Checking if the medium is busy to avoid packet collisions. This is done using a matched filter with an expected pulse shape that can be programmed attached to a pulse detector. After every pulse the medium is marked as busy for a preset time.

2. Detecting new data transfers (packets). A counter is used to count pulses that occur within a preset distance of each other. The detection of such a pulse train will trigger the receiving buffer to record data. This prevents single pulse detections cause by noise to trigger unnecessary recordings.

3. Buffering and transmitting waveforms at a preset sample rate when the medium becomes free. A 16,384 Bit long buffer is used for this purpose. It can be written to by the CPU using a 9 Mbps SPI bus. Once the data is marked as valid by the CPU the FPGA will wait for the medium to become available and will immediately start transmitting the waveform.

4. Continuously pre-buffering the signal so that when a packet is detected the waveform is recorded including samples ahead of the packet. This serves to compute the amplitude of the noise and to possibly detect pulses that we missed by the FPGA internal detector. A 512 sample deep buffer is used to continuously delay the incoming signal before it reaches the record buffer.

5. Record the waveform upon packet detection. The waveform is recorded into a 12,288 sample deep buffer. The buffers size is limited by the memory available inside the FPGA.

The software inside the CPU provides an interface to the user through the use of UDP. It handles the following tasks:

1. Receive commands over the network to set configuration values inside the FPGA, such as the oscillator to a desired frequency. This will affect the sample rate for

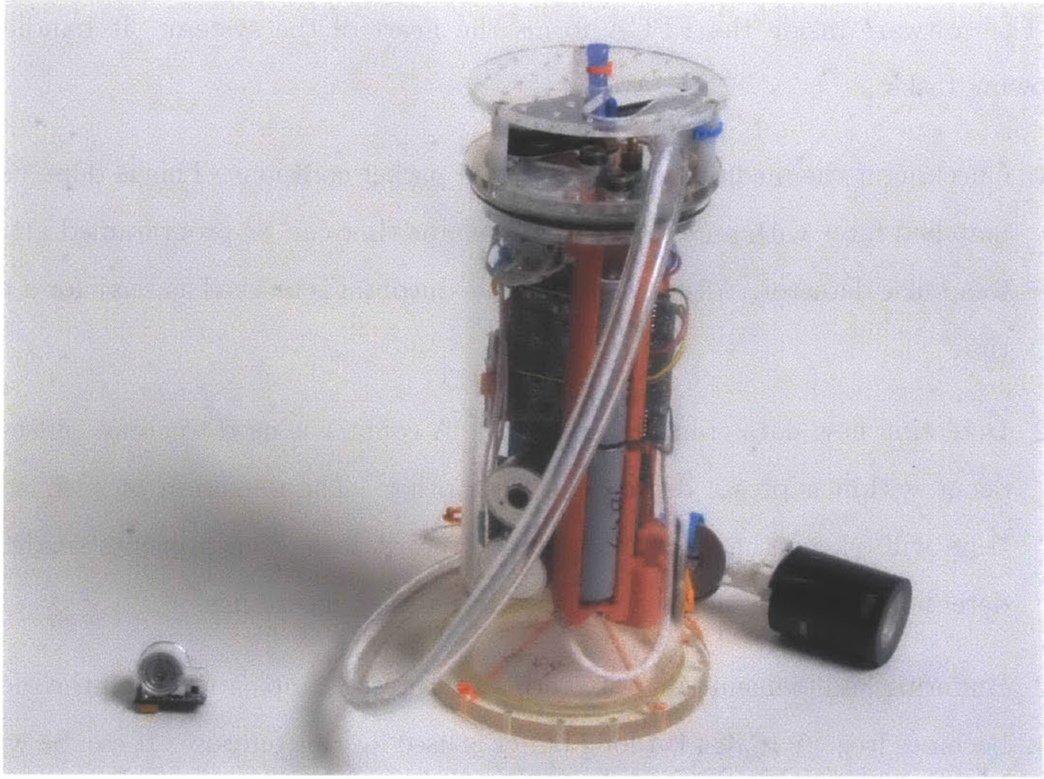


Figure 3-10: AquaOptical Micro modem (left) next to a 31 cm high AquaNode into which it can be integrate.

transmission and reception. Further configuration parameters are the desired matched filter shape, the number of pulses needed to trigger a recording, and the ability to shut down the receiver to reduce energy consumption[2].

2. Receive commands to transmit a waveform. The waveform is first cued inside the CPU and sent to the FPGA as soon as its transmit buffer becomes available (i.e. the previous waveform was sent).

3. Transmit received waveforms over UDP back over the network to the host machine. The host machine is determined by remembering the IP address of the last machine that sent a configuration packet.

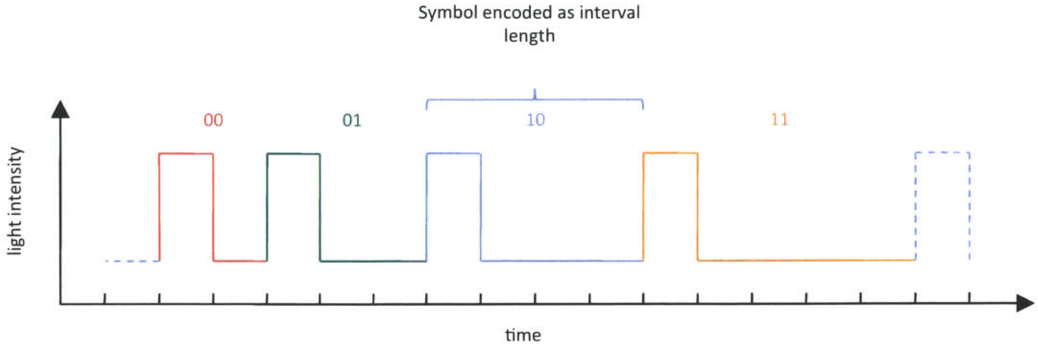


Figure 3-11: Digital Pulse Interval Modulation.

3.2.5 AquaOptical Micro

The high-level architecture of the short-range optical modem is similar to the architecture of the AquaOptical II. Each AquaOptical Micro modem includes an optical receiver and an optical transmitter. AquaOptical Micro can communicate at ranges of 3 m and data rates of 1 Mbps.

The AquaOptical Micro receiver uses a commonly available and inexpensive photodiode produced by Advanced Photonix, Inc. (part number PDB-C156). The actual sensor of the diode is 8.02 mm^2 large and has a response of 1418 A per watt of light received. The output of the diode is digitized using an IrDA receiver chip produced by Linear Technology (part number LT1328). Just as in AquaOptical II, the digitized output is decoded using an FPGA and serialized into a byte stream. The AquaOptical Micro transmitter consists of a 5 W LED that emits 480 nm light. An FPGA is used to encode a raw data stream into symbols for the physical layer using DPIM. Each byte is converted into four symbols, each of which is represented by a different length pulse. The packets are delimited by a fifth symbol whose pulse length is larger than the other symbols used to encode data. Each packet begins with a 4-byte header containing transmitter address, receiver address, packet length, and packet type and terminates with a CRC byte.

In contrast, AquaOptical II uses an array of 18 LEDs for transmission and an avalanche photodiode, which is far more sensitive than the photodiode used in the

[2] The transmitter circuitry consumes no significant amounts of energy when not transmitting and has thus no option to shut it down.

short-range system and much more expensive. The costs of an avalanche photodiode with the primary control circuitry are two orders of magnitude higher than those of a simple photodiode such as the PDB-C156. Thus, the advantages for the PDB-C156 diode and the LT1328 combination are as follows:

- small size (see Figure 3-10),
- easy to use/control,
- low power consumption,
- low heat dissipation,
- and low cost. Its disadvantages are as follows:
- a worse signal response than the avalanche diode
- and digitization of the signal at an early stage (no low-noise amplifier, just a preamp), thus losing possible information. The advantages for the long-range modem are as follows:
  - higher sensitivity,
  - better amplifiers (this making the detection of very weak pulses possible and increasing communication range)

  The disadvantages are as follows:
- size (more than 10 times the volume of the PDB-C156/LT1328 assembly),
- its power consumption (5 W for cooling and additional power for control),
- the resultant heat dissipation which requires active cooling and a large heat sink, and
- its significantly higher cost.

3.3 Underwater Sensor Network PANDA

We have integrated of our AquaOptical underwater optical communication system with the Pop-up Ambient Noise Data Acquisition (PANDA) system, which was developed by the Acoustic Research Lab at National University Singapore [72, 25]. PANDA is an underwater sensor node equipped with a long range acoustic modem. The current version of PANDA is called UNET-PANDA, which is presented, along with the UNET2 acoustic modem used, in Chitre *et al.* [25]. The UNET2 modems use a carrier frequency of 27 kHz with a transmission bandwidth of 18 kHz. The maximum power level for transmissions is 180 dB measured at 1 m. In this thesis we use Orthogonal Frequency Division Multiplexing with 256 carriers per symbol. For the inner code we chose a 1/3-rate convolution code and for the outer code we chose a 12/23 Golay code. The acoustic modem on the PANDA can transmit an 18 byte long ranging beacon every at least 3 s that is received by the acoustic modem on the robot and provides it with a range measurement. For simplicity UNET-PANDA is referred to as PANDA in the remainder of this thesis.

Figure 3-12 shows a PANDA node with an attached AquaOptical II modem. The connection is made with an underwater Ethernet cable and establishes a 10 Mbps half-duplex link between AuqaOptical II and the onboard computation on the PANDA node. A java program running on the PANDA was used to interface with AquaOptical II.

The goal of the integration was to create a dual layer communication network, with the acoustic layer serving as a long range, lower bandwidth communication channel, and the optical layer as a shorter range, but high bandwidth communication channel. This combined system serves as a network that provides communication and localization to our underwater robot AMOUR and potentially other AUVs. The acoustic communication can be used for basic status and configuration messages and for ranging and localization measurements. The optical communication will serve the purpose of high speed data downloads by AUVs. This enables users to leave the PANDA sensor undisturbed on the ocean floor during downloads, whereas previously

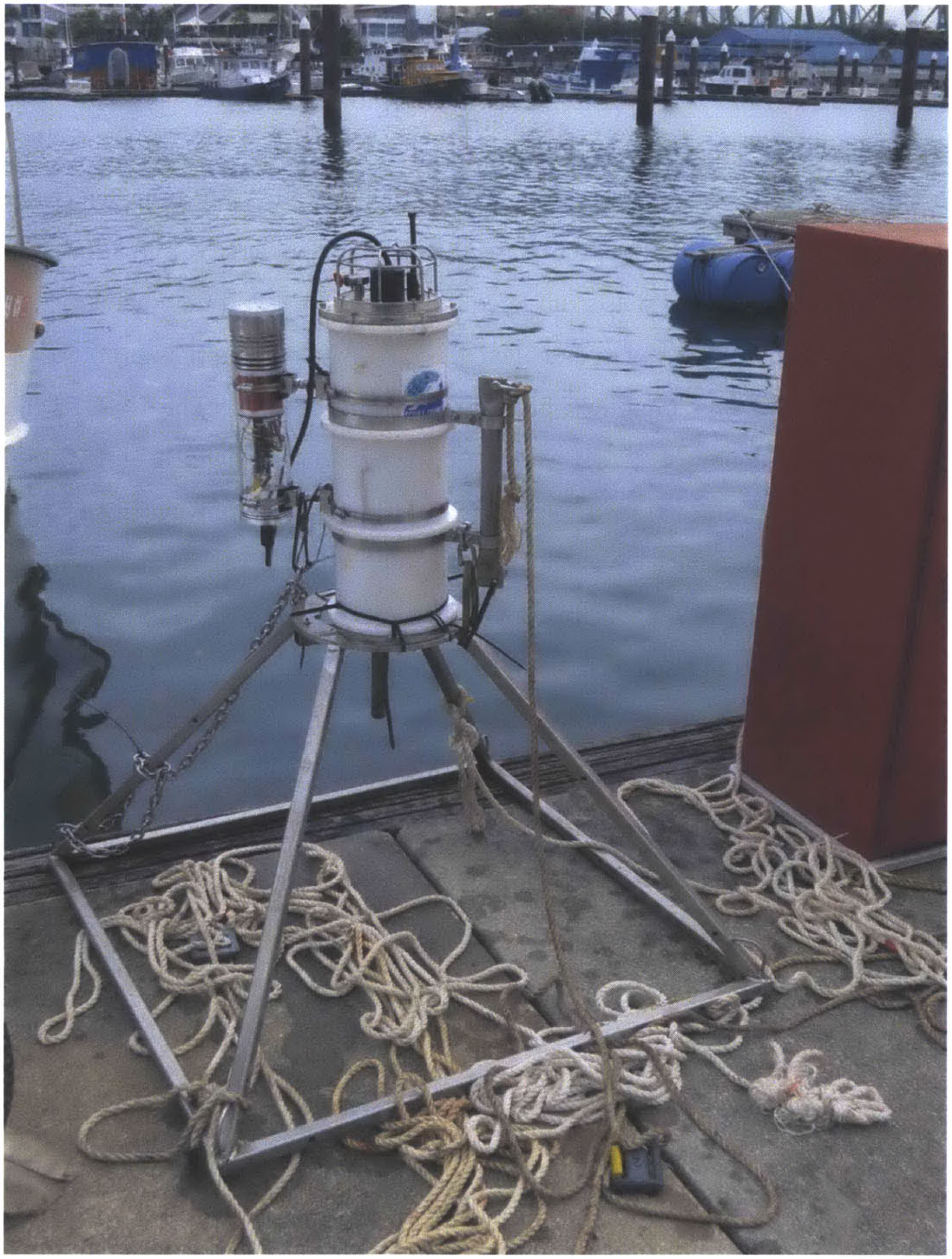


Figure 3-12: PANDA node (white cylinder on tripod) with Optical Modem attached on the left. A black underwater ethernet cable can be seen connecting the two.

they have to be retrieved for data recovery. We have used acoustic ranging to guide AUVs towards the PANDA node for data retrieval by means of data muling. During a data muling mission the acoustic system was used by the AUV to find the PANDA nodes that have freshly logged sensor data ready for download. Further the system helped maintain the vehicles location during optical data retrieval.

Chapter 4

Control of Reconfigurable Robotic Mobile Nodes

This chapter addresses the issue of controlling the class of robots with arbitrary thruster configurations in up to 6 degrees of freedom. In particular, we focus on our robot AMOUR VI. The algorithms presented in this chapter allow AMOUR VI to act as an agile mobile node in the observatory. This is an essential step in permitting detailed interaction with the environment for the purpose of observation, manipulation, and communication.

This chapter first address control of AMOUR VI under the assumption that the thruster configurations are known. The second part of this chapter relaxes this constraint and provides a solution to cases in which the thruster configurations are not known a priori.

4.1 Control without known Vehicle Dynamics

AMOUR VI is a small, highly maneuverable, underwater autonomous vehicle designed to be a flexible underwater platform that can operate in shallow ocean environments. The robot's tasks include deploying static sensor nodes, taking pictures and videos of coral reefs for environmental monitoring, performing visual and acoustic ship-hull

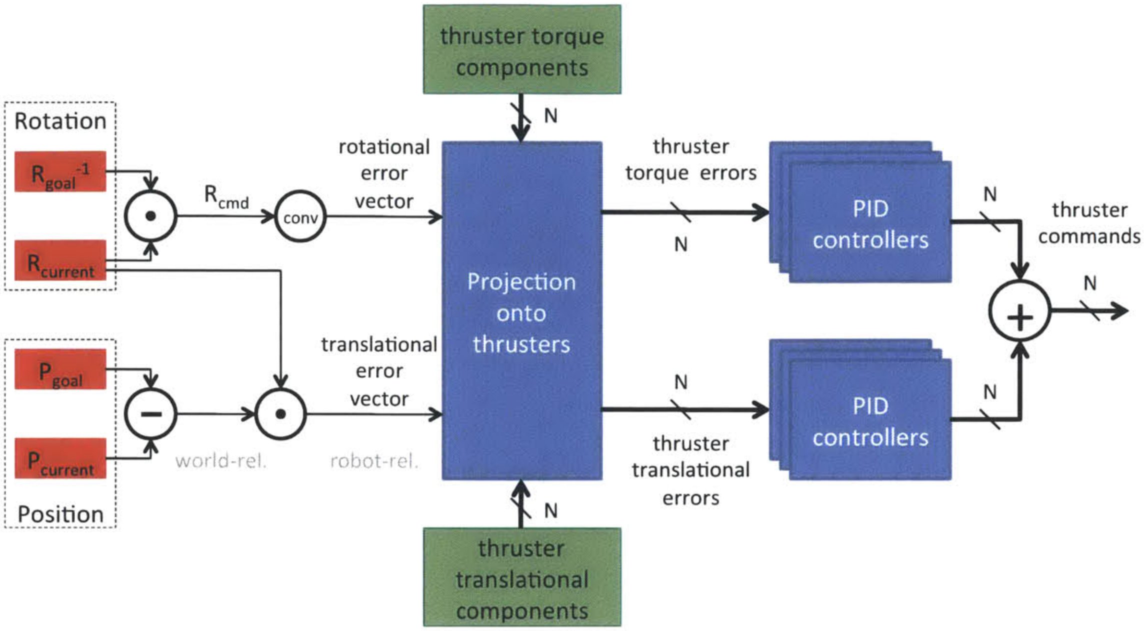


Figure 4-1: Modular Thruster Control Algorithm (MTCA).

inspection, and docking with other underwater vehicles.

To accommodate this wide range of activities we must be able to quickly and easily reconfigure the shape, payload and thruster configuration of the robot. For instance, for coral reef monitoring we attach a relatively large professional SLR camera and flash to the robot [114]. This changes the dynamics of the robot while requiring an extremely stable platform that can point the camera in arbitrary orientations to capture the desired photographs. Other tasks, such as sensor deployment, requires dropping off a number of different sensors to monitor the environment [39]. As each sensor is deployed the dynamics of the robot changes.

Further, we intend to use the robot as a data mule for the static sensor nodes by using optical communication. The more precise the alignment of the optical transceivers the better the optical link quality and the resulting data rate. Aligning the optical transceivers can be challenging in the dynamics of the ocean. Having a stable platform which can aim the optical transceiver at arbitrary locations is critical for this application.

In this chapter we describe a new Modular Thruster Control Algorithm that supports an arbitrary number of thrusters and thruster configurations and can effect arbitrary orientations on a robot with a modular thruster system. The orientation of the robot can be arbitrarily set and changed while maintaining independent control of the translational components (e.g. the robot can be rolling head-over-heels while maintaining depth). More specifically, the attitude can be arbitrarily chosen from the special orthogonal group SO^3 (rotation group for three-dimensional space) while choosing the translational velocities from R^3. So far, the algorithm requires knowledge of the thruster configuration. The second part of this chapter presents and algorithm for estimating previously unknown thruster configurations in situ. This allows for control of robots with previously unknown configurations as well as recovery if thrusters fail or are moved during a collision.

This chapter is organized as follows. We first present the algorithm for modular thruster control. We discuss the implementation of the algorithm on the AMOUR underwater vehicle and present the results of experiments demonstrating the stability of our controller in many different orientations. Finally, we show that the controller supports the addition of another thruster to the robot.

4.2 Control with Arbitrary Thruster Configurations

This section describes the Modular Thruster Control Algorithm. We first state the assumptions, the necessary input data, and what is provided as output. Then we give a high level explanation of how we achieve complete control in the special euclidean group SE^3 (group of all translations and rotations in three-dimensional space). Finally, we present the details of the Modular Thruster Control Algorithm.

4.2.1 Assumptions, input, and output

We assume that the robot knows the position and orientation of the thruster's and that the robot is capable of determining its own attitude. In particular, the algorithm presented takes as input a measured acceleration vector, $\mathbf{a} \in \mathcal{R}^3$, and a measured

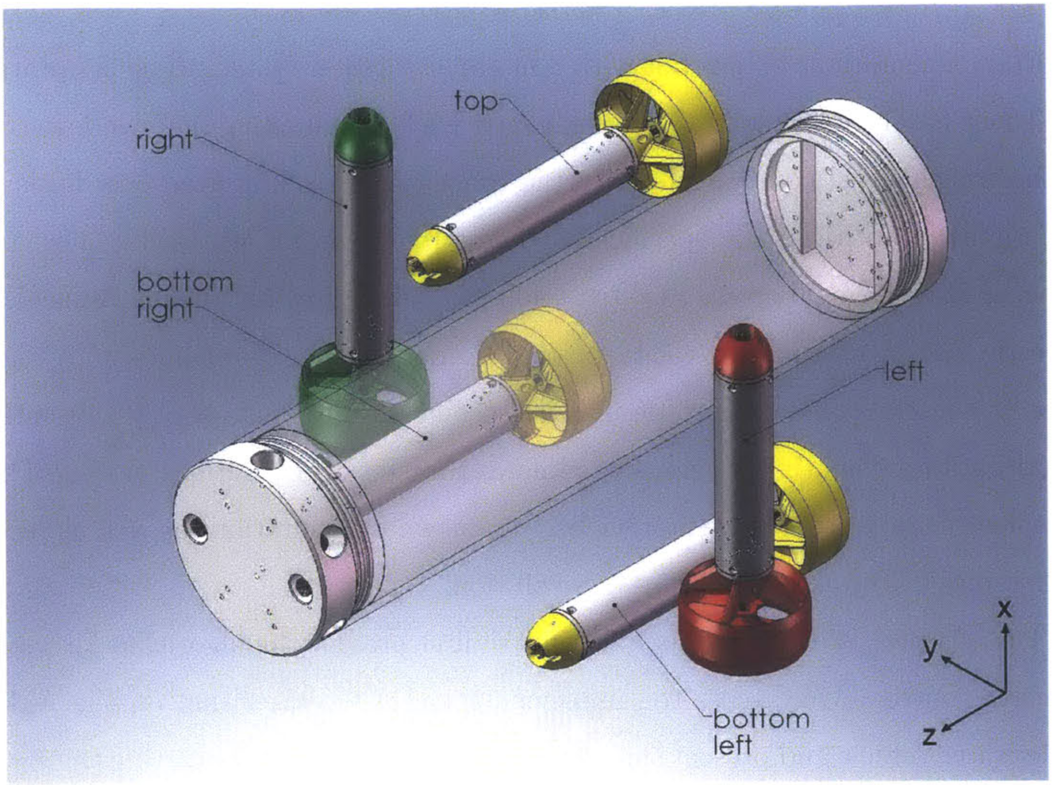


Figure 4-2: AMOUR's standard configuration with 5 thrusters. The thrusters can be arbitrarily mounted along the body. The body measures 17.8cm in diameter and 72.4cm in length. The thruster measure 4.8cm in diameter and 25.4cm in length. They drive a 9.4cm diameter and 9.6cm pitch propeller which is housed in a 10.1cm diameter nozzle.

magnetic field vector, $\mathbf{m} \in \mathcal{R}^3$. We assume that the acceleration vector is pointing towards the ground and that the magnetic field vector is pointing north and upwards. Both vectors $\mathbf{a}$ and $\mathbf{m}$ are measured in the robot's coordinate frame. Further, we assume the robot has a pressure sensor to measure the robot's depth, $d_{is} \in \mathcal{R}$. At sea level depth is assumed to be 0 m and as the robot dives the value d_{is} decreases.

The control algorithm presented in this chapter takes the current and desired robot attitude and depth as input and outputs speed commands for each thruster. The current state of the robot is computed from IMU measurements.

The goal attitude is provided as a rotation matrix and the goal depth is provided as a scalar:

$$\mathbf{R}_{goal} \in \mathcal{SO}^3, \quad d_{goal} \in \mathcal{R}. \tag{4.1}$$

The algorithm is configured by providing a list of thruster positions, $\mathbf{p}_i \in \mathcal{R}^3$, and orientations, $\mathbf{o}_i \in \mathcal{R}^3$, in the robot's coordinate frame for the N thrusters. Further, the proportional, integral, and derivative constants for the PID controller need to be provided. Adding or removing thrusters is as simple as updating the position and orientation vectors and specifying the PID loop.

The center of mass is assumed to be at the origin of the robot coordinate system in this chapter. However, the algorithm can easily be extended to incorporate a moving center of mass by simply subtracting the center of mass at the appropriate places. We assume the remaining dynamics of the robot, such as total mass and inertia or drag coefficients, to be unknown. Knowledge of these values would almost certainly improve the response rate and accuracy of control. However, because these values are very difficult to determine for a modular system such as AMOUR VI our controller is aimed at operating without the knowledge of these values.

The output of the main stage of the algorithm (the parts described in detail in this section) is an error for each thruster that is derived from the robot's rotational error and an error for each thruster that is derived from the robot's translational error.

4.2.2 Overview

The Modular Thruster Control Algorithm computes the rotational error in radians between the robot's current pose and desired pose. It also computes the axis around which this rotation should occur. Using a unit torque representation for all thrusters we solve for individual thruster errors that when summed will result in the total current rotational error. This is equivalent to projecting the robot's rotational error onto the thrusters.

Next, we compute the translational error given the thruster outputs using depth and position.[1] Using unit direction vectors for each thruster we solve for individual thruster errors that when multiplied with each thruster's direction will result in the total translational error. This is equivalent to projecting the robot's translational

[1] In the experiments presented in this chapter we did not have robot XY-position and so computed our translational error only based on depth.

error onto the thrusters.

The computed rotational and translational errors are each used as input to a separate PID controller on each thruster. Thus we can tune a thruster's response to translational and rotation errors independently. Finally, the output of both PID controllers is summed for each thruster to generate the thruster output.

To visualize why a separate rotational and translational PID controller is necessary for every thruster, consider the robot assembly shown in Figure 4-2. We remove the top and two bottom thrusters leaving only the left and right thruster and then try to roll the robot along its axis. We will encounter very little resistance as essentially no displacement of water occurs as the robot's body rolls. Thus we will only require a small P-gain for rotation control of the left and right thrusters. However, if we try to change depth with this two thruster configuration then we have to move the entire robot body broadside through the water. This will create a lot of drag and will require a far higher P-gain for the translational control of the two thrusters.

While we do not directly consider robot dynamics the use of two separate PID controllers for each thruster allows for a good approximation of simple robot dynamics. For example, in the above case we can tune the PID controllers derivative parameter to account for the momentum that occurs when the robot turns to prevent overshoot.

Our method is related to, but differs from Hanai *et al.* [64] in that they directly compute the required thrust necessary to be produced by each thruster. Hanai *et al.* assume that the voltage-to-thrust curve is known and drive each thruster directly from this curve. However, such a curve is static and will not accommodate changes in the vehicle's dynamics. Our method allows the user to quickly adjust the controller to such changes.

4.2.3 Computing the current robot state

Using the data provided by the IMU we first compute the robot's attitude by computing the east, north, and up vectors in its frame of reference and then generate the respective rotation matrix:

$$\begin{gathered}\mathbf{e} = \mathbf{a} \times \mathbf{m}, \quad \mathbf{n} = \mathbf{e} \times \mathbf{a} \\ \mathbf{R}_{is} = [\hat{\mathbf{e}}, \hat{\mathbf{n}}, -\hat{\mathbf{a}}]\end{gathered} \tag{4.2}$$

4.2.4 Computing the attitude error

Next we compute the rotation that will take the robot from its current attitude to the goal attitude ($\mathbf{R}_{is} \rightharpoonup \mathbf{R}_{goal}$). This rotation has to be represented in the robot's local coordinate frame and can be computed as:

$$\mathbf{R}_{cmd} = (\mathbf{R}_{goal}^{-1} \cdot \mathbf{R}_{is})^{-1} \tag{4.3}$$

We then convert this rotation matrix into a desired rotation angle and axis:

$$\omega = \arccos((\mathrm{Trace}(\mathbf{R}_{cmd} - 1)/2) \tag{4.4}$$

$$\mathbf{r} = \frac{1}{2 \cdot \sin(\omega)} \cdot \begin{bmatrix} \mathbf{R}_{cmd,32} - \mathbf{R}_{cmd,23} \\ \mathbf{R}_{cmd,13} - \mathbf{R}_{cmd,31} \\ \mathbf{R}_{cmd,21} - \mathbf{R}_{cmd,12} \end{bmatrix} \tag{4.5}$$

where $\mathbf{R}_{cmd,xy}$ represents entry (x, y) of $\mathbf{R}_{cmd}$. Note that when $\sin(\omega) = 0$ a singularity occurs and we have to treat these cases separately to avoid the singularity. There are two cases for which $\sin(\omega) = 0$:

1. The first case occurs when $\omega = 0$ and implies $\mathrm{Trace}((\mathbf{R}_{cmd} - 1)/2) = 1$. Because $\mathbf{R}_{cmd}$ is an orthogonal matrix this only happens when $\mathbf{R}_{cmd}$ is the identity matrix and thus $\mathbf{R}_{is} = \mathbf{R}_{goal}$. This implies that the robot's rotational error is zero. In this case we can set $\omega = 0$ and $\mathbf{r} = \mathbf{0}$.

2. The second case occurs when $\omega = \pi$. In this case there is a sign ambiguity for the axis as rotation around $\mathbf{r}$ by and angle π clockwise has the same result as rotation around $\mathbf{r}$ by and angle π counter clockwise. Because rotation preserves vectors which are parallel to the axis of rotation we have $\mathbf{R}_{cmd} \cdot \mathbf{r} = \mathbf{r}$. That

means that $\mathbf{r}$ is an eigenvector of $\mathbf{R}_{cmd}$ with eigenvalue 1 and can be computed by looking for the eigenvalues and eigenvectors of $\mathbf{R}_{cmd}$.

Next we compute the depth error in meters and the torque error vector. The direction of the torque error vector represents the torque axis of the error, while the length gives the amplitude of the error in radians:

$$e_{robot,depth} = d_{goal} - d_{is} \tag{4.6}$$

$$\mathbf{e}_{robot,rot} = \omega \cdot \mathbf{r} \tag{4.7}$$

4.2.5 Computing separate thruster errors for rotation

To solve for rotation we compute the torque vector for every thruster:

$$\mathbf{t}_i = \mathbf{p}_i \times \mathbf{o}_i, \quad i \in 1 \ldots N. \tag{4.8}$$

We then concatenate all torque vectors to form an equation system. Extra user constraints, like the definition of thruster symmetries, can be added as extra lines $\mathbf{C}_{rot}$ to the equation system:

$$\mathbf{A}_{rot} = \begin{bmatrix} \mathbf{t}_1 & \ldots & \mathbf{t}_N \\ & \mathbf{C}_{rot} & \end{bmatrix} \tag{4.9}$$

To finally compute the error for every thruster we use linear least squares approximation. The advantage of this method is that if a perfect solution does not exist, for instance if the robot does not have enough degrees of freedom (DOF), then we get the best possible solution. To compute the result we first compute the Moore-Penrose pseudo-inverse:

$$\mathbf{A}^{+}_{rot} = \mathbf{A}^{T}_{rot} \cdot (\mathbf{A}_{rot} \cdot \mathbf{A}^{T}_{rot})^{-1} \tag{4.10}$$

and use it to solve for thruster errors. Because we added constrains $\mathbf{C}_{rot}$ to $\mathbf{A}_{rot}$ we need to pad $\mathbf{e}_{robot,rot}$ with zeros equal to the number of rows in $\mathbf{C}_{rot}$:

$$\mathbf{E}_{thrusters,rot} = \mathbf{A}^{+}_{rot} \cdot \begin{bmatrix} \mathbf{e}_{robot,rot} \\ 0 \\ \vdots \\ 0 \end{bmatrix} . \tag{4.11}$$

Note that because of thruster redundancy it is likely that infinitely many solutions exist. Adding constraints to $\mathbf{A}_{rot}$ are one way of choosing the right solution. Alternatively, as in [64] a specific solution can be chosen (for example to minimize the thruster output while avoiding thruster dead bands). We have chosen the constraint adding solution because in our experiences thruster dead bands did not affect the performance of our robot.

4.2.6 Computing separate thruster errors for translation

We only use the depth error $e_{robot,depth}$ to computer the translational error $\mathbf{e}_{robot,trans}$. However, provided position information for the robot an error in the XY plane can be computed and added to $\mathbf{e}_{robot,trans}$.

$$\mathbf{e}_{robot,trans} = [0 \quad 0 \quad e_{robot,depth}]^T \tag{4.12}$$

Using the concatenation idea that allowed us to create a system of equations from the torque vectors to solve for rotation, we can also concatenate all unit orientation vectors:

$$\mathbf{A}_{trans} = \begin{bmatrix} \mathbf{o}_1 & \dots & \mathbf{o}_N \\ & \mathbf{C}_{trans} & \end{bmatrix} \tag{4.13}$$

Again we use least square approximation to solve for thruster errors and remember to pad $\mathbf{e}_{robot,trans}$ with zeros equal to the number of rows in $\mathbf{C}_{trans}$:

$$\mathbf{A}_{trans}^{+} = \mathbf{A}_{trans}^{T} \cdot (\mathbf{A}_{trans} \cdot \mathbf{A}_{trans}^{T})^{-1} \tag{4.14}$$

$$\mathbf{E}_{thrusters,trans} = \mathbf{A}_{trans}^{+} \cdot \begin{bmatrix} \mathbf{e}_{robot,trans} \\ 0 \\ \vdots \\ 0 \end{bmatrix} . \tag{4.15}$$

4.2.7 Combined Error Computation

We can combine the translational and rotational error projection matrices from Eq. 4.10 and Eq. 4.14 into a combined projection matrix

$$\mathbf{A}^{+} = \begin{bmatrix} \mathbf{A}_{trans}^{+} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{rot}^{+} \end{bmatrix} , \tag{4.16}$$

that projects the translational and rotational robot errors onto the thrusters:

$$\mathbf{E} = \begin{bmatrix} \mathbf{E}_{thrusters,trans} \\ \mathbf{E}_{thrusters,rot} \end{bmatrix} . = \begin{bmatrix} E_1 \\ \vdots \\ E_{2N} \end{bmatrix} = \mathbf{A}^{+} \cdot \begin{bmatrix} \mathbf{e}_{robot,trans} \\ \mathbf{e}_{robot,rot} \end{bmatrix} . \tag{4.17}$$

Note that unless the number or position of thrusters changes (or if the constraints change) then $\mathbf{A}_{rot}$ and $\mathbf{A}_{trans}$ are constant and do not need to be computed during every update step.

A summary of the Modular Thruster Control Algorithm (excluding the PID control loops for clarity) is given above in Algorithm 1.

4.2.8 PID Control

We described how we can compute a rotational and a translational error for every

Algorithm 1 Modular Thruster Control Algorithm

```
$\mathbf{A}^+_{rot} = \mathbf{A}^T_{rot} \cdot (\mathbf{A}_{rot} \cdot \mathbf{A}^T_{rot})^{-1}$
$\mathbf{A}^+_{trans} = \mathbf{A}^T_{trans} \cdot (\mathbf{A}_{trans} \cdot \mathbf{A}^T_{trans})^{-1}$
loop
    $\mathbf{R}_{is} = [\hat{\mathbf{e}}, \hat{\mathbf{n}}, -\hat{\mathbf{a}}]$
    $\mathbf{R}_{cmd} = (\mathbf{R}^{-1}_{goal} \cdot \mathbf{R}_{is})^{-1}$
    $\mathbf{r} = axis(R_{cmd}), \quad w = angle(R_{cmd})$
    $\mathbf{E}_{thr.,rot} = \mathbf{A}^+_{rot} \cdot [\omega \cdot \mathbf{r}^T \quad 0 \dots 0]^T$
    $\mathbf{E}_{thr.,trans} = \mathbf{A}^+_{trans} \cdot [0 \quad 0 \quad (d_{goal} - d_{is}) \quad 0 \dots 0]^T$
end loop
```

thruster. Each of these errors is used as input to a PID controller and the output of these controllers is added to produce a thruster output.

For N thrusters there are a total of $2N$ PID controllers. Each controller has 3 tunable gains: the proportional gain K_p, integral gain K_i, and differential gain K_d. We represent all gains using 6 diagonal matrices, where the j-th diagonal entry of each K matrix is the appropriate gain for the j-th thruster:

- $\mathbf{K}_{R,p}$, $\mathbf{K}_{R,i}$, and $\mathbf{K}_{R,d}$ for the rotational error PID controllers and
- $\mathbf{K}_{T,p}$, $\mathbf{K}_{T,i}$, and $\mathbf{K}_{T,d}$ for the translational error PID controllers.

We represent the state of the PID controllers as a vector

$$\mathbf{x}_t = \begin{bmatrix} p_1 \dots p_{2N} & i_1 \dots i_{2N} & d_1 \dots d_{2N} \end{bmatrix}^T. \tag{4.18}$$

where

- p_j, i_j, and d_j for $j \in \{1 \dots N\}$ are the proportional, integral, and differential values of the j-th rotational error PID controller and
- p_{N+j}, i_{N+j}, and d_{N+j} for $j \in \{1 \dots N\}$ are the proportional, integral, and differential values of the j-th rotational error PID controller and

We assume the time step *Deltat* between two successive updates of the PID states. Using the above notation and the state transition can then be expressed as

$$\mathbf{x}_{t+1} = \begin{bmatrix} \mathbf{I}_{2N\times 2N} \\ \frac{1}{\Delta t}\cdot \mathbf{I}_{2N\times 2N} \\ \Delta t \cdot \mathbf{I}_{2N\times 2N} \end{bmatrix} \cdot \mathbf{E}_{t+1} + \begin{bmatrix} \mathbf{0}_{2N\times 2N} & \mathbf{0}_{2N\times 2N} & \mathbf{0}_{2N\times 2N} \\ -\frac{1}{\Delta t}\cdot \mathbf{I}_{2N\times 2N} & \mathbf{0}_{2N\times 2N} & \mathbf{0}_{2N\times 2N} \\ \mathbf{0}_{2N\times 2N} & \mathbf{0}_{2N\times 2N} & \mathbf{I}_{2N\times 2N} \end{bmatrix} \cdot \mathbf{x}_t \qquad (4.19)$$

and the output of our controller then is defined as

$$\mathbf{y}_t = \begin{bmatrix} \mathbf{K}_{R,p} & \mathbf{K}_{T,p} & \mathbf{K}_{R,i} & \mathbf{K}_{T,i} & \mathbf{K}_{T,d} & \mathbf{K}_{T,d} \end{bmatrix} \cdot \mathbf{x}_t. \qquad (4.20)$$

where the j-th entry of $\mathbf{y}_t$ is the control signal to thruster j.

It is worth noting that when using only proportional control ($\mathbf{K}_{R,i} = \mathbf{K}_{R,d} = \mathbf{K}_{T,i} = \mathbf{K}_{T,d} = \mathbf{0}$) the above system of equations can be summarized as

$$\begin{aligned} \mathbf{y}_t &= \begin{bmatrix} \mathbf{K}_{R,p} & \mathbf{K}_{T,p} \end{bmatrix} \cdot \mathbf{A}^{+} \cdot \begin{bmatrix} \mathbf{e}_{robot,trans} \\ \mathbf{e}_{robot,rot} \end{bmatrix} && (4.21) \\ &= \begin{bmatrix} \mathbf{K}_{R,p} \cdot \mathbf{A}^{+}_{rot} & \mathbf{K}_{T,p} \cdot \mathbf{A}^{+}_{trans} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{e}_{robot,trans} \\ \mathbf{e}_{robot,rot} \end{bmatrix}. && (4.22) \end{aligned}$$

At this point it is important to note, that if the robot is fully actuated (i.e. the thrusters are configured such that control in six degrees of freedom is possible), then $\text{rank}(\begin{bmatrix} \mathbf{A}^{+}_{rot} & \mathbf{A}^{+}_{trans} \end{bmatrix}) = 6$. Keeping in mind that $\mathbf{K}_{R,p}$ and $\mathbf{K}_{T,p}$ are diagonal and with the assumption that their entries are non-zero this implies that for a fully actuated robot we have $\text{kern}(\begin{bmatrix} \mathbf{A}^{+}_{rot} & \mathbf{A}^{+}_{trans} \end{bmatrix}) = 0$ and all non-zero errors are mapped to non-zero thruster commands.

We can relax the constraint $\mathbf{K}_{R,i} = \mathbf{K}_{R,d} = \mathbf{K}_{T,i} = \mathbf{K}_{T,d} = \mathbf{0}$ if we assume that the errors $\mathbf{e}_{robot,trans}$ and $\mathbf{e}_{robot,rot}$ are constant. In this case we can modify Eq. 4.19 to get:

$$\mathbf{x}_{t+1} = \begin{bmatrix} \mathbf{I}_{2N\times 2N} & \mathbf{0}_{2N\times 2N} & \mathbf{0}_{2N\times 2N} \\ \mathbf{0}_{2N\times 2N} & \mathbf{0}_{2N\times 2N} & \mathbf{0}_{2N\times 2N} \\ \Delta t \cdot \mathbf{I}_{2N\times 2N} & \mathbf{0}_{2N\times 2N} & \mathbf{I}_{2N\times 2N} \end{bmatrix} \cdot \mathbf{x}_t. \tag{4.23}$$

Under the assumption that the robot is fully actuated we can show that x_t cannot attain a steady zero-state for a nonzero error. The only exception to this is a degenerate case when we bound the integral terms.

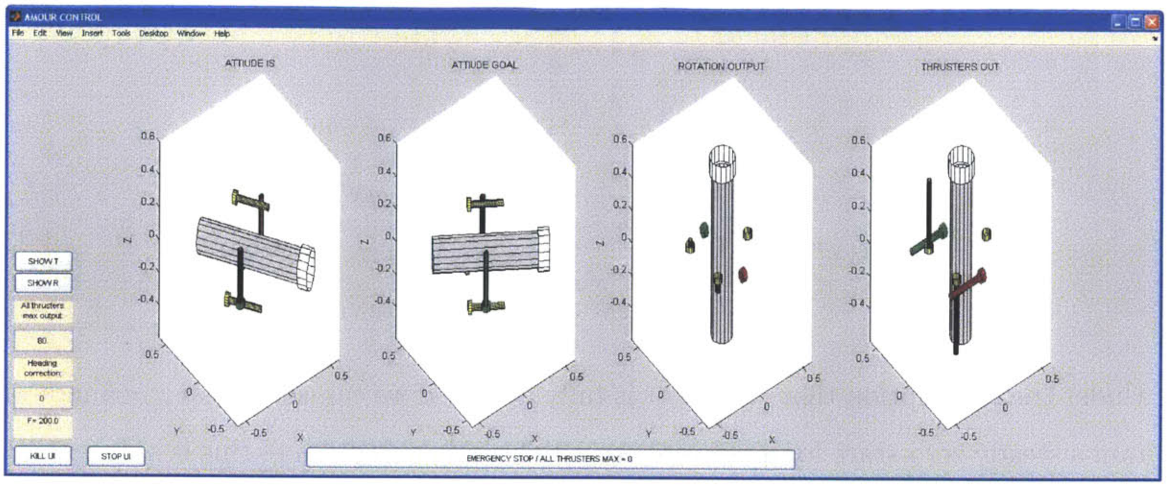
4.2.9 PID controller tuning

The tuning of the PID controllers can be performed manually or automatically with both methods described in [9, 10, 127, 126]. In this thesis PID controller tuning was performed manually by following these steps:

1. Set all three gains K_p, K_i, and K_d to 0.
2. Keep increasing the proportional gain K_p until the controller starts to overshoot by 5-10 %.
3. Increase the derivative gain K_d to correct for the overshoot.
4. Increase the integral gain K_i to minimize offset errors but without causing instability.

4.2.10 Thruster failures

It is worth mentioning that thruster failures can be handled by removing the appropriate line from $\mathbf{A}_{rot}$ and $\mathbf{A}_{trans}$. Failure can be detected in several ways, for example the thruster not responding to communication requests. We also envision a system that actively can detect changes in thruster response by using a disturbance observer (DOB) similar to [125].



All thrusters max output: 80

Heading correction: 0

F= 200.0

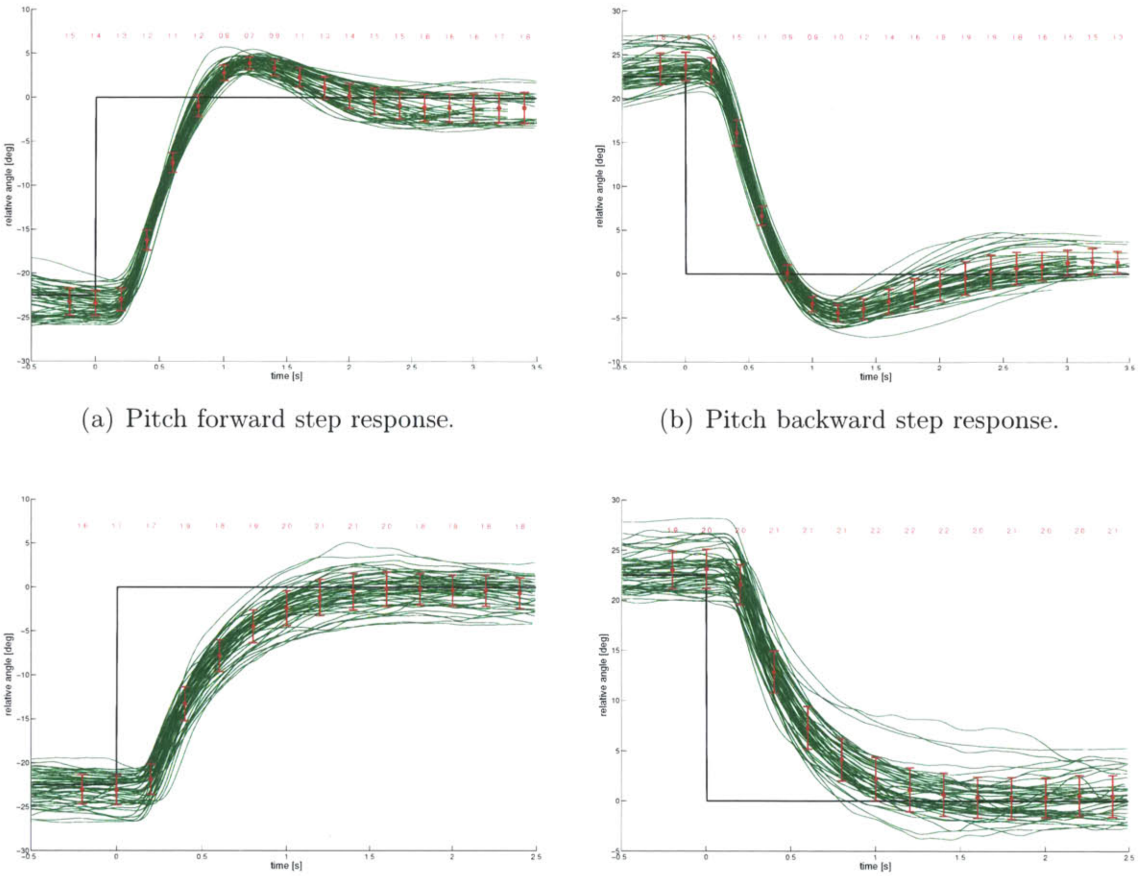
Figure 4-3: AMOUR's user interface. The UI can run on a laptop connected to the robot through a serial cable. The UI provides visual feedback and allows for easy on-the-fly tuning of parameters.

4.2.11 Experimental Evaluation

We evaluated the performance of the Modular Thruster Control Algorithm during a set of trials performed at the MIT Alumni swimming pool. The pool depth varies between 2 and 3 meters. We typically operated the robot at a depth of 1 meter. For the first set of experiments the robot was fitted with 5 thrusters as shown in Figure 4-2. At one point during the experiment one thruster failed. We were able to replace the thruster and resume operations in under five minutes. In the second set of experiments we added a sixth thruster as shown in Figure 3-1. We trimmed the buoyancy and the balance of the robot by adding stainless steel washers to 4 rods distributed around the robot's body. After trimming the robot was 50g negatively buoyant and almost neutrally balanced.

We then tuned the translation and rotation PID controllers for each of the five thrusters according to the method presented in Section 4.2.9. As described in Section 4.2, we need two different controllers for the translation and rotation to account for the fact that, for example, in the roll case there is very little drag, however, there is a lot of drag during depth translation.

We were able to take advantage of the symmetries of the robot to reduce the actual number of PID loops that we needed to tune to four (the two vertical and three



(a) Pitch forward step response.

(b) Pitch backward step response.

(c) Roll counter-clockwise step response.

(d) Roll clockwise step response.

Figure 4-4: Experimental results for pitch and roll step response. The robot was commanded to performed 48 discrete 22.5 degree turns (three full rotations) around both the pitch and roll axes. The black lines represent the step command and the red lines represent AMOUR's response. In the pitch direction the robot settles within 1.5 degrees (1.8 degree standard deviation) of the commanded angle in an average of 1.9 seconds. In the roll direction the robot settles within 1 degree (2.0 degree standard deviation) of the commanded angle in an average of 1.4 seconds.

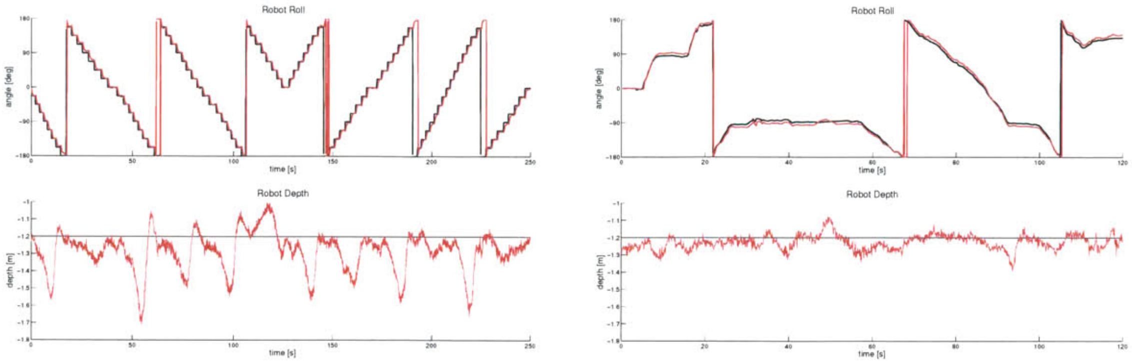
horizontal thrusters have the same parameters). First, we stabilized the rotational controllers and then we tuned the translation controller. In our setup we did not have an XY-positioning system, so our translation controller just dealt with depth control. When we added a sixth thruster we were able to keep all of the parameters for the other thrusters the same and just tuned the parameters for the new thruster.

Our general approach to tuning the PID controllers was to increase the proportional term until they responded quickly and had small overshoots and oscillations. Then we added a derivative component to dampen the oscillations. Finally, an integral component can be added to compensate for any constant offsets, however, we did not find this necessary in these trials as the robot was well balanced. Tuning all of the parameters in this experiment took under 15 minutes and only needed to be done once.

Retuning individual thruster PID controllers is not necessary unless the position of the thrusters changes significantly, which never occurred during our experiments. In our experience rotating the thruster by as much as 10 degrees did not visibly affect controller performance. It should also be noted that the thrusters were attached by hand and their positions were only estimated as opposed to measured. This shows the robustness of our controller to inaccuracies.

We performed a number of experiments to validate the control algorithm. In the first experiment we examined the step response of the controller to changes in orientation. The robot was configured with five thrusters. For both clockwise and counterclockwise pitch and roll we commanded 48 discrete 22.5 degree turns (three full turns). The results of these experiments are shown in Figure 4-4. In the pitch direction the robot settles within 1.5 degrees of the commanded angle in an average of 1.9 seconds. In the roll direction the robot settles to within 1 degree in an average of 1.4 seconds. While we do not have room to report on the yaw controller, other experiments using the yaw controller yield similar results.

The change in the commanded angle is illustrated by the black square signal in the figure. Note that there is a 160 millisecond delay from the command changing to



(a) Depth control at various roll angles, using 5 thrusters.

(b) Depth control at various roll angles, using 6 thrusters.

Figure 4-5: This graph shows the performance of the control algorithm in two configurations: (a) 5 thrusters and (b) 6 thrusters. The top graphs show the roll angle and the bottom graphs show the depth. The black lines plot the roll and depth commands and the red lines plot the actual roll and depth. When using 5 thrusters, the robot is not fully holonomic and cannot control depth when rolled 90 degrees. At this angle the robot sinks due to its negative buoyancy. When a sixth thruster is attached the robot become approximately holonomic. In this configuration the robot can control its depth in any orientation.

the robot responding. We suspect that this delay is caused by our user interface. The pitch controller initially overshoots but then quickly settles to the target angle. The roll controller does not overshoot. Impulses given to the robot by a human swimmer resulted in similar quick settling times.

Figure 4-5(a) shows the depth of the robot as the robot rolled with five thrusters. In this configuration the robot can only control five degrees of freedom, the sixth is uncontrolled. When the robot is on its side it does not have any thrusters to compensate for the slight negative buoyancy. Thus, the robot starts to sink in that orientation, as seen in Figure 4-5(a). In this situation the controller alerts the user that the desired command cannot be executed precisely and outputs the type and magnitude of the error.

To make the robot approximately holonomic we added a sixth thruster to the nose of the robot in a horizontal configuration as shown in Figure 3-1. The robot is then able to maintain depth when it is rotated on its side. However, in doing so the bow thruster exerts a torque on the robot that the control algorithm compensates for by using the other thrusters. Figure 4-5(b) shows the depth of the robot as the robot rolls with six thrusters. This plot shows that we are able to maintain depth control of the robot even when it is on its side. In this experiment the robot was controlled using the human input device, keeping the robot on its side most of the time where it is most difficult to maintain depth.

The ability of the robot to track the desired trajectory of the human input device can be seen in the top portion of Figure 4-5(b). The black line indicates the target orientation based on the input device and the red line is the actual orientation of the robot. Most of the time these lines are completely overlapping indicating that the robot achieves the desired orientation. Multiple users operated the robot in the pool environment with the input device. All users reported that robot could be easily and intuitively controlled. Complex rolls and orientations can easily be achieved due to the high update rate of the input device and the quick response time of the robot.

4.3 Estimation of Thruster Configurations

4.3.1 Estimation Algorithm

In this section we describe the algorithm used for thruster parameter learning. We present assumptions, input and output values, and finally the algorithm itself.

Algorithm 2 for estimating the thruster configuration of an underwater modular robot makes the following assumptions: (1) the robot is capable of determining its own attitude by measuring the acceleration vector, $\mathbf{a} \in \mathcal{R}^3$, and the magnetic field vector, $\mathbf{m} \in \mathcal{R}^3$; (2) the number of thrusters N is known but their location is not known; (3) each thruster is rigidly mounted to the robot at some unknown position with an unknown orientation of thrust; (4) thrusters take a command between -100 and $+100$ (full thrust backward/full thrust forward), 0 means the thruster is off. The output of Algorithm 2 is an inverse model of the thrusters, i.e. a mapping from a desired rotation and position change of the robot to thruster commands.

While the robot is at rest, the algorithm measures σ_{acc} and σ_{rot}, the acceleration noise levels and the changes in rotation of the robot. Estimation is performed by pulsing individual thrusters and recording the measured rotational changes *rot* and accelerations *acc* of the robot together with the thruster output. Each thruster is pulsed for 1 second with a settling time of 3 seconds in between pulses.

Data samples are only chosen during pulse time. Samples are selected when they show a change of at least k times larger than the noise level of the corresponding signal (σ_{acc} and σ_{rot}). In our implementations we use $k = 3$. Initially a thrust level of 50% maximum thrust is selected as the pulse strength. However, the flags $flag_{rot}$ and $flag_{acc}$ ensure, that if no samples are gathered during a pulse because the accelerations and changes to rotation are smaller than k times the noise levels, then the thrust level is increased by 10%.

After a sufficient number of pulses (currently determined manually by an observer) we compute a least squares solution to the linear equation system $\mathbf{S} \cdot \mathbf{A} = \mathbf{T}$ for both accelerations and rotational changes. This is done by computing the Moore-Penrose

Algorithm 2 Thruster Configuration Estimation Algorithm

```
$thrust[1 \dots N] \Leftarrow 0$,   $power \Leftarrow 0.5$
measure IMU noise levels $\sigma_{acc}$ and $\sigma_{rot}$
while learning do
  PICK $I \in 2^N$
  $flag_{acc} \Leftarrow 1$, $flag_{rot} \Leftarrow 1$
  for 1 sec do
    $thrust[I] \Leftarrow power$
    if $||acc|| > k \cdot \sigma_{acc}$ then
      APPEND $(acc, thrust)$ TO $(\mathbf{S}_{acc}, \mathbf{T}_{acc})$
      $flag_{acc} \Leftarrow 0$
    end if
    if $||rot|| > k \cdot \sigma_{rot}$ then
      APPEND $(rot, thrust)$ TO $(\mathbf{S}_{rot}, \mathbf{T}_{rot})$
      $flag_{rot} \Leftarrow 0$
    end if
  end for
  if $flag_{acc}$ or $flag_{rot}$ then
    $power \Leftarrow \max(power + 0.1, 1.0)$
  end if
  $thrust[i] \Leftarrow 0$,   PAUSE(3 sec)
end while
$\mathbf{A}_{rot} = \mathbf{S}_{rot}^{+} \cdot \mathbf{T}_{rot}$,   $\mathbf{A}_{trans} = \mathbf{S}_{acc}^{+} \cdot \mathbf{T}_{acc}$
```

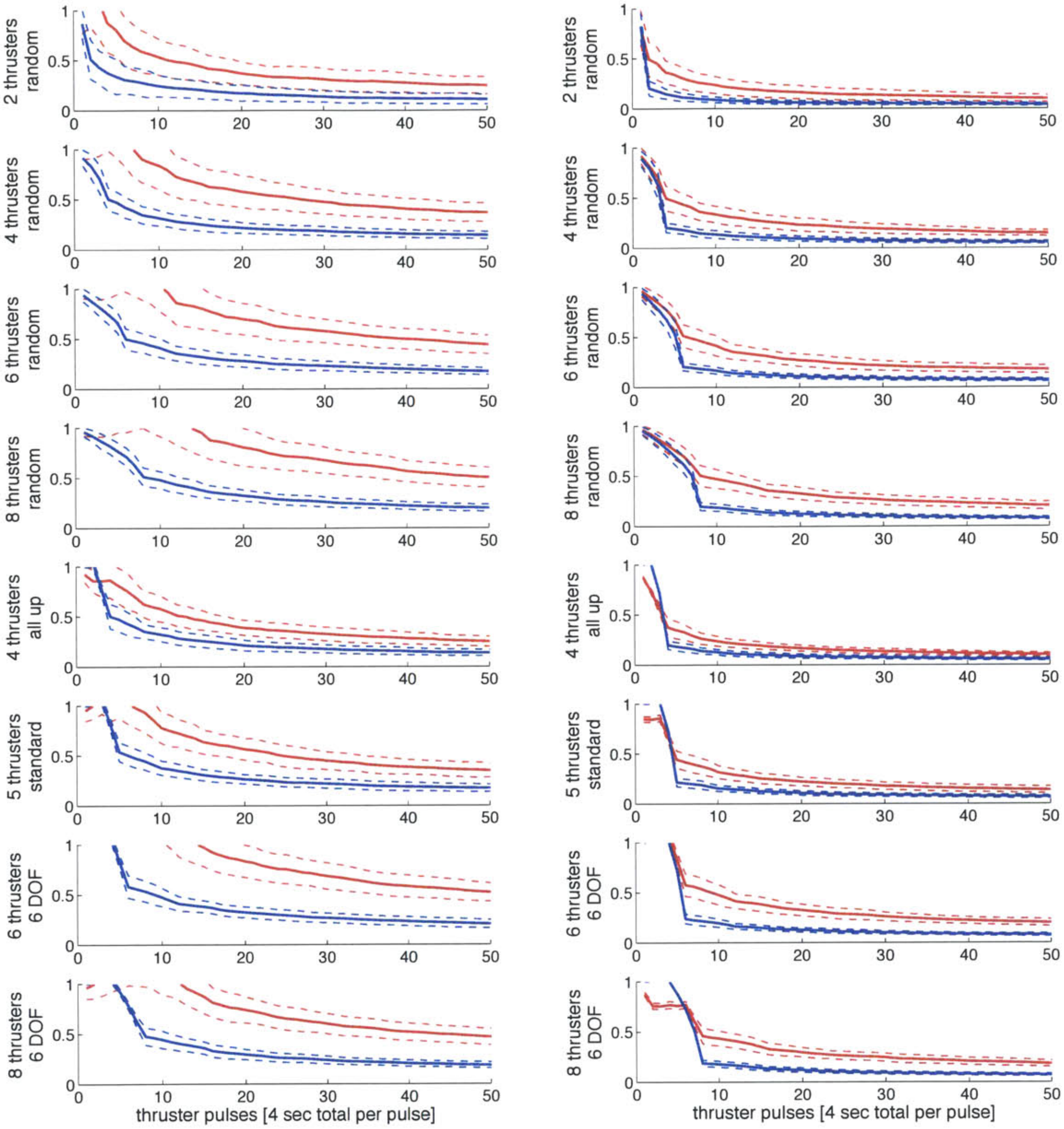
pseudo-inverse for the matrix **S**. The matrices $\mathbf{A}_{acc}$ and $\mathbf{A}_{rot}$ represent inverse thruster models of the underwater vehicle. These matrices can be used to project the desired changes in attitude and position onto the robots thrusters as is described in the first part of this chapter.

4.3.2 Simulation Evaluation

We studied the thruster configuration estimation algorithm in simulation to verify the correctness of the estimation and its performance. We developed a model of the robot and considered configurations with 2, 4, 6, and 8 thrusters. In each case we selected some fixed thruster configurations to experiment with. We also ran the thruster estimation algorithm in segments of 100 runs for each thruster number (2, 4, 6, 8). For each of the 100 runs we generated randomly the thruster configuration. We measured the difference of the learned thruster configuration from the true thruster configuration as the main evaluation metric.

More specifically for each simulation we first chose the thruster configuration at random by sampling the effect of each thruster on the vehicles rotation and acceleration from a uniform distribution. The simulation parameters were selected as follows. We chose the maximum possible rotation speed of the vehicle caused by a single thruster to be 5 rad/s at full thrust. (By comparison we recorded rotation speeds of 2.5 rad/s during rolls effected by a single thruster commanded to operate at half thrust on AMOUR.) We choose the maximum possible acceleration of the vehicle caused by a single thruster to be 0.2 g. (This value was also chosen in accordance with accelerations measured on AMOUR.) For a simulation of a vehicle with N thrusters this resulted in two matrices that described the rotation and acceleration contributions of each thruster:

$$R \in \mathcal{R}^{3\times N} | \text{ s.t. } ||R_{i,1..3}|| \leq 0.05 \text{ and } M_{acc} \in \mathcal{R}^{3\times N} \text{ s.t. } ||R_{i,1..3}|| \leq 0.02$$



(a) pulses at 20% power ($p = 20$)

(b) pulses at 50% power ($p = 50$)

Figure 4-6: Thruster estimation algorithm simulation results for 4 randomly drawn thruster configurations (4 top rows) and 4 manually defined thruster configurations (4 bottom rows). The x axes denote the number of thruster pulses used so far. Each pulse is 1 s long with a 3 s pause following it for a total of 4 s per thruster pulse. The y axes denote the error, 0 meaning a perfect estimate and values above 0 indicating deviation from the ground truth as described in Section 4.3.2. The blue solid line shows the error of the estimation of the rotation matrix and the red solid line shows the error of the estimation of the acceleration matrix. All values are averaged from 100 experiment runs. The dashed lines represent error bounds (1 standard deviation).

During simulation the thrusters were pulsed individually at a manually defined thrust which corresponded to thrust values of $-p$ and $+p$. We performed two sets of simulations, one with $p = 20$ and one with $p = 50$. We pulsed thrusters in order, each first in one direction and then in the other. After all thrusters have been pulsed the simulation continued again with the first thruster and so on. This was done for a total of 50 pulses during each simulation.

For every pulse a thrust vector $T \in \mathcal{R}^N$ with $T_i = \pm p$ where i corresponded to the pulsed thruster and $T_i = 0$ for all other thrusters. We computed the resultant rotation and acceleration by multiplying the rotation matrix R and the acceleration matrix A with the thrust vector T. The computed rotation and acceleration were corrupted by adding a noise value drawn from a normal distribution with mean 0 and standard deviation 0.25 for rotation and standard deviation 0.05 for acceleration. This corresponds to 0.25 rad/s error for rotations and 0.05 g error for acceleration. We determined both values experimentally using AMOUR. The noise represented the average noise over all samples collected during a real world experiment.

For every simulated thruster pulse we store the thrust vector and the rotation and acceleration measurements in the same manner as presented in Algorithm 2. After each pulse we computed a new estimate of the rotation and acceleration matrices R_{est} and A_{est} using the data collected so far. We defined the error as

$$E_R = \frac{||R_{est} - R||}{||R||} \text{ and } E_A = \frac{A_{est} - A}{||A||}.$$

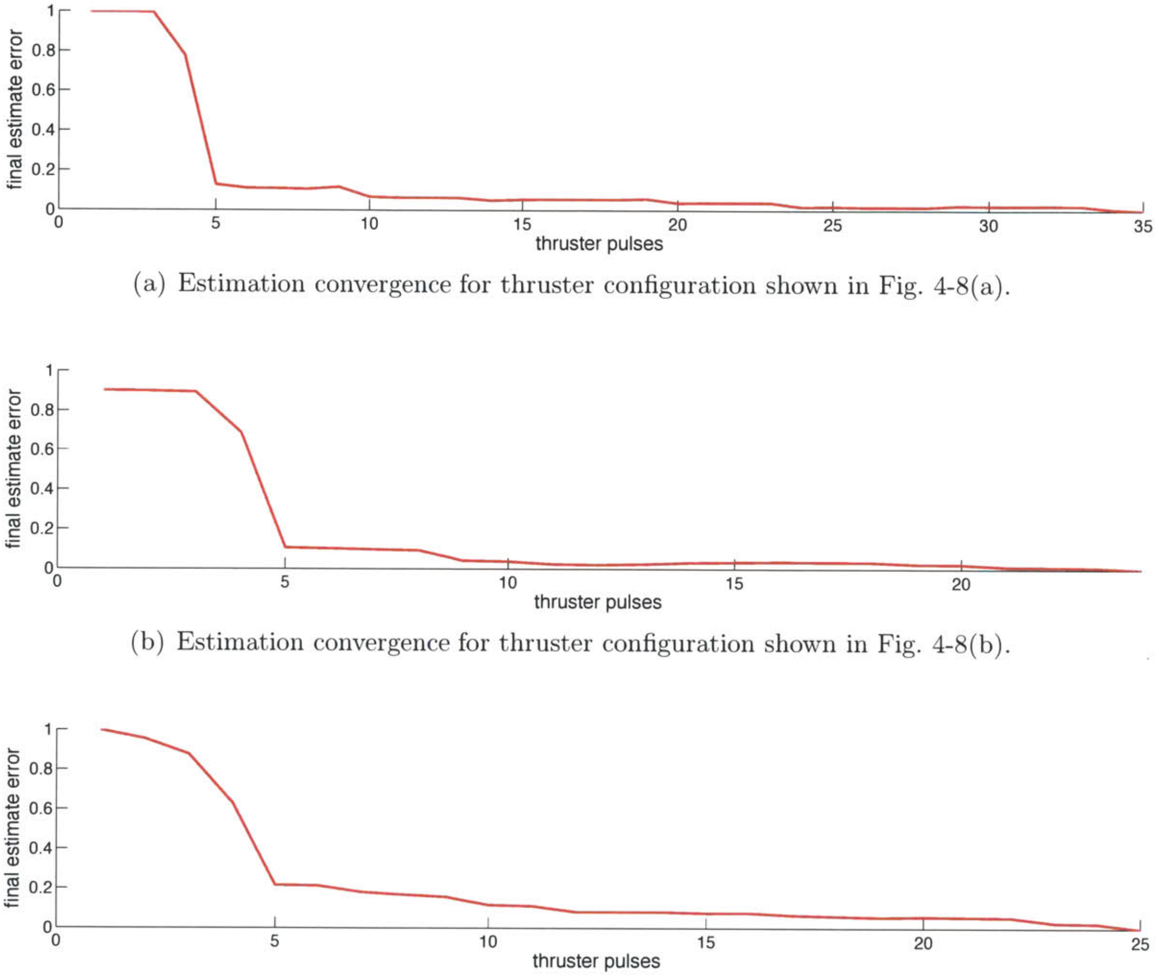
For each thruster number (2,4,6,8) we ran 100 simulations. Figure 4-6 shows simulation data for the aggregated set of 100 simulations for $k = 2, 4, 6, 8$ thrusters as well as for 4 manually defined configurations. These manual configuration are: (1) 4 thrusters with 2 on each side of the robot, all oriented upwards (2) 5 thrusters in the default configuration as seen in Fig 4-8(a), (3) 6 thrusters oriented to achieve 6 degrees of freedom (DOF), and (4) 8 thrusters oriented to achieve 6 DOF. The left column of graphs shows simulation results for $p = 20$ and the right column for $p = 50$. The x axes denote the number of thruster pulses used so far. Since each pulse is 1 s long with a 3 s pause following it this corresponds to 4 s per pulse. The y axes denote the error. The solid lines show the error after a given number of pulses averaged over all 100 simulations. The dashed lines mark the error bounds (1 standard deviation).

The first 4 rows show a gradual decrease in performance. This illustrates that with an increasing number of thrusters the algorithm takes a linearly increasing time to achieve the same estimation accuracy. This is grounded in the fact that the thrusters are pulsed round-robin and so to pulse each thruster a times takes $n \cdot a$ pulses for n thrusters.

Because the noise is mainly a function of the accelerometer and magnetometer sensor we chose it to be independent of the actual rotational or translational speed of the vehicle. A direct result is that simulations will converge quicker when thrusters are commanded with higher values (faster speed). This behavior is directly visible when comparing the performance of the estimation algorithm in Fig 4-6(a) ($p = 20$) with the performance in Fig 4-6(b) ($p = 50$). All simulations shown assume the same amount of noise, but the estimation accuracy converges much faster to a small value when pulse with a higher thrust are used ($p = 50$). The same behavior was observed also experimentally.

4.3.3 Experimental Evaluation

In addition to the simulation studies, we also conducted a set of hardware experiments in the pool using our underwater robot AMOUR VI. Three different experiments were conducted to evaluate the performance of Algorithm 2. For each experiment we first estimated the robot's thruster configuration using Algorithm 2 and then used the produced estimate to control the robot with Algorithm 1. We evaluate the goodness of the estimated configuration by how well Algorithm 1 can execute attitude commands. This is neccssary, as we do not have accurate group truth for the thruster configuration, since the thrusters are manually mounted and their location and orientation is hard to measure precisely. The three configurations used during

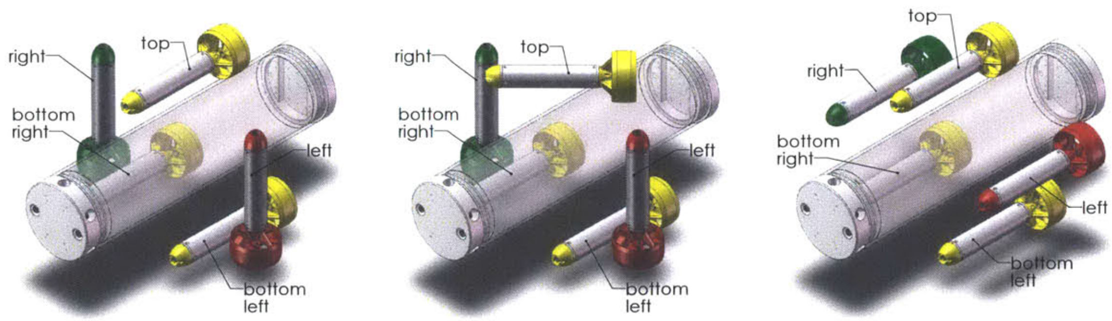


(a) Estimation convergence for thruster configuration shown in Fig. 4-8(a).

(b) Estimation convergence for thruster configuration shown in Fig. 4-8(b).

(c) Estimation convergence for thruster configuration shown in Fig. 4-8(c).

Figure 4-7: The convergence of the thruster rotation matrix estimation computed by the estimation algorithm from experimental data. The x axis denotes the number of thruster pulses (for each pulse multiple data points were collected). The y axis denotes the error. Each graph displays the error of estimate E_t when compared to the final estimate E_f for that particular experiment. The error is defined as $error = \frac{||E_t - E_f||}{||E_f||}$. The final estimate was used to compute the error because it was impractical to measure ground truth.



(a) default (b) top rotated 45° (c) all aligned

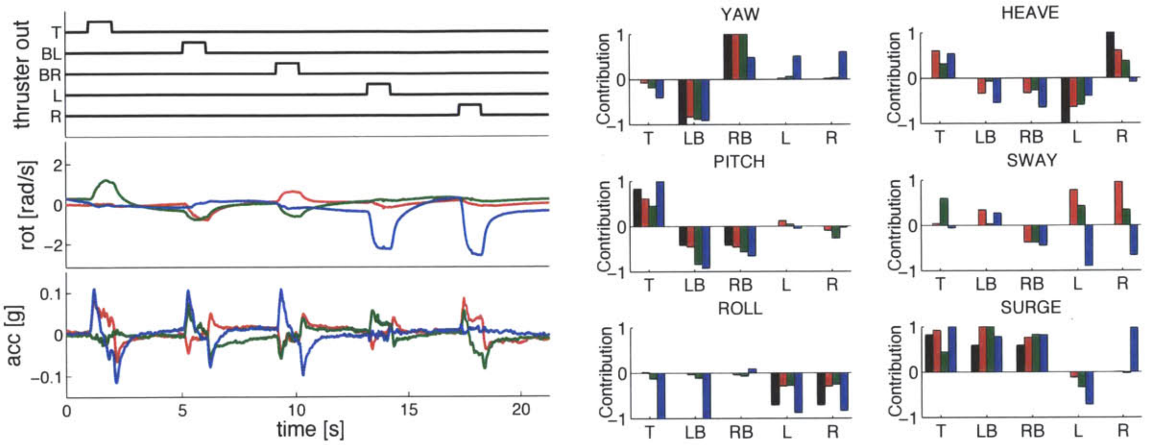
Figure 4-8: The thruster configurations used in the hardware experiments.

experiments are shown in Figure 4-8.

In the first set of experiments AMOUR is configured with the standard five thruster configuration visible in Fig. 4-8(a). Fig. 4-10(a) shows that when using the learned parameters the robot can successfully execute and maintain commanded changes in yaw, pitch, and roll. Fig. 4-10(a) and 4-10(c) show that the performance obtained by using the learned parameters is similar to that when using manually tuned parameters.

In the second set of experiments we rotate the top yellow thruster by 45 degrees off-axis to simulate a thruster misalignment as seen in Fig. 4-8(b). We again run the estimation algorithm. Fig. 4-10(b) shows the performance of the manual parameters for the default thruster configuration. The rotation of the top thruster means that it now has an effect on both pitch and yaw and this is reflected in the yaw error during pitching. Fig. 4-10(d) shows that the performance of the controller with newly learned parameters. With the learned parameters the robot successfully compensates for the thruster misalignment and the yaw errors during pitch maneuvers are reduced by more than a factor of 2.

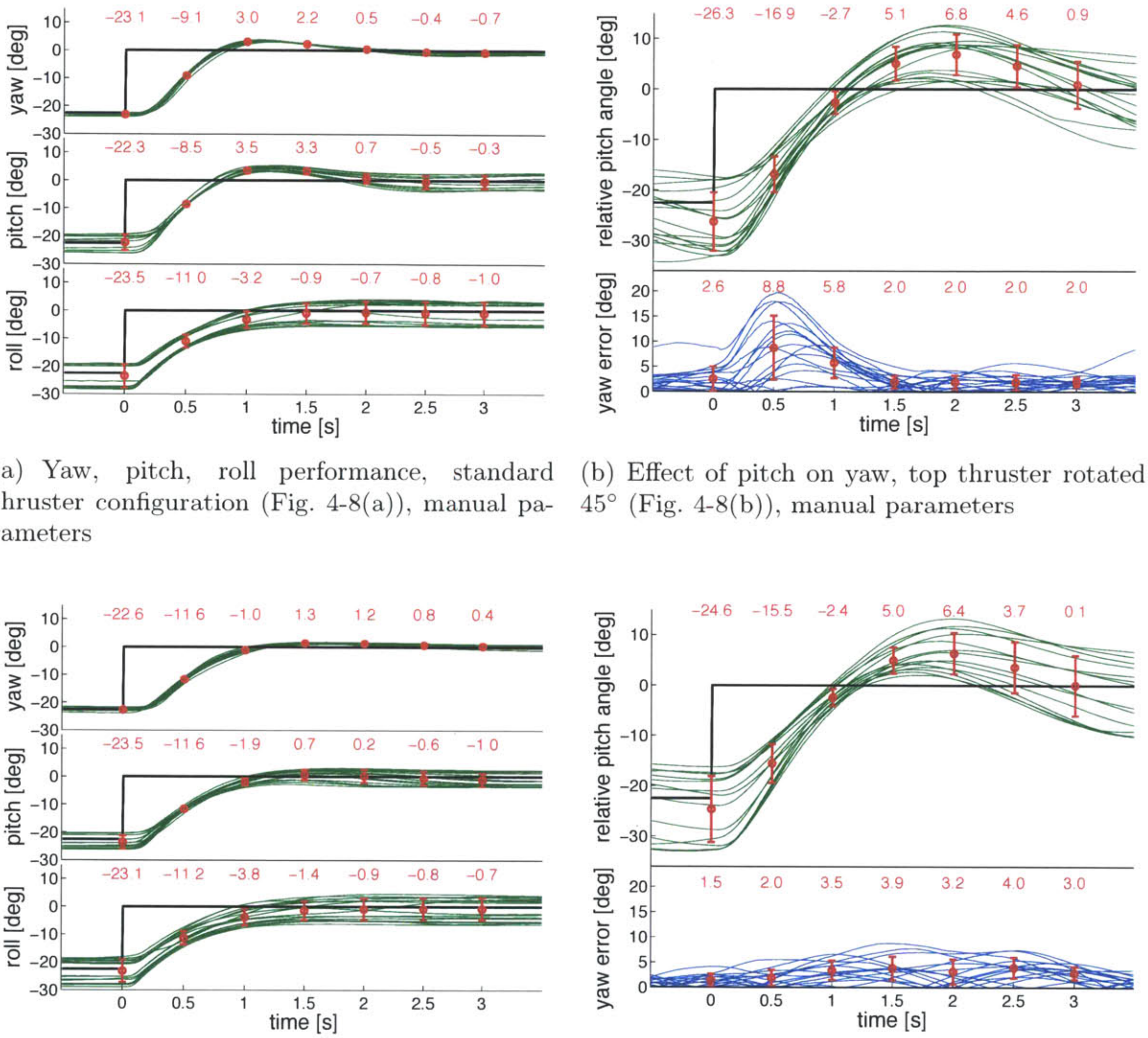
In the third set of experiments all thrusters are aligned with the robot's main body as shown in Fig. 4-8(c). The blue bars in Fig. 4-9(b) represent the learned parameters for this configuration. It can be seen that the algorithm correctly recognizes all thrusters contribution to yaw, pitch, and surge maneuvers (the left thrusters screw



(a) Accelerations and rotation changes in response to thruster pulsing.

(b) thruster configs: manual (black), Fig. 4-8(a) (red), Fig. 4-8(b) (green), Fig. 4-8(c) (blue)

Figure 4-9: Experimental data (a) and results of the thruster estimation algorithm (b). In plot (a) the x axis denote time in seconds and the y axis shows: (top) the 5 thruster commands with the thrusters being pulsed successively, (middle) the rotational speed of the robot in response to the thrusters, (bottom) the accelerations as measured on the robot. Red denotes a rotation around or an acceleration along the x-axis of the robots coordinate system. Green represents the robots y-axis and blue the z-axis. In plot (b) the learned rotation matrices and acceleration matrices are shown projected respectively onto the planes of yaw, pitch, roll as well as heave, sway, and surge. The y axis denote the magnitude of a thrusters contribution towards that particular motion. T, LB, RB, L, and R on the x axis denote the thruster as labeled in Fig 4-8. For each thruster 4 bar graphs are shown: (black) for the manual configuration, (red) for the default robot configuration shown in Fig. 4-8(a), (green) for the robot configuration shown in Fig. 4-8(b), and (blue) for the configuration shown in Fig. 4-8(c).



(a) Yaw, pitch, roll performance, standard thruster configuration (Fig. 4-8(a)), manual parameters

(b) Effect of pitch on yaw, top thruster rotated 45° (Fig. 4-8(b)), manual parameters

(c) Yaw, pitch, roll performance, standard thruster configuration (Fig. 4-8(a)), learned parameters

(d) Effect of pitch on yaw, top thruster rotated 45° (Fig. 4-8(b)), learned parameters

Figure 4-10: Experimental results showing performance of robot with thruster positions estimated by the algorithm. In plots (a) and (c) the y axes denote angular attitude of the vehicle projected onto the yaw, pitch, and roll planes. In plots (b) and (d) the y axes denote the pitch angle of the robot (upper part of each graph) and the angular yaw error (lower part of each graph). The black line in all plots represents the desired goal attitude. The green and blue lines represent the robots current attitude and attitude error respectively. The red numbers denote the mean angular error between the robots position and goal.

is left handed, so its thrust direction is reversed). However it is also visible that the algorithm does not detect missing degrees of freedom and 'learns' to heave, sway, and roll. This sometimes caused oscillations when controlling the robot.

4.4 Discussion

In this chapter we described the Modular Thruster Control Algorithm for autonomous underwater robots with modular thruster configuration. This algorithm can orient a robot in any configuration in SE^3. We describe how the control algorithm maps a desired change in robot state to individual thruster outputs. The algorithm accommodates the addition, removal, and arbitrary positioning of thrusters.

We show experimentally that our robot quickly achieves the target configurations with little or no oscillations. We are able to maintain accurate depth control even when performing barrel or head-over-heels rolls. Making use of this control algorithm on our robot gives us a robust and stable platform that can be reconfigured for a variety of tasks including deploying sensor nodes, taking pictures and videos, harbor and port security, and docking with other vehicles.

We also presented a thruster configuration estimation algorithm that eliminates the need for precise manual calibration of thruster positions. It allows quick in-situ estimation of the inverse thruster model, which enables direct control of the robot. In all experiments the algorithm estimates the final thruster positions within 40 seconds: only two thrust pulses per thruster (one in each direction) were necessary to learn an inverse model that allowed highly maneuverable control of the robot. The stability of the controller using the estimated parameters and the manual parameters for the default robot configuration was very good (Fig. 4-8(a)). In the case of a misaligned thruster (Fig. 4-8(b)) the estimation algorithm correctly identifies the misalignment and the controller outperforms the default manual parameters.

Chapter 5

Effective Communication over the Free-Space Optical Channel

In this chapter we focus on predicting underwater wireless optical communication performance. In particular, we present an end-to-end signal strength model used to predict signal strength and link quality and evaluate the performance of AquaOptical II.

Wireless optical communication links are crucial for underwater observatories as they allow the creation of high data rate connections between underwater robots and sensor nodes or data hubs without the need for docking and while maintaining full maneuverable of the robot. Being able to predict the communications signal strength allows us to forecast link quality and also to use real signal strength measurement in position estimation applications. These are useful tools that will allow to better utilize the optical communications channel underwater.

5.1 An End-to-End Underwater Optical Signal Strength Model

In this chapter we develop an end-to-end model that helps to estimate the signal strength and resulting communication range in all directions. The model incorporates

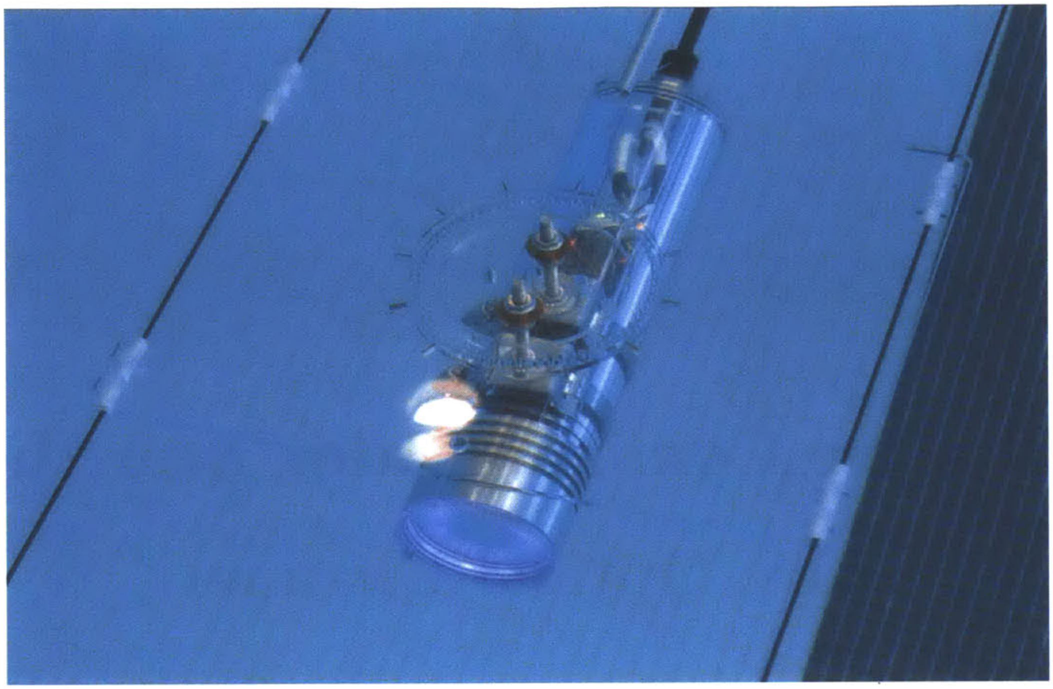


Figure 5-1: The transmitter is mounted to one end of the test setup. The receiver can be transmitter in steps of 1 degree relative to the direct line of sight.

all components of the modem, including light sources, optics, detectors, amplifiers, and finally analog to digital converters. Our focus is to model the signal strength at any relative pose between transmitter and receiver. This includes the distances between the two and their individual poses.

The model supports underwater robotics in two ways. First, it enables estimation of the effects of the optical modem's components, providing a tool for examining how changes to individual components will affect range and coverage across larger angles with the goal of increasing robustness to pointing. Second, the information provided by the model can be used for location estimation. We showed previously that an AUV can use the optical signal strength to maintain position within communication range without external localization [46]. We believe that an optical signal strength model together with a Bayesian filter will enable enhanced localization.

This chapter contributes the following:

- An end-to-end model of signal strength in underwater optical communication.

- An experimental evaluation of the model using the AquaOptical II modems.
- A discussion of the model in the context of the current state of the art in underwater optical communication.

The remainder of this chapter is organized as follows. Section 5.3 presents the signal strength model. Section 3.2.3 gives an overview of AquaOptical II, the hardware used for experimental validation. We instantiate the model for AquaOptical II in Section 5.4 by learning the relevant parameters. In Section 5.5 we evaluate the predictions made by the model using AquaOptical II in the pool. We provide further discussions in Section 5.6.

5.2 High-Level Overview

The model proposed allows to compute the power arriving at the detector and from this the resulting signal strength available for decoding. For a chosen hardware configuration the model computes the received power as a function of transmission angle θ, transmission distance r, and reception angle ϕ.

5.3 Signal Strength Model

Figure 5-2 shows the general layout of our model from the digital signal to be sent to the digital signal to be received. We will focus on the blocks starting from the high power amplifier to the low noise amplifier because they have the most significant effect on optical signal transmission through water. The following model blocks will be treated in individual subsections:

- The high power amplifier drives the optical source from logic level signals. We model it in Section 5.3.1.
- We model how the signal is radiated in different directions, the source angular power density, in Section 5.3.2.

- In Section 5.3.3 we model the effect of source optics on the signal direction and intensity.
- The optical channel is modeled in Section 5.3.4.
- Detector optics are modeled in Section 5.3.5.
- Detector angular sensitivity is modeled in Secion 5.3.6.
- The low noise amplifier converts the sensed signal to detectable levels before it is digitized by the analog to digital converter. We model them in Section 5.3.7.

5.3.1 High Power Amplifier

The high power amplifier (HPA) converts the modulated signal into a drive current for the transmit element, which is a light source. Currently all optical modems deployed in water operate at bandwidths below 100 MHz and use amplitude modulation. The amplifiers used for this purpose are simple, and we approximate them as an identity function.

5.3.2 Source Angular Power Density

First we look at the energy radiated from the light source and at how this energy is spread spatially. We consider point light sources. Let P_{total} be the total radiant

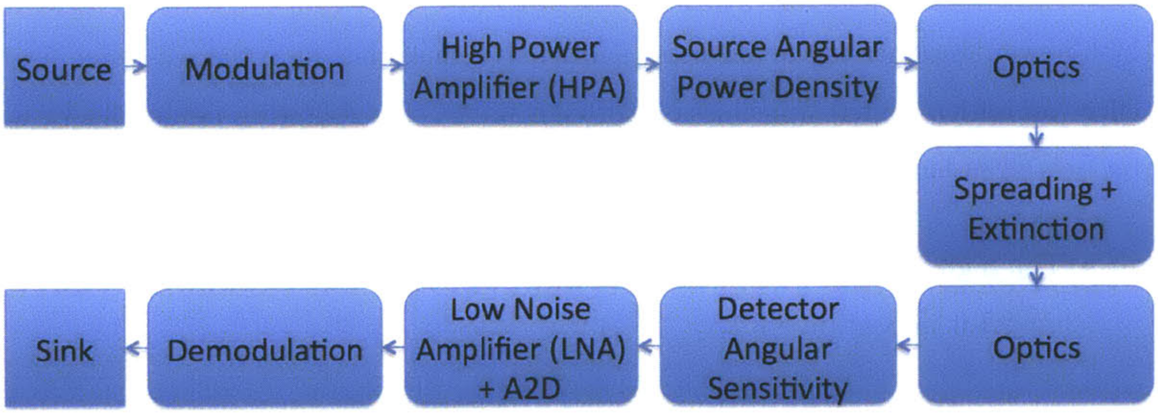

Source
Modulation
High Power Amplifier (HPA)
Source Angular Power Density
Optics
Spreading + Extinction
Optics
Detector Angular Sensitivity
Low Noise Amplifier (LNA) + A2D
Demodulation
Sink

Figure 5-2: Block diagram of end-to-end signal strength model.

power of the light source, $\theta \in [0, \pi/2]$ be the angular displacement relative to the optical axis of the light source, and I_{peak} be the peak radiant intensity of the light source (usually at $\theta = 0$). Let $f(\theta) \in [0, 1]$ be the relative intensity function. We make the assumption that $f(\theta)$ is rotationally symmetrical around the optical axis, but if necessary this constraint can be relaxed. The units are [W] for P_{total}, [W/sr] for I_{peak}, [rad] for θ, and [1] for $f(\theta)$. The relationship between these values is defined as

$$P_{total} = \int I_{peak} f(\theta_\Omega) \mathrm{d}\Omega \tag{5.1}$$

and by substituting $\mathrm{d}\Omega = \sin(\theta)\mathrm{d}\theta\mathrm{d}\varphi$ we get

$$\begin{aligned} P_{total} &= \int_{-\pi}^{\pi} \int_{0}^{\pi/2} I_{peak} f(\theta) \sin(\theta) \mathrm{d}\theta \mathrm{d}\varphi \\ &= 2 \cdot \pi \cdot \int_{0}^{\pi/2} I_{peak} f(\theta) \sin(\theta) \mathrm{d}\theta. \end{aligned} \tag{5.2}$$

Solving for I_{peak} allows us to express the optical signal radiant intensity I_θ for a given angular displacement θ as

$$I_\theta = I_{peak} f(\theta). \tag{5.3}$$

5.3.3 Source Optics

Refraction

Before the optical signal reaches the water, it travels through multiple materials with different refractive indices. For example, in AquaOptical II the optical signal first travels through air inside the modem. The signal then transitions from air to acrylic, the material from which the viewport is made. Finally the signal transitions from the acrylic viewport into the water. Each transition can result in refraction, reflection, and transmittance. Refraction can have a focusing effect on the beam and may be desirable. Reflection is usually undesirable at these interfaces, as it prevents the signal from exiting the modem. Transmittance defines the amount of power that will pass through the material; the remainder is absorbed. All of these properties depend on the incident angle of the optical signal and on the refractive indices of the materials. Table 5.3.3 shows the refractive indices for the materials used in AquaOptical.

Snell's law defines the relationship between the incident angle θ_i, the transmitted angle θ_t, the refractive index of the medium that the signal is leaving n_1, and the refractive index of the medium that the signal is entering n_2, as

$$\frac{\sin\theta_i}{\sin\theta_t} = \frac{n_2}{n_1}. \tag{5.4}$$

Solving for the transmitted angle θ_t we get

$$\theta_t = \sin^{-1}\left(\frac{n_1}{n_2}\sin\theta_i\right). \tag{5.5}$$

Reflection

The reflected angle θ_r is equal to the incident angle θ_i. Using Fresnel's equations, we get the reflection coefficients (fraction of power reflected) for in-plane (R_s) and out-of-plane (R_p) polarizations:

$$R_s = \left|\frac{n_1\cos\theta_i - n_2\cos\theta_t}{n_1\cos\theta_i + n_2\cos\theta_t}\right|^2 \tag{5.6a}$$

$$R_p = \left|\frac{n_1\cos\theta_t - n_2\cos\theta_i}{n_1\cos\theta_t + n_2\cos\theta_i}\right|^2 \tag{5.6b}$$

respectively. For unpolarized light the reflection coefficient R is the average of these values, and, because of conservation of energy, the transmission coefficient T is one minus R;

Material	n at 470 nm
Air	1.00
Water	1.33
Acrylic	1.50
Quartz glass	1.46

Table 5.1: Refractive Indices for selected materials.

$$R = \frac{R_s + R_p}{2} \qquad\qquad T = 1 - R. \tag{5.7}$$

Intensity Change

Refraction at a surface causes a beam to collimate when $n_1 < n_2$) or de-collimate when $n_1 > n_2$. This will increase or decrease the radiant intensity of the signal. This change in intensity can be computed by looking at the solid angle change at the boundary. In the case that the direction of light is oblique to the boundary between n_1 and n_2, the change of the solid angle Ω at the boundary is given in [18] as

$$d\Omega_1 = \frac{n_2^2 \cos\theta_t}{n_1^2 \cos\theta_i} d\Omega_2 \tag{5.8}$$

where Ω_1 is the incident solid angle and Ω_2 is the departing solid angle. This implies that the radiant intensity change across the boundary is

$$\frac{I_i}{n_1^2} = \frac{I_t}{n_2^2}. \tag{5.9}$$

5.3.4 Optical Channel

Spreading

In this chapter we consider a spherical spreading model with exponential decay. This model works well in clear waters, such as pool water, where scattering does not play a crucial role.

First we consider power loss due to spherical spreading. Let r be the distance in [m] at which we measure the signal intensity. Using the radiant intensity I_θ from Eq. 5.3, we get irradiance $E_{\theta,spherical}(r)$ at distance r, measured in [W/m^2]:

$$E_{\theta,spherical}(r) = I_\theta/r^2. \tag{5.10}$$

Extinction

Let c be the absorption coefficient for the water in which we are transmitting. We assume that the coefficient is uniform across the entire length of transmission. This is a valid approximation for pool water but it can be imprecise for shallow ocean environments where currents cause inhomogeneity of the water quality. Beer's law gives the signal degradation at distance r caused by absorption as

$$S_r(r) = \exp(c \cdot r) \tag{5.11}$$

Combing spherical spreading with exponential decay we get a resulting irradiance of

$$E_\theta(r) = I_\theta \cdot \exp(c \cdot r)/r^2. \tag{5.12}$$

Temporal Signal Spreading and Multi-path

If c_{vacuum} is the speed of light in vacuum, then the speed of light in a solid with refractive index $n_{material}$ is:

$$c_{material} = \frac{c_{vacuum}}{n_{material}} \tag{5.13}$$

In the simplest case, this means that, if the signal has to travel an extra distance Δd, the exhibited temporal spreading is

$$\Delta t = \frac{\Delta d}{c_{material}} = \frac{\Delta d \cdot n_{material}}{c_{vacuum}} \tag{5.14}$$

which in the case of water will amount to 100 ns for every 22.49 m. This means that even traveling a distance of 22.49 m only results in 100 ns temporal spreading. Because the signal will have been heavily attenuated by then we can discard the effect of multi-path for communication speeds below 10 Mbps.

5.3.5 Detector Optics

The optics at the detector follow the same laws as the optics at the source. We need to again consider Snell's law (Eq. 5.5), Fresnel's equations (Eq. 5.7), and intensity change cause by refraction (Eq. 5.9).

5.3.6 Detector Angular Sensitivity

There are three commonly used optical detectors for sensing light intensity at low levels: (1) a conventional photodiode, (2) an avalanche photodiode, and (3) a photomultiplier tube. All three convert incident radiant power into current[1]. The output current I_{out} is linearly related to the incident power P_{in}. For a given sensor this relationship, called responsivity, can be expressed as a constant

$$R_d = I_{out}/P_{in}. \tag{5.15}$$

The incident power can be computed from the signal irradiance arriving at the detector (Equation 5.10), the incident angle ϕ, and the area A_d (assumed flat) of the

[1] This is true for all photodiodes, even when used in photovoltaic mode, where an additional resistor is used to generate and output voltage as opposed to output current.

detector as

$$P_{in} = E_\theta(r) \cdot A_d \cdot \cos(\phi). \tag{5.16}$$

This yields a photodiode output current of

$$\begin{aligned} I_{PD} &= R_d \cdot P_{in} \\ &= R_d \cdot E_\theta(r) \cdot A_d \cdot \cos(\phi). \end{aligned} \tag{5.17}$$

5.3.7 Low Noise Amplifier

Amplification

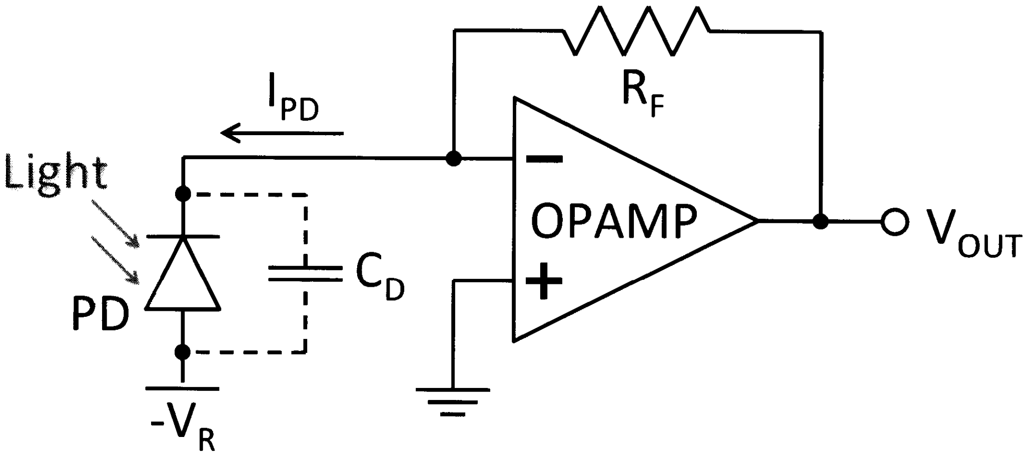


Figure 5-3: Transimpedance amplifier (TIA) with photodiode (PD). C_D is the parasitic photodiode capacitance, R_F the feedback resistor to the operational amplifier (OPAMP), I_{PD} the current through the photodiode (dark current and photon induce current), and V_{OUT} the output voltage of the amplifier.

The current produced by the photodiode generally needs to be amplified and converted into a voltage before it can be digitized by means of an analog to digital converter. This is typically done using a transimpedance amplifier (TIA). The schematic of a TIA is shown in Figure 5-3. The TIA will convert the current from

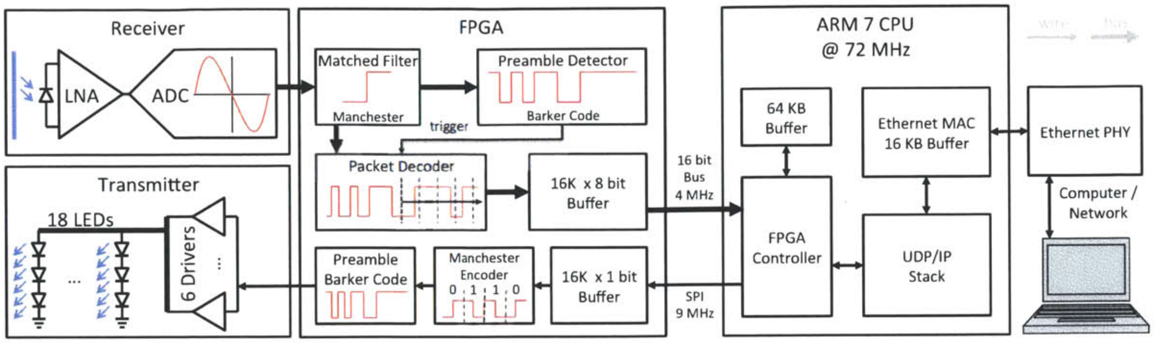


Figure 5-4: System overview of AquaOptical II. The system consists of a transducer comprising the Receiver and Transmitter units, an FPGA that implements the physical and link layer, and an ARM 7 CPU that implements the MAC layer and Ethernet communication.

Equation 5.17 into a voltage V_{out}, and this relation can be simplified as being proportional for frequencies below the bandwidth:

$$V_{out} = R_{TIA} \cdot I_{PD}. \tag{5.18}$$

Bandwidth

Section 5.3.6 mentioned 3 different detectors (photodiodes, avalanche photodiodes, and photomultiplier tubes). Two of the tradeoffs between these detectors are sensitivity and bandwidth. Detectors should be chosen such that they provide the bandwidth necessary and have the highest possible sensitivity, assuming constant noise.

Usually the more sensitive a device, the higher its intrinsic capacitance C_D, which results in higher bandwidth. However, lower intrinsic capacitance C_D results in higher diode impedance, which is computed as

$$X_c = \frac{1}{2\pi \cdot f \cdot C_D}. \tag{5.19}$$

The TIA uses an operational amplifier to amplify the incoming signal. The bandwidth f_{BW} of the TIA amplifier can be computed from the operational amplifier's gain-bandwidth-product GBW and set gain $G = R_F/X_c$ as

$$f_{BW} = \frac{GBW}{G} = \frac{GBW \cdot X_c}{R_F} = \frac{GBW}{2\pi \cdot f \cdot C_D \cdot R_F}. \tag{5.20}$$

To get the bandwidth of the TIA for a photodiode with a given capacitance C_D and impedance X_c (Equation 5.19), we set $f_{BW} = f$ which yields

$$f = \left(\frac{GBW}{2\pi \cdot C_D \cdot R_F}\right)^{1/2}. \tag{5.21}$$

Noise

The photodiode current noise is expressed as noise current spectral density $I_{N,PD}$ in $[\mathrm{A}/\sqrt{\mathrm{HZ}}]$. The current noise at a given frequency f is computed as

$$I_{N,PD}(f) = I_{N,PD} \cdot f^{1/2}. \tag{5.22}$$

An additional source of noise is Johnson-Nyquist noise originating from the feedback resistor R_F. At a frequency f and temperature T in Kelvin, this noise is equal to

$$I_{N,RF}(f) = (4 \cdot k_B \cdot T \cdot R_F \cdot f)^{1/2} \tag{5.23}$$

where $k_B = 1.38 \cdot 10^{-23}$ J/K is the Boltzmann constant. Both noise figures will be amplified by the TIA by the same factor as the signal, so they should be compared to the current signal exiting the photodiode.

5.4 Estimation of AquaOptical's Signal Strength

In this section we instantiate the model described in Section 5.3 to the hardware described in Section III.

5.4.1 Source Angular Power Density and Source Optics

AquaOptical II uses Luxeon Rebel Blue LEDs (LXMLPB01-0023). The relative intensity graph of this LED is shown in Figure 5-5. We used a polynomial fit of degree 6 (with R-squared = 0.9999) in order to facilitate further computation. The LED's typical luminous flux is reported as 70 lm at a drive current of 1 A. The relationship

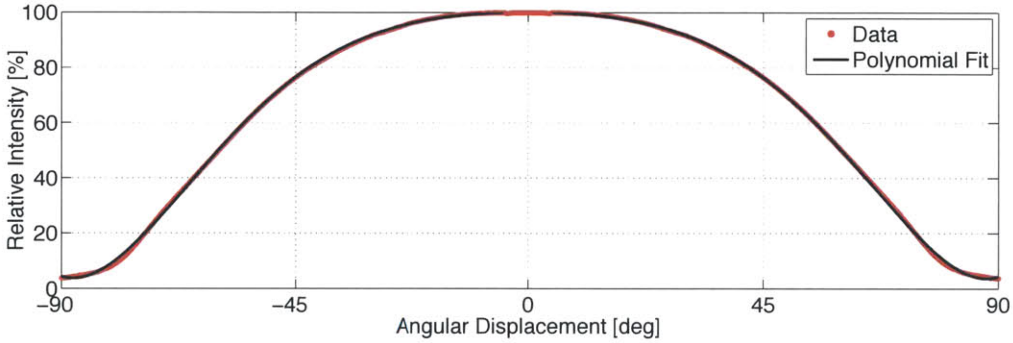


Figure 5-5: Relative intensity of LED taken from data sheet (red) and polynomial fit (black). The x-axis shows the angular displacement in degrees, the y-axis shows the relative intensity in percent.

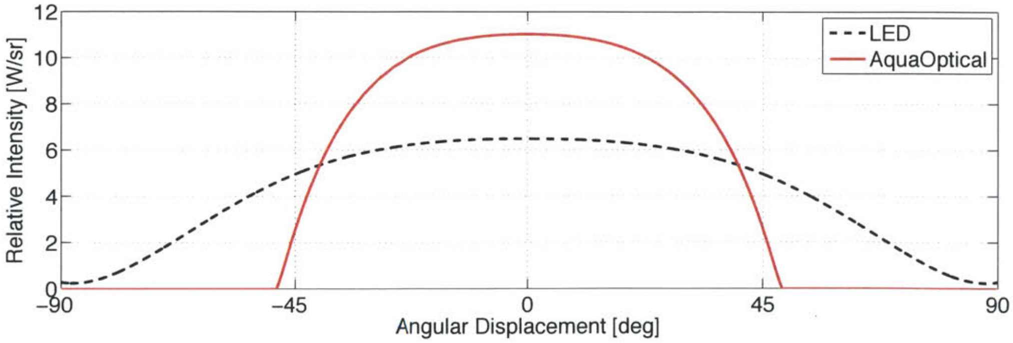


Figure 5-6: Accumulated radiant intensity of the 18 LEDs inside the AquaOptical II modem (black dashed line) and the AquaOptical II modem itself (red line). The x-axis shows the angular displacement in degrees, the y-axis shows the radiant intensity in W/sr.

between luminous flux F and radiant power P for a fixed wavelength λ is defined as $F = 683\ \text{lm/W} \cdot \bar{y}(\lambda) \cdot P$. The luminous efficiency $\bar{y}(\lambda)$ at 470 nm is specified as 9.1 % [98]. This means that the LED's total radiant power is $P_{total} = 1.126$ W. Solving Equation 5.2 yields $I_{peak,LXML-PB01} = 0.36$ W/sr in peak radiant intensity. AquaOptical II uses 18 Luxeon Rebel Blue LEDs to transmit. That means that AquaOptical's peak radiant power will be 18 times 0.36 W/sr. Further, our signal first transitions from air into acrylic, and then from acrylic into water, so we need to apply Equations 5.5, 5.7, and 5.9. This yields the radiant intensity as shown in Figure 5-6.

5.4.2 Receiver Sensitivity and Receiver Optics

AquaOptical II uses an avalanche photodiode (APD) that has a circular active area with a diameter of 5 mm. We are interested in the optical energy that arrives and is absorbed by the flat surface of the LED. The radiance arriving at the receiver is given in Equation 5.12. The incoming signal needs to transition from water, through acrylic, air, an optical filter, and air again before it finally reaches the detector. It should be noted, that we approximated the filter's refractive properties with those of glass, as no data for off-optical-axis performance is available. The computed relative angular sensitivity of the receiving modem is shown in Figure 5-7.

The transimpedance amplifier (TIA) used by AquaOptical is part of the APD module sold by Advanced Photonix as part number SD197-70-74-661. The specifications

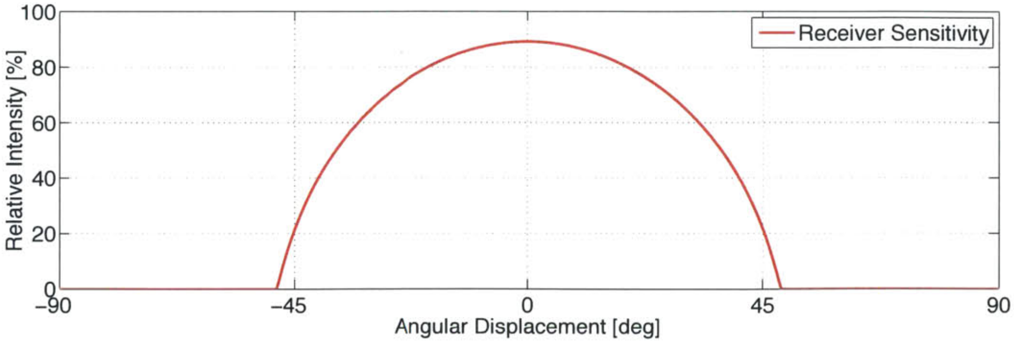


Figure 5-7: Computed relative receiver sensitivity. The x-axis shows the angular displacement in degrees, the y-axis shows the relative intensity in percent.

of the module give a sensitivity $S = 9.5 \cdot 10^5$ V/W and a bandwidth of no less than 12 MHz. The noise current spectral density of the APD is $I_{N,APD} = 0.8 \cdot 10^{-12}$ A/$\sqrt{\text{Hz}}$, and the internal feedback resistor of the TIA is $R_F = 10$ KΩ. Our signal is sampled at $f = 32$ MHz. Using Equation 5.22, we get the photodiode induced voltage output noise $U_{N,PD} = 45$ μV. Using Equation 5.23, we get the noise originating from the feedback resistor $U_{N,RF} = 75$ μV (at 25 °C). Both of these values are well below the 1 mV quantization induced by the digital to analog converter.

5.4.3 Full AquaOptical II Signal Strength Model

Let d be the distance in meters between the two AquaOptical II modems, θ the transmission angle relative to the source's optical axis, and ϕ the incident signal angle relative to the detector's optical axis. We previously estimated the water in our pool to have an e-folding length of 36 m [42]. Let $c = 1/36$ m^{-1} be the absorption coefficient of pool water. Let $A_d = 19.64$ mm^2 be the detector area. We call $I(\theta)$ the source radiant intensity function of AquaOptical II as depicted in Figure 5-6 and $S(\phi)$ the detector relative angular sensitivity function as depicted in Figure 5-7. The power in [W] arriving at the detector is expressed as

$$P(\theta, \phi, d) = I(\theta) \cdot \frac{\exp(c \cdot r)}{r^2} \cdot S(\phi) \cdot A_d \cdot \cos(\phi) \tag{5.24}$$

Figures 5-13(a) and 5-13(b) show the received power for various spatial configurations of the transmitter and the receiver. In Figure 5-13(a) the transmitter is fixed at the origin point right along the x-axis. The x- and y-axes denote the relative location of the receiver, which is always pointing straight at the transmitter ($\phi = 0$). Since the transmitter orientation is fixed, θ depends on the location of the receiver.

The opposite is true for Figure 5-13(b). Here, the receiver is fixed at the origin point right along the x-axis. The x- and y-axes denote the relative location of the transmitter, which is always pointing straight at the receiver ($\theta = 0$). In this case, ϕ depends on the location of the transmitter.

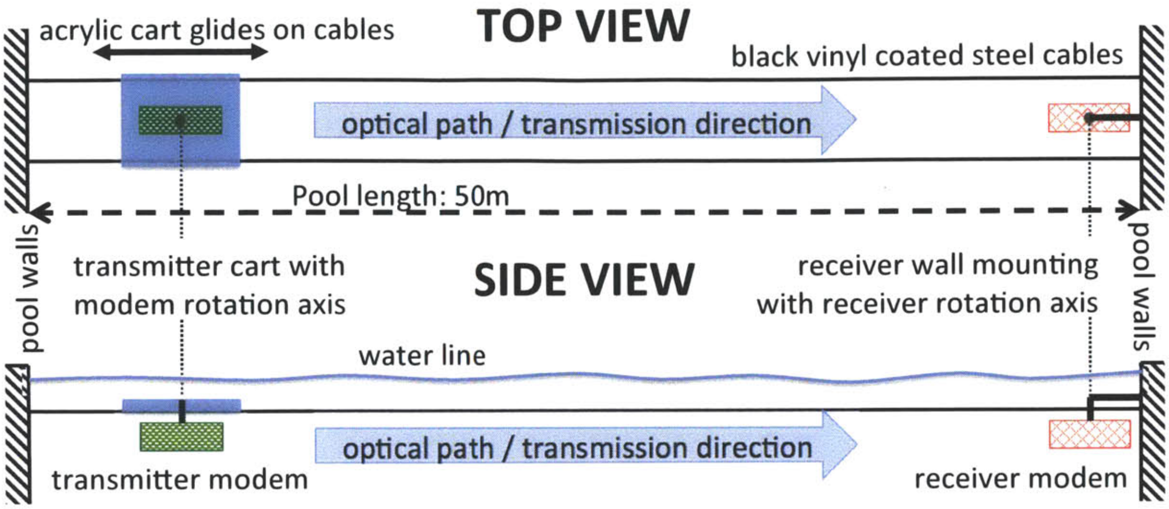

acrylic cart glides on cables
TOP VIEW
black vinyl coated steel cables
optical path / transmission direction
Pool length: 50m
pool walls
transmitter cart with
modem rotation axis
SIDE VIEW
receiver wall mounting
with receiver rotation axis
pool walls
water line
optical path / transmission direction
transmitter modem
receiver modem

Figure 5-8: Experimental setup. Two steel cables are tensioned between two opposing pool walls over a distance of 50 m. The cables are coated in black vinyl to reduce reflections and friction. The receiving modem is mounted to one of the pool walls. The mounting angle of the receiving modem is variable in the horizontal plane to allow arbitrary choice of receiver angle. An acrylic plate is attached to the cables using two PTFE bushings per cable to ensure stability. This plate can be repositioned to any distance from the receiver. The transmitting optical modem is mounted to the bottom of the acrylic plate. The angle of mounting of the transmitting modem is variable as well to allow arbitrary choice of transmitter angle.

5.5 Experimental Validation

We experimentally validate our model using the AquaOptical II modem [43]. We have designed a test fixture that allows us to arbitrarily position two modems relative to each other in the pool. The fixture is shown in Figure 5-9. It allows us to translate the transmitter relative to the receiver to arbitrary distances (limited by the pool size, which is up to 50 m long) and to change both the receiver angle and the transmitter angle relative to the line of sight.

Our model is evaluated in clear pool water but can be extended to turbid waters such as the ocean environment by changing the water channel model. In the easiest case this can be achieved by simply adjusting the extinction coefficient of the water.

5.5.1 Experimental Setup

Figure 5-8 shows a schematic of the experimental setup. Two steel cables are tensioned between two opposing pool walls over a distance of 50 m. The cables are coated in black vinyl to reduce optical reflections and friction for the moving parts. The receiving modem is mounted to one of the pool walls using a specially designed rig. The mounting angle of the receiving modem is variable in the horizontal plane. This allows arbitrary choice of the receiving angle. An acrylic plate is attached to the cables using two PTFE bushings. Two bushings are used per cable to ensure stability. The transmitting optical modem is mounted to the bottom of the acrylic plate. The

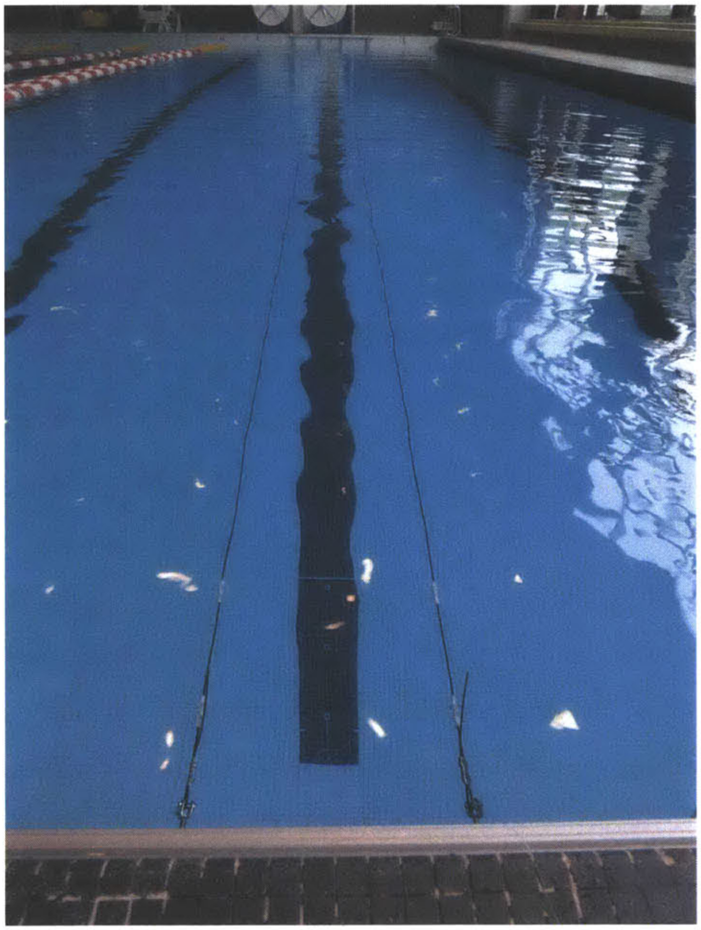


Figure 5-9: Picture of the pool with the test setup. Two steel wires coated in black vinyl to minimize reflections are strained across the pool. A receiver is attached at one end, and a transmitter (Figure 5-1) can travel on a cart attached to the wires. The cart is moved using a pulley system driven by transparent fishing line. In the final setup four additional wires form a 0.6 m by 0.6 m tunnel around the setup that is draped in black cloth to minimize reflections from pool walls and the water surface.

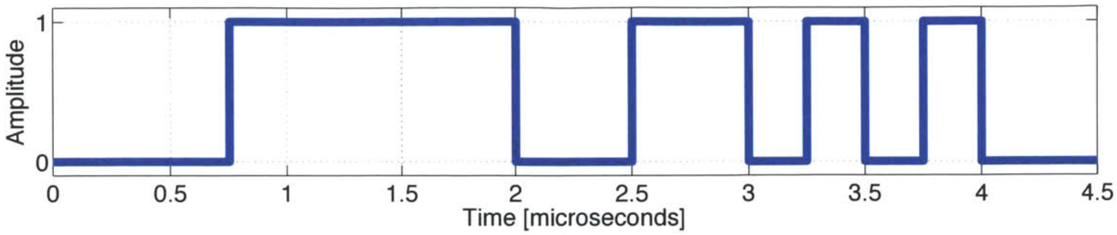


Figure 5-10: Header code sent by transmitter and used in the matched filter by the receiver.

mounting angle of the transmitting modem is variable in the horizontal plane. This allows arbitrary choice of the transmitting angle. The acrylic plate together with the attached transmitting optical modem can glide along the cables allowing the transmitter to be repositioned anywhere along the cables. This allows arbitrary choice of transmission distance between the receiver and the transmitter limited only by the pool size of 50 m.

5.5.2 Experimental Procedure

The transmitting modem is programmed to transmit symbols at a rate of 250 Hz. Each symbol consists of a 18-bit code transmitted at a bit rate of 8 Mbps. Thus, a single symbol is 4.5 μs long. The code is based on a 13-bit Barker code and is displayed in Figure 5-10. The receiver samples the incoming signal at 32 MSPS (4 times oversampling) and applies a 72 sample wide matched filter to the incoming signal. The output of the matched filter is divided by the current signal power. When the resulting quotient exceeds a threshold of 0.8, a received symbol is registered and the peak power is recorded as a single signal strength measurement.

We call a specific set (α, β, d) of transmitter angle α, receiver angle β, and transmission distance d a data point. For each data point, the modems are moved to the correct configuration and then all registered signal strength measurements are recorded for 30 s. Because of false negatives when a sent symbol is not detected, it is possible that we record fewer than 30 s $\cdot$ 250 Hz $=$ 7500 measurements for a specific data point. This is especially true for data points where the signal strength is weak, and, as a result, the symbols are harder to detect.

We collected two sets of data points. For the first set we collected data points with the transmitting modem always aimed straight at the receiving modem ($\alpha = 0$). For this set we first move the transmitter to a fixed distance. The transmitter remains at this distance while we alter the receiver angle (β) and collect a 30 s long data point each time. Then we move the transmitter to the next distance. For the second set, we collected data points with the receiving modem always aimed straight at the transmitting modem ($\beta = 0$). For this set we first set the transmitting angle (α) to a fixed value. We then move the transmitter to different distances and collect a 30 s long data point each time. We repeat this for different values of α. The distances used for both sets range from 5 m to 50 m in multiples of 5 m.

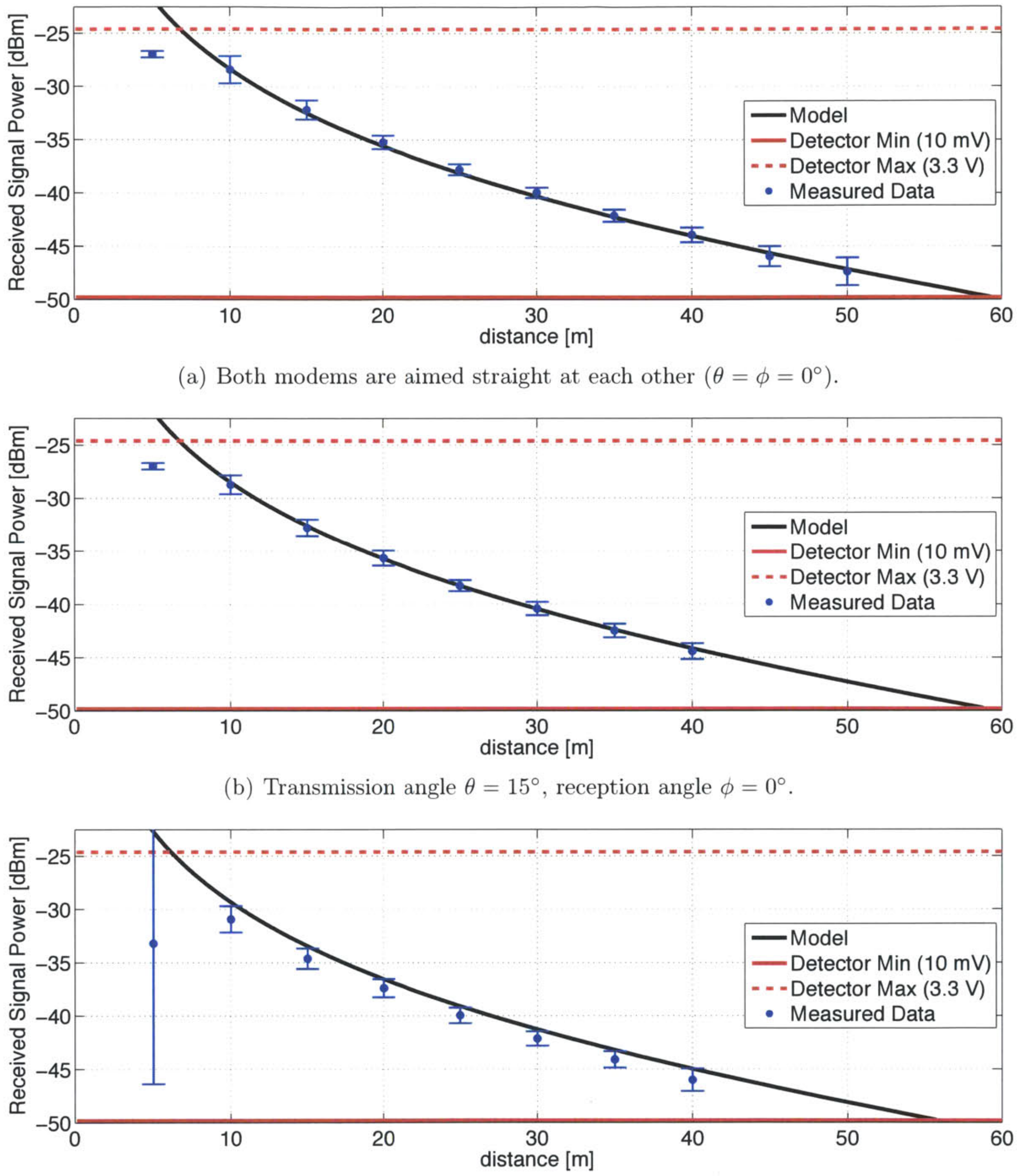
5.5.3 Experimental Results

First, let us consider the case in which both modems are aimed at each other ($\theta = \phi = 0$). Figure 5-11(a) plots both the modeled signal strength and the experimental results for this case over transmission distances of 5 to 50 m. All measurements are reported in dBm[2]. Plotted in black is the model prediction. The blue dots correspond to the collected data. For distances between 10 m and 50 m the measurements agree with the model with a maximum error[3] of 0.35 dB. The only outlier is the measurement at 5 m, which is off from the modeled value by -5.24 dB. The reason for this is sensor saturation. Plotted as a horizontal dashed red line is the detector saturation, which is the equivalent of 3.3 V at the TIA output. Near this range the amplifiers do not behave linearly and the signal becomes heavily distorted. The solid red horizontal line marks the detector threshold, which is set by the modem firmware to be 10 mV at the TIA output. The model predicts that this threshold is reached at 59.36 m.

Figures 5-11(b), 5-11(c), 5-12(a) and 5-12(b) plot both the modeled signal strength

[2] Absolute power referenced to 1 mW: 1 mW = 0 dBm, 1 μW = -30 dBm.

[3] The relative difference between two absolute power values is expressed in dB.

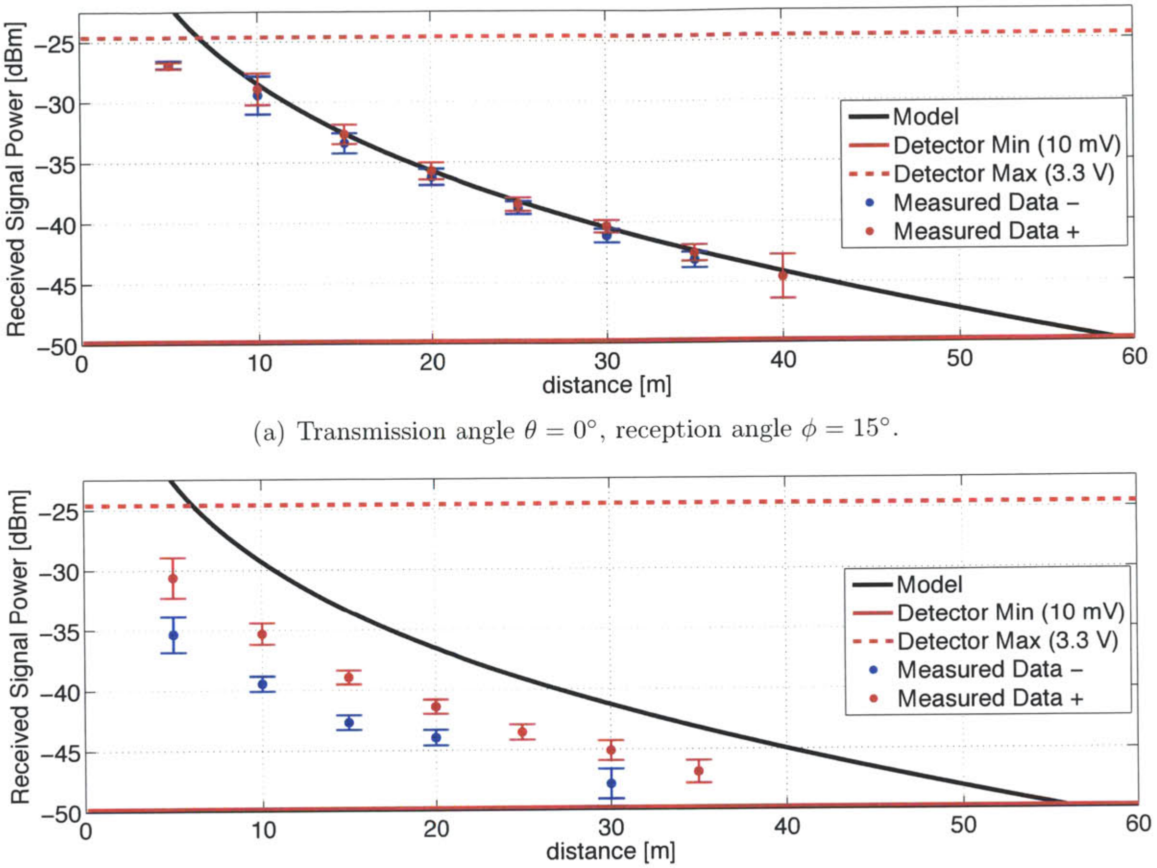


(a) Both modems are aimed straight at each other ($\theta = \phi = 0°$).

(b) Transmission angle $\theta = 15°$, reception angle $\phi = 0°$.

(c) Transmission angle $\theta = 30°$, reception angle $\phi = 0°$.

Figure 5-11: Modeled signal strength and experimental data. The x-axis shows distance in [m]. The y-axis shows received signal strength in [dBm]. The black line shows the output of the model from Section 5.4.3. The red lines mark the detection threshold (solid) and saturation (dashed). Experimental data is plotted blue with 3σ error bars.

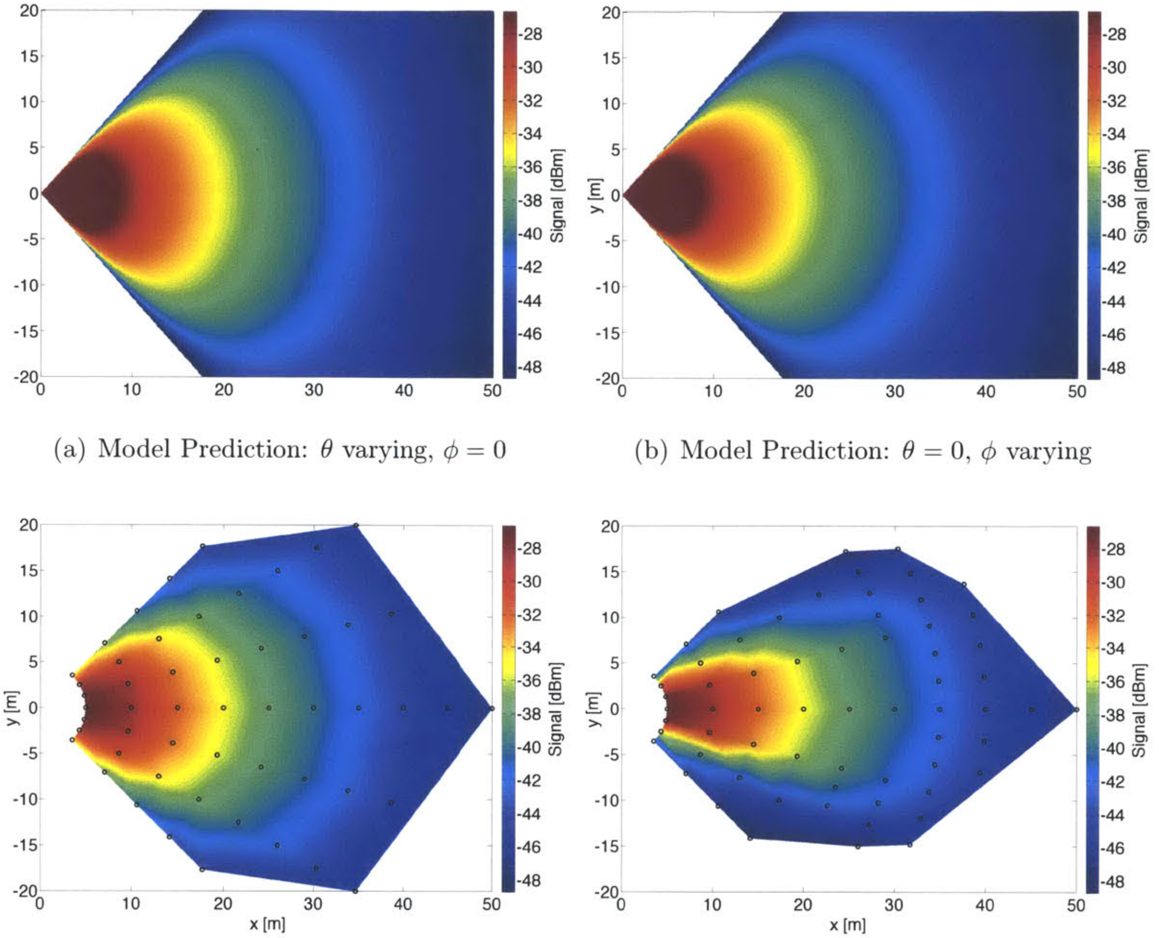


(a) Transmission angle $\theta = 0°$, reception angle $\phi = 15°$.

(b) Transmission angle $\theta = 0°$, reception angle $\phi = 30°$.

Figure 5-12: Modeled signal strength and experimental data. The x-axis shows distance in [m]. The y-axis shows received signal strength in [dBm]. The black line shows the output of the model from Section 5.4.3. The red lines mark the detection threshold (solid) and saturation (dashed). Experimental data is plotted blue (negative reception angle) and red (positive reception angle), each with 3σ error bars.

and the experimental results for the cases with transmit and receive angles of $(\theta, \phi) \in \{(15, 0), (30, 0), (0, 15), (0, 30)\}$. The entire experimental data set is plotted as heat maps in Figures 5-13(c) and 5-13(d). At a distance of 5 m all data points are corrupted by detector saturation, which is why we do not include them in the following error analysis. Figure 5-14 plots the maximum modeling errors for different values of θ and ϕ. For the cases where $\phi = 0$, the maximum errors over the distances of 10 to 50 m are 0.30 dB for $\theta = 15$ degrees, 1.60 dB for $\theta = 30$ degrees, and 2.92 dB for $\theta = 45$ degrees. For the cases where $\theta = 0$, the maximum errors over the distances



(a) Model Prediction: θ varying, $\phi = 0$

(b) Model Prediction: $\theta = 0$, ϕ varying

(c) Experimental Data: θ varying, $\phi = 0$

(d) Experimental Data: $\theta = 0$, ϕ varying

Figure 5-13: Signal strength as measured at the detector: Simulation results (top: a,b) and experimental data (bottom: c,d). The signal strength is color coded from blue (-48.7 dBm, weak signal) to red (-26.7 dBm, strong signal). For each plot the x- and y-axes show displacement of one of the modems relative to the other in [m]. In the left plots (a,c) the transmitter is located at the origin and always points right (positive x-axis). The x and y-axes show receiver displacement and the receiver is always pointing straight at the transmitter (θ varying, $\phi = 0$). In the right plots (b,d) the receiver is located at the origin and always points right (positive x-axis). The x- and y-axes show transmitter displacement, and the transmitter is always pointing straight at the receiver ($\theta = 0$, ϕ varying). The black circles show actual data points taken during the experiments; the data in-between is interpolated. In (d) all measurement points are independent. In (c) data points were collected only for positive θ and the data was mirrored.

of 10 to 50 m are 0.89 dB for $\phi = -15$ degrees, 0.35 dB for $\phi = 0$ degrees, and 0.39 dB for $\theta = 15$ degrees. For angles ϕ great than 30 degrees the maximum errors

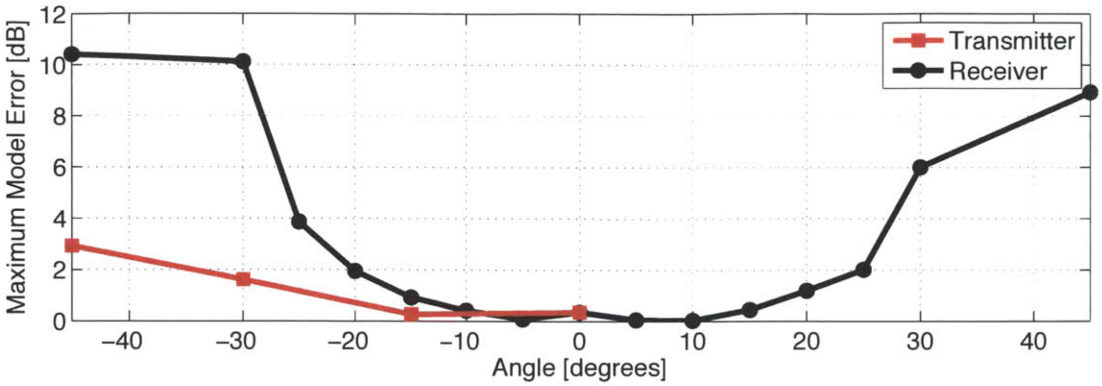


Figure 5-14: Maximum model errors when compared to experimental data for distances between 10 and 50 m. The x-axis shows the angle (θ for the red and ϕ for the black line). The y-axis shows the error in [dB]. The red line plots the errors when transmitter angle was changing. The black line plots the errors when the receiver angle was changing.

are between 6 and 10.47 dB.

5.6 Discussion

The experimental data is within 0.89 dB of the model prediction for values of $|\theta| \leq$ 15 degrees and $|\phi| \leq 15$ degrees. This means we are correctly predicting signal strength to within a factor of 23 %. The transmitter model error grows to 2.9 dB at $\theta = 45$ degrees. At this point it is off by a factor of 2 but it still captures the right order of magnitude. The receiver model on the other hand is off by a factor of 10 at 45 degrees and does not perform much better at 30 degrees.

Potential sources of error are the pool walls and water surface because they can reflect the signal and cause distortions to the measurements. However such an error would cause an increase in measured signal strength. Since we observed mostly lower experimental measurements than what the model predicted it is unlikely that the walls were a major source of errors.

We believe that more likely causes are yet un-modeled geometrical constraints in the modems. For the transmitter the LEDs are close to the viewport and there is sufficient space on the sides for the signal to spread, thus we observe a good fit. However, in the receiver the stack of viewport, filter, and screens to mask ambient light, creates shadows as the light comes in at an angle. This effect is amplified by the refraction at the viewport, which increases the incident angle significantly. For example, a signal arriving at 30 degrees is refracted by the viewport to travel at an angle of 41.8 degrees inside the modem. This is likely to be at least partially occluded by the screens inside the detector. Additionally we have not yet modeled the behaviors of the fluorescence bandpass filter used, which starts reflecting a large portion of the signal already at low angles.

Chapter 6

Dual-Medium Data Muling for Underwater Observatories

In this chapter we present how all the pieces presented in this thesis come together to form a closed system aimed at autonomous data retrieval from underwater sensor nodes. This is a major milestone in the development of underwater observatories that will allow easy expansion of existing observatories without the need for cabling. In particular, in this chapter we present both, the system integration of AMOUR VI with the AquaOptical II and UNET2 modems, as well as the algorithms needed to perform data muling in an underwater observatory. The presented data muling system is then experimentally evaluated. AMOUR VI is used as a mobile node and PANDA is used as a stationary node. The UNET 2 modem is used for acoustic communication and the AquaOptical II modem for optical communication.

This chapter first describes an intuitive gradient descent controller for data muling and then an improved controller based on a particle filter. Both algorithms are evaluated experimentally using the above system at the end of the chapter.

6.1 Autonomous, Localization-Free Underwater Data Muling using Acoustic and Optical Communication

Our goal is to develop technologies that enable users to interact with ocean observatories. In an ocean observatory robots and in-situ sensors collect information about the underwater environment and deliver this information to remote users, much like a web-cam delivers remote data to users on the ground. In this thesis we focus on developing effective technologies for wireless data transmission underwater. When the amount of data from an ocean observatory is large (e.g. in the case of image feeds), low-bandwidth acoustic communication is not adequate. We instead propose using optical data muling with a robot equipped with an optical modem that can retrieve data fast from underwater nodes with line-of-site connection to the robot. An important problem is locating the underwater sensor node. When distances between the robot and the nodes are large, and their locations are unknown, positioning the data muling robot within optical communication range is challenging. In this chapter we present a solution to autonomous data muling underwater, where the node's location is unknown. The algorithm has three phases. In the first phase, acoustic communication is used to bring the data muling robot within some close range of the desired sensor where it can detect the optical signal. In the second phase, the robot does a local search using the optical signal strength to precisely locate the sensor and position itself within communication range. In the third phase the robot uses optical communication to collect the data from the sensor. In practice, phase two and three overlap once the signal strength becomes strong enough to transmit data.

We implemented and experimentally evaluated the data muling system described in this chapter. We use a new version of the Pop-up Ambient Noise Data Acquisition sensor node called UNET-PANDA, which is presented, along with the acoustic modem used, in [25]. For simplicity UNET-PANDA is referred to as PANDA in the remainder of the chapter. The optical modem has been described in [42]. The Autonomous Modular Optical Underwater Robot (AMOUR) was presented in [45]. Our implementation of the data muling system was repeatedly able to acoustically locate the sensor node from distances of 25 m and 100 m and to download a 1.2 MB data file optically once the node was found.

6.2 Problem Formulation

We consider a sensor node that is deployed at a fixed location on the seafloor. We assume that the sensor node is equipped with an acoustic modem and an optical modem. We use the acoustic modem for low data rate ($\leq$ 1 Kbps) and long-range ($\geq$ 100 m) communications. We use the optical modem for high data rate ($\geq$ 1 Mbps) and short-range ($\leq$ 100 m) communications. We do not require a precise external positioning system but we assume that a coarse location estimate of the node exists. By coarse we mean that the margin of error for this position estimate is within the acoustic communication range. This is usually on the order of hundreds of meters to a few kilometers, though acoustic communication ranges of over 100 km are possible [105]. Examples in which such a situation can arise are (1) when a node is deployed in deep waters from a boat and drifts before it finally reaches the ocean

Algorithm 3 Acoustic Stochastic Gradient Descent

```
$YAW_{robot} \Leftarrow \pi * Random(-1.0...1.0)$
$SPEED_{robot} \Leftarrow 0.25$ Knots forward
$RANGE_{th} \Leftarrow$ inf
while No optical link available do
    Receive $RANGE_m$
    if $RANGE_m \geqslant RANGE_{th} + 1m$ then
        $YAW_{robot} \Leftarrow YAW_{robot} + \pi + \pi * Random(-0.5...0.5)$
        $RANGE_{th} \Leftarrow RANGE_m$
    end if
    if $RANGE_m < RANGE_{th}$ then
        $RANGE_{th} \Leftarrow RANGE_m$
    end if
end while
Begin Optical Gradient Descent
```

Algorithm 4 Optical Stochastic Gradient Descent

```
Retain $YAW_{robot}$ from Algorithm I.
Retain $SPEED_{robot}$ from Algorithm I.
$SSI_{th} \Leftarrow 0$
while Optical Link Established do
    Wait 0.25 Seconds. Measure $SSI_m$
    if $SSI_m \leqslant 0.9 * SSI_{th}$ then
        $YAW_{robot} \Leftarrow YAW_{robot} + \pi + \pi * Random(-0.5...0.5)$
        $SSI_{th} \Leftarrow SSI_m$
    end if
    if $SSI_m > SSI_{th}$ then
        $SSI_{th} \Leftarrow SSI_m$
    end if
end while
Switch back to Acoustic Gradient Descent
```

floor; (2) when a node is deployed by an autonomous underwater vehicle using dead reckoning the placement can have a large error; (3) when a node is not rigidly moored and its position changes with time because of water currents.

Further, we assume that an autonomous hovering underwater vehicle (AUV) is equipped with identical acoustic and optical modems and capable of communicating through these with the sensor node when in range. Hovering enables the vehicle to hold its attitude and depth statically and to execute surge (forward / backward) velocity commands.

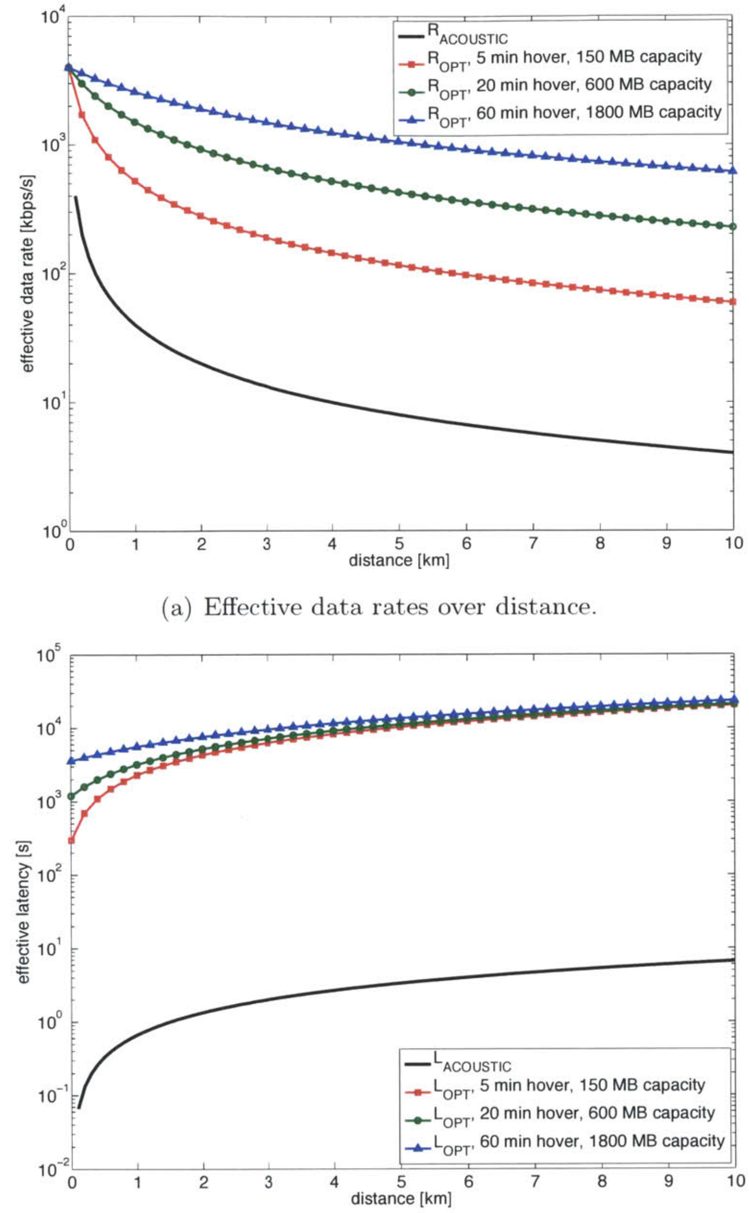
Our problem statement concerns the case in which the sensor node is collecting data at a faster rate than can be transmitted using the long distance acoustic channel. Kilfoyle *et al.* show empirically that the product of acoustic communication rate and bandwidth rarely exceeds 40 km-Kbps for state of the art acoustic modems [70]. For a single sensor separated by 5 km from the user this would result in a communication rate of 8 Kbps. If we consider an application that collects ambient acoustic signals or video our data stream will far exceed the available acoustic channel capacity.

6.3 Theoretical Performance of Data Muling

Figure 6-1(a) shows the upper bounds of achievable data rates for our approach plotted against the travel distance for the AUV (one way). The black line shows the achievable data rates using the acoustic channel computed as

$$R_{acoustic}(d) = \frac{40,000 \text{ m-Kbps}}{d}.$$

according to Kilfoyle *et al.* [70]. In color are shown the upper bounds for different time intervals that the AUV spends above the sensor nodes. We compute the upper bounds of the effective data rates for data muling as follows. We assume the AUV travels on a direct path (best possible case) to the sensor node at a speed of $v_{AUV} = 1$ m/s. This is the speed at which AMOUR can travel. We assume the optical data rate to be



(a) Effective data rates over distance.

(b) Effective latencies over distance.

Figure 6-1: The left graph shows effective data rates for given distances between the sensor node and the user. The x-axis shows the distance in km and the y-axis the data rate. Black shows data rates for acoustic communication. The colored lines show data rates for data muling with different times spent hovering above the sensor to download the data. The right graph shows effective latencies for the same cases as in the left figure. The x-axis shows the distance in km and the y-axis the resulting latency in seconds. The latencies are reported as the entire round trip time (worst-case latency).

$r_{OPT} = 4$ Mbps. This is the data rate of AquaOptical as used during the experiments. We call d the distance of the sensor from the user (one way trip in meters) and t_{hover} the time that the robot hovers above the sensor node to download the data. Under these assumptions the total travel time of the robot from the user to the node and back, including the time spent hovering above the sensor node, is

$$t_{travel}(d) = t_{hover} + \frac{2 \cdot d}{v_{AUV}}$$

This results in an effective data rate of

$$r_{OPT,t_{hover}}(d) = \frac{t_{hover} \cdot r_{OPT}}{t_{travel}(d)}.$$

Figure 6-1(b) shows the resulting latencies under the same assumptions. For the acoustic communication we compute the latency as travel time of sound in water, i.e.

$$L_{acoustic}(d) = \frac{d}{1,500 \text{ m/s}}$$

The optical latency is equivalent to $t_{travel}(d)$. Figure 6-1(a) shows that the achievable data rate when using data muling far exceeds the currently available acoustic data rates. This effect can even be amplified by using multiple AUVs that can travel in parallel to either a single or multiple sensor nodes. Using acoustic communication neighboring nodes often have to share the medium reducing the effective data rate per node. The disadvantage of data muling is its higher latency as seen in Fig. 6-1(b).

6.4 Acoustic and Optical Gradient Descent

We developed a combined acousto-optical communication network capable of large scale data recovery that does not require precise localization of the robot nor the sensor node. The robot uses acoustic communication and ranging to come close to the sensor node. High bandwidth optical channel allows the robot to download the payload data. More specifically, our approach to data muling is as follows:

1. We use acoustic ranging between the robot and the PANDA sensor node. The acoustic modem on the PANDA transmits a ranging beacon every 6 s that is received by the acoustic modem on the robot and provides it with a range measurement. The robot uses the stochastic gradient descent algorithm shown in Algorithm 3 to travel close to the PANDA.

2. At all times the PANDA is streaming the payload data using the optical modem and random linear rateless erasure codes known as Luby transform (LT) codes [78]. The payload data is a random file consisting of 2048 blocks of 576 bytes. The LT-Codes require on average an overhead of 3 %, so about 2109 packets have to be received by the robot to decode the entire file. Because of the nature of the LT-Codes it does not matter which packets are received.

3. Once the robot is close enough to the sensor node to receive an error-free packet it switches from acoustic gradient descent to maintaining position using the optical gradient descent algorithm described in Algorithm 4. If the optical connection breaks at any point in time we return to step 1.

4. While AMOUR is in optical communication range every packet is used to (a) measure the signal strength and (b) decode the payload data if the CRC matches.

5. The experiment is considered to have completed successfully once the entire 1.2 MB file has been received and decoded by the robot. In a real world scenario the robot would now continue on the approximate location of the next sensor node to begin acoustic gradient descent there.

6.5 Acoustic Particle Filter

The stochastic gradient descent approach described in Section 6.4 has no memory of previous decisions. The only state variables are the current heading and a range threshold used to make the decision whether to keep going straight or to turn. When the algorithm encounters an increasing range it changes the direction of the robot in a random direction at least 90 degrees different from the current direction of travel. This new choice of direction takes into account only the most recent few measurements as reflected in the threshold stored. Because so little information is taken into account, bad choices are made frequently. Further, even when the robot is moving in the right direction a single spurious measurement caused by noise can make it veer of the correct course. The algorithm will recover from this mistake with high probability as the ranges will keep increasing from here on, but this comes at the cost of time and energy. It also causes a large variance in the time that it takes to find the target sensor node.

A more effective algorithm should keep a belief of where the robot is relative to the

Algorithm 5 Acoustic Particle Filter

```
$YAW_{robot} \Leftarrow \pi * Random(-1.0...1.0)$,   $SPEED_{robot} \Leftarrow 0.25$ Knots forward
Receive first measurement $RANGE_m$
for $k = 1 \ldots N$ do
    $\begin{bmatrix} X_k \\ Y_k \end{bmatrix} = RANGE_m \cdot \begin{bmatrix} \cos(\alpha_k) \\ sin(\alpha_k) \end{bmatrix}$, where   $\alpha_k = Random(-\pi \ldots \pi)$
end for
while No optical link available do
    for $k = 1 \ldots N$ do
        Independently draw $e_x$ and $e_y$ from $\mathcal{N}(0.0, \sigma_{robot})$
        $\begin{bmatrix} X_k \\ Y_k \end{bmatrix} = \begin{bmatrix} X_k \\ Y_k \end{bmatrix} + SPEED_{robot} \cdot \begin{bmatrix} \cos(YAW_{robot}) \\ sin(YAW_{robot}) \end{bmatrix} + \begin{bmatrix} e_x \\ e_y \end{bmatrix}$
    end for
    if Received new measurement $RANGE_m$ then
        for $k = 1 \ldots N$ do
            $W_k = \frac{1}{\sqrt{2 \cdot \pi \cdot \sigma_{range}}} \cdot \exp(\frac{-D_k^2}{2 \cdot \sigma_{range}})$,   where   $D_k = RANGE_m - \sqrt{X_k^2 + Y_k^2}$
        end for
        RESAMPLE particles using weights $W_k, k \in \{1 \ldots N\}$.
    end if
    $\begin{bmatrix} \bar{Z}_X \\ \bar{Z}_Y \end{bmatrix} = \frac{1}{N} \sum_{n=1}^{N} \begin{bmatrix} -X_k \\ -Y_k \end{bmatrix}$,   $\mu_\theta = \text{atan}(\bar{Z}_Y, \bar{Z}_X)$,   $\sigma_\theta = \sqrt{\ln(1/\|\bar{Z}\|)}$
    $YAW_{robot} = \mu_\theta + \sigma_\theta/4$
end while
Begin Optical Gradient Descent
```

sensor node and update this belief with every measurement. An Extended Kalman Filter (EKF) delivers such a behavior. It represents the current belief of the robot's location as a mean and covariance. Because of this it needs to be initialized, for example by performing a circular maneuver such as in [109]. Further, because we are representing the robot's state with a multidimensional Gaussian, we cannot represent multimodal distributions, for example when we have a baseline ambiguity because our vehicle has been traveling straight.

In order to represent multimodal distributions we implement a particle filter algorithm, as shown in Algorithm 6. The filter represents the current belief using N particles. The sensor node position is assumed to be fixed at the origin. The filter localizes the robot relative to the sensor node. Each particle stores one guess $[X_k\ Y_k]$ of the robot's location. The algorithm initializes the particles when the first range measurement R_m is received by randomly placing all of the particles on a circle of radius R_m around the origin (lines 3 to 5). Once the particles have been initialized, we perform a prediction step every $100ms$, taking into account robot movement noise with a standard deviation σ_{robot} (lines 7 to 10). When a new range measurement is received we compute the probability of observing such a range for every particle taking into account the measurement noise σ_{range}. The particles are then re-weighted according to the algorithm presented in Thrun *et al.* [106]. At the end of every iteration we compute a new heading for the robot. This is done by computing the heading required for every particle to travel towards the node. We assume these headings form a wrapped normal distribution and we compute its mean and standard deviation (line 17). Setting the robot heading to the mean would result in the particles traveling directly towards the node and can create a baseline ambiguity. Thus, we chose to set the new headings as the mean plus the standard deviation divided by a factor of 4. The more uncertain the particle filter is, the more the robot will deviate from the straight path, and this in turn helps resolve the baseline ambiguity.

Algorithm 6 Acousto-Optical Particle Filter

```
$N \in \mathcal{N},\ k \in [1 \dots N],\ \theta \in [-\pi \dots \pi],\ v, x_k, y_k, r_m, s_m \in \mathcal{R}$
$\sigma_{robot}, \sigma_{range}, \sigma_{opt} \in \mathcal{R}$
$\mathbf{p}_k = \begin{bmatrix} x_k \\ y_k \end{bmatrix},\ \mathbf{u}(\theta) = \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix}$
$\theta = Random(-\pi \dots \pi),\ v = 0.25$ Knots forward
Receive first range measurement $m_r$
for $k = 1 \dots N$ do
    $\mathbf{p}_k = m_r \cdot \mathbf{u}(\alpha_k)$, where $\alpha_k = Random(-\pi \dots \pi)$
end for
while true do
    for $k = 1 \dots N$ do
        Independently draw $e_x$ and $e_y$ from $\mathcal{N}(0.0, \sigma_{robot})$
        $\mathbf{p}_k = \mathbf{p}_k + v \cdot \mathbf{u}(\theta) + \begin{bmatrix} e_x \\ e_y \end{bmatrix}$
    end for
    if Received new range measurement $m_r$ then
        for $k = 1 \dots N$ do
            $w_k = \frac{1}{\sqrt{2 \cdot \pi \cdot \sigma_{range}}} \cdot \exp(\frac{-d_k^2}{2 \cdot \sigma_{range}})$, where $d_k = m_r - \|\mathbf{p}_k\|$
        end for
        RESAMPLE particles using weights $w_k, k \in \{1 \dots N\}$.
    end if
    if Received new optical signal stength $m_o$ then
        $w_k = \frac{1}{\sqrt{2 \cdot \pi \cdot \sigma_{opt}}} \cdot \exp(\frac{-\|\mathbf{p}_k\|^2}{2 \cdot \sigma_{opt}})$
        RESAMPLE particles using weights $w_k, k \in \{1 \dots N\}$.
    end if
    if Received new optical signal stength gradient $m_g$ then
        $a = \text{acos}\left(\frac{\mathbf{p}_k \cdot \mathbf{u}(\theta)}{\|\mathbf{p}_k\| \cdot \|\mathbf{u}(\theta)\|}\right)$
        $w_k = \begin{Bmatrix} 1 + a & \textbf{if}\, m_g > 0 \\ \frac{1}{1+a} & \textbf{if}\, m_g < 0 \end{Bmatrix}$
        RESAMPLE particles using weights $w_k, k \in \{1 \dots N\}$.
    end if
    $\bar{\mathbf{z}} = \begin{bmatrix} \bar{\mathbf{z}}_x \\ \bar{\mathbf{z}}_y \end{bmatrix} = \frac{1}{N} \sum_{n=1}^{N} -\mathbf{p}_k$
    $\mu_\theta = \text{atan}(\bar{\mathbf{z}}_y, \bar{\mathbf{z}}_x),\quad \sigma_\theta = \sqrt{\ln(1/\|\bar{\mathbf{z}}\|)}$
    $\theta = \mu_\theta + \sigma_\theta / 4$
end while
```

6.6 Acoustic and Optical Particle Filter

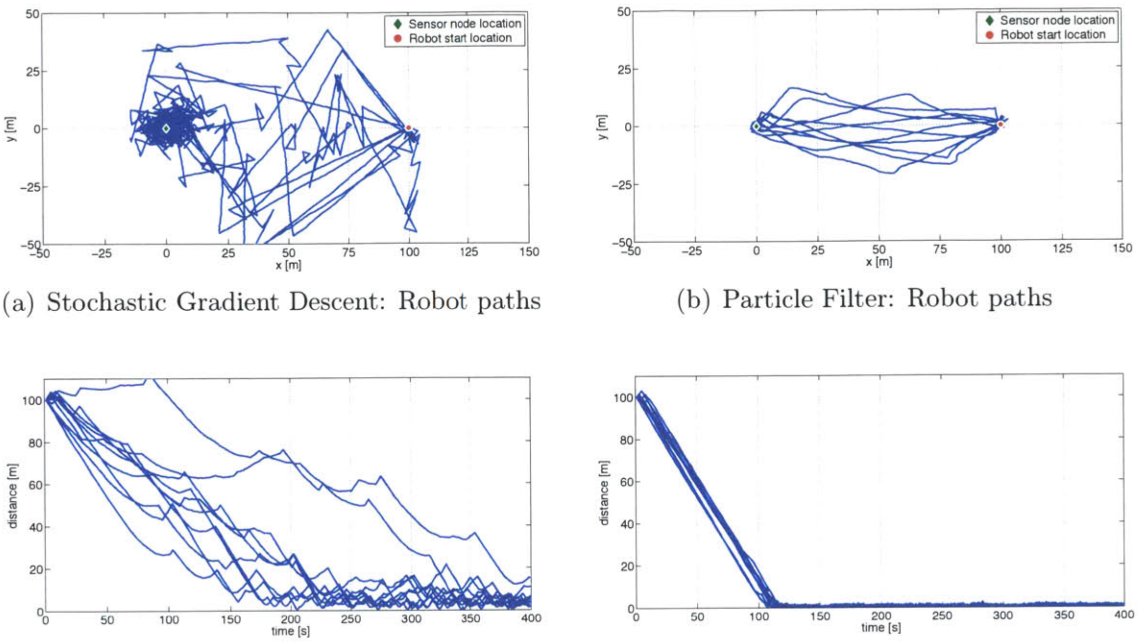
In order to represent multimodal distributions we implement a particle filter algorithm, as shown in Algorithm 6. The filter represents the current belief using N particles. The sensor node position is assumed to be fixed at the origin. The filter localizes the robot relative to the sensor node. Each particle stores one guess $[X_k\ Y_k]$ of the robot's location. The algorithm initializes the particles when the first range measurement R_m is received by randomly placing all of the particles on a circle of radius R_m around the origin (lines 3 to 5). Once the particles have been initialized, we perform a prediction step every $100ms$, taking into account robot movement noise with a standard deviation σ_{robot} (lines 7 to 10). When a new range measurement is received we compute the probability of observing such a range for every particle taking into account the measurement noise σ_{range}. The particles are then re-weighted according to the algorithm presented in Thrun *et al.* [106]. At the end of every iteration we compute a new heading for the robot. This is done by computing the heading required for every particle to travel towards the node. We assume these headings form a wrapped normal distribution and we compute its mean and standard deviation (line 17). Setting the robot heading to the mean would result in the particles traveling directly towards the node and can create a baseline ambiguity. Thus, we chose to set the new headings as the mean plus the standard deviation divided by a factor of 4. The more uncertain the particle filter is, the more the robot will deviate from the straight path, and this in turn helps resolve the baseline ambiguity.

6.7 Simulations

We evaluated both the acoustic stochastic gradient descent algorithm (Algorithm 3), and the acoustic particle filter algorithm (Algorithm 6) in simulation. In each simulation the robot state was represented as $[X_{robot}\ Y_{robot}\ YAW_{robot}]$. The robot was simulated with a constant speed of $SPEED_{robot} = 1$ m/s. Independent white Gaussian noise with a standard deviation σ_{robot} in meters was added to the robot's position every second to simulate movement errors. Thus, every second the new robot position was computed as

$$\begin{bmatrix} X_{robot}(t+1) \\ Y_{robot}(t+1) \end{bmatrix} = \begin{bmatrix} X_{robot}(t) \\ Y_{robot}(t) \end{bmatrix} + SPEED_{robot} \cdot \begin{bmatrix} \cos(YAW_{robot}) \\ sin(YAW_{robot}) \end{bmatrix} + \begin{bmatrix} e_x \\ e_y \end{bmatrix}$$

where e_x and e_y are independently drawn from $\mathcal{N}(0.0, \sigma_{robot})$. Measurements were simulated every second with added Gaussian noise with a standard deviation σ_{range}. Each new measurement $RANGE_m$ is computed as



(a) Stochastic Gradient Descent: Robot paths

(b) Particle Filter: Robot paths

(c) Stochastic Gradient Descent: Distances

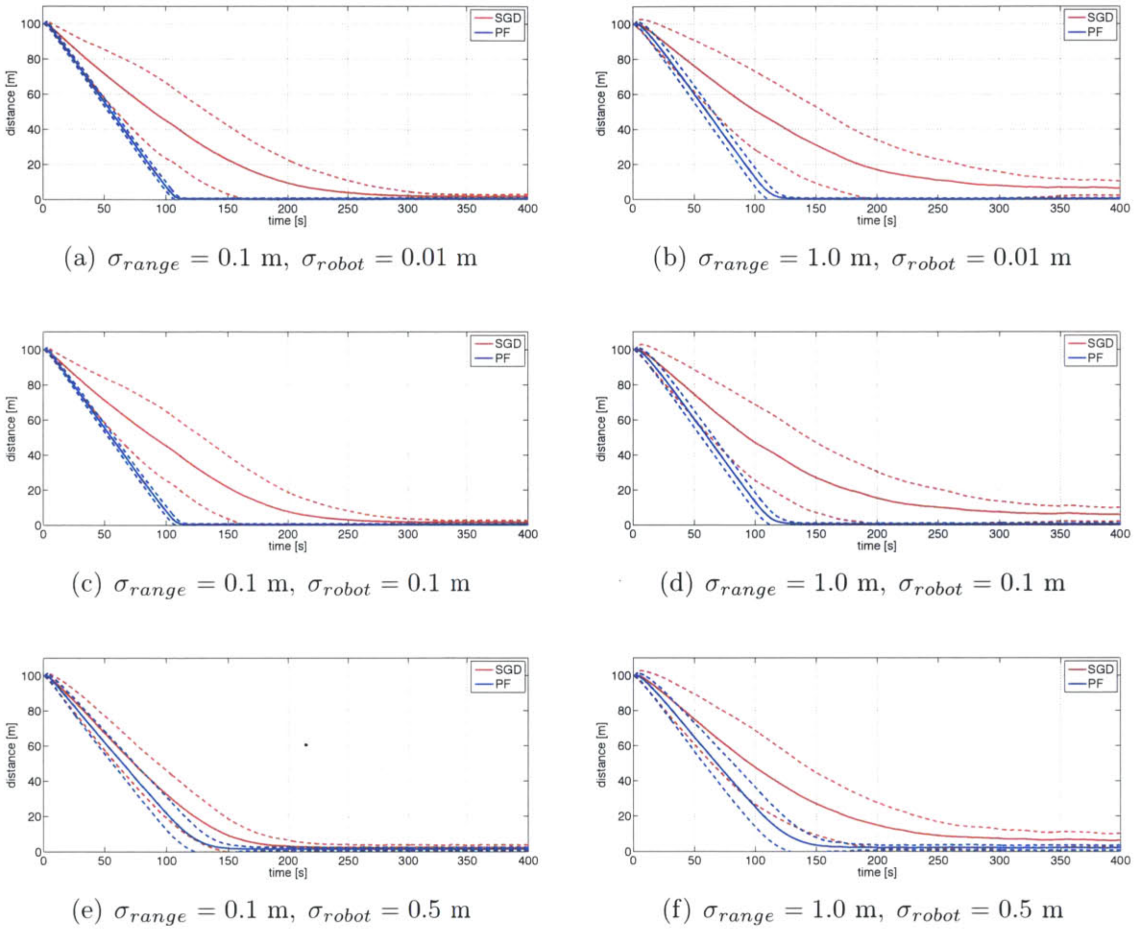
(d) Particle Filter: Distances

Figure 6-2: Sample simulation results of 10 stochastic gradient descents and 10 particle filter node localizations using acoustic ranging in both cases. All simulations were performed with a robot speed of 1 m/s, measurement noise $\sigma_{range} = 1$ m and robot motion noise $\sigma_{robot} = 0.1$ m. Range measurements occurred every 1 s. Plots (a) and (b) show the resulting robot paths. The x and y-axes show displacement in meters. The sensor node is located at the origin (green circle) and the robot starts at location (0 m, 50 m) (red diamond). Each continuous blue line denotes one simulation run. Plots (c) and (d) show the corresponding distances of the robot from the sensor node in m on the y-axis over time in seconds on the x-axis.

$$RANGE_m = \sqrt{X_{robot}^2 + Y_{robot}^2} + e_r$$

where e_r is drawn from $\mathcal{N}(0.0, \sigma_{range})$.

Figure 6-2 shows two sets of 10 simulated paths taken by the robot using stochastic gradient descent (Fig. 6-2(a)) and a particle filter (Fig. 6-2(b)). The simulations were performed using Algorithm 3 and 6. The parameters for these simulations were $\sigma_{range} = 1$ m and $\sigma_{robot} = 0.1$ m. These plots visualize the characteristic difference in paths generated by the stochastic gradient descent and the particle filter. When the stochastic gradient descent encounters an increasing range, it picks a new direction almost entirely at random. The particle filter, on the other hand, continuously merges all gathered information about the sensor node location and continuously updates the robot's heading resulting in a more direct path. Plotted in Figures 6-2(c) and 6-2(d) are the corresponding distances of the robot to the sensor node over time.



(a) $\sigma_{range} = 0.1$ m, $\sigma_{robot} = 0.01$ m

(b) $\sigma_{range} = 1.0$ m, $\sigma_{robot} = 0.01$ m

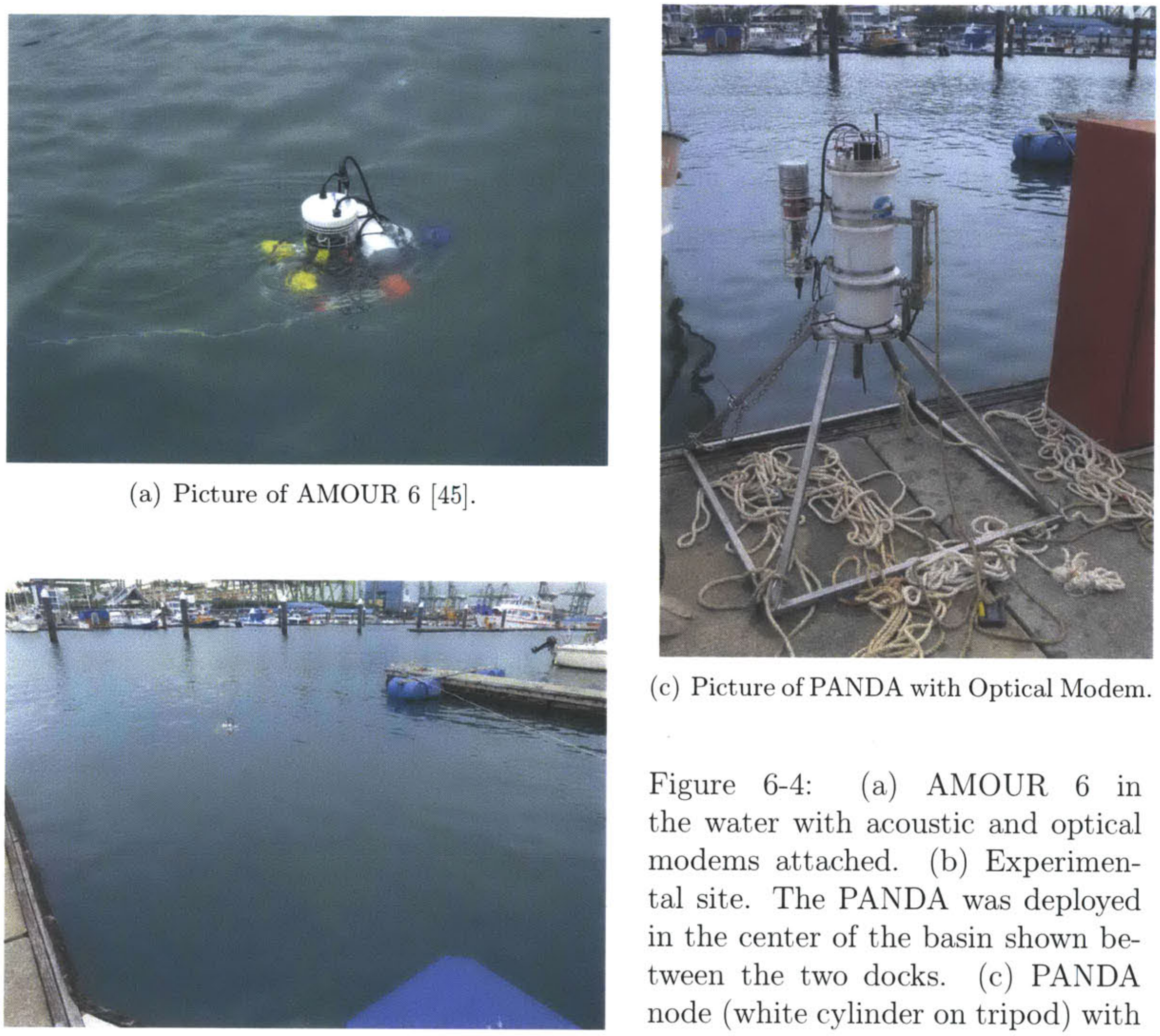
(c) $\sigma_{range} = 0.1$ m, $\sigma_{robot} = 0.1$ m

(d) $\sigma_{range} = 1.0$ m, $\sigma_{robot} = 0.1$ m

(e) $\sigma_{range} = 0.1$ m, $\sigma_{robot} = 0.5$ m

(f) $\sigma_{range} = 1.0$ m, $\sigma_{robot} = 0.5$ m

Figure 6-3: Simulation results. Each plot corresponds to one of six different choices for $(\sigma_{range}, \sigma_{robot})$, the measurement noise and robot motion noise used in the simulation. All simulations were performed with a robot speed of 1 m/s and range measurements occurred every 1 s. The x-axis corresponds to time in seconds and the y-axis corresponds to the distance in meters of the simulated robot to the sensor node. Plotted in red are the results of stochastic gradient descent and in blue the results of the particle filter. For each algorithm and choice of $(\sigma_{range}, \sigma_{robot})$ 1000 simulations were performed for a total of 12000 simulations. The solid lines correspond to the mean, the dotted lines represent the 1σ boundaries.



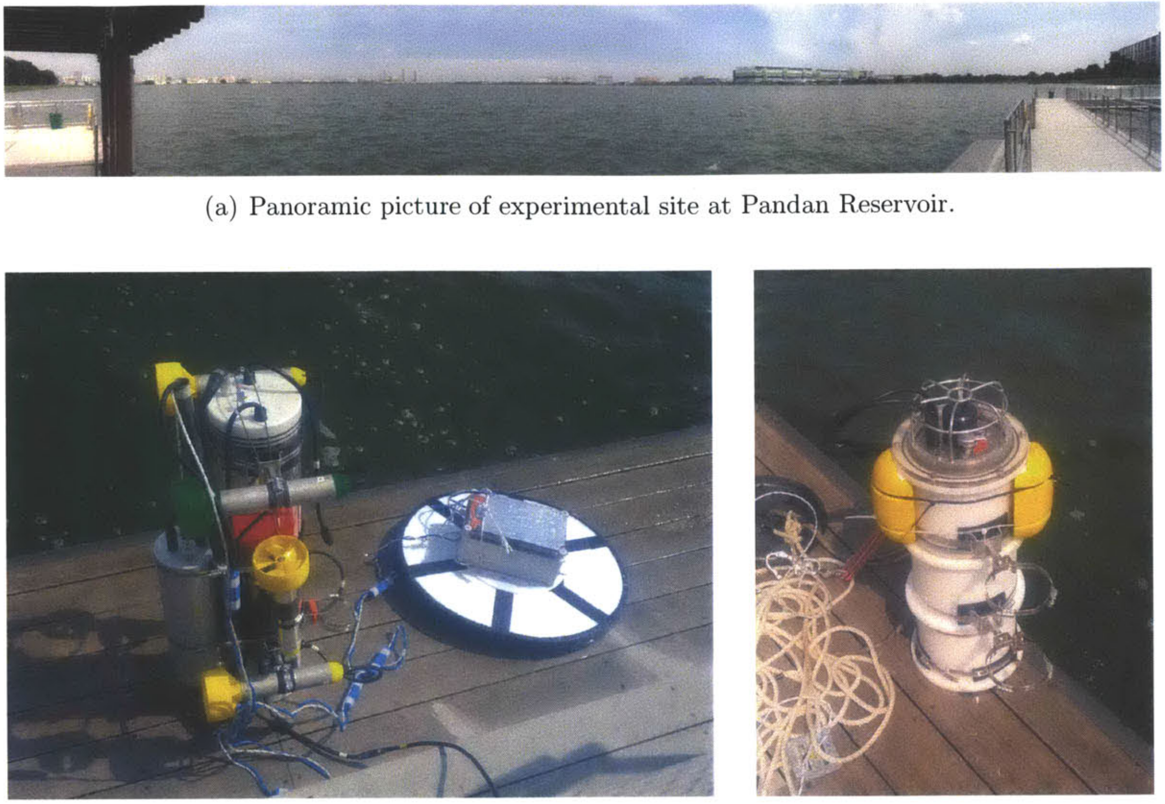
(a) Picture of AMOUR 6 [45].

(b) Picture of experimental site.

(c) Picture of PANDA with Optical Modem.

Figure 6-4: (a) AMOUR 6 in the water with acoustic and optical modems attached. (b) Experimental site. The PANDA was deployed in the center of the basin shown between the two docks. (c) PANDA node (white cylinder on tripod) with Optical Modem attached on the left.

Finally we conducted six sets of simulation runs with parameters chosen as $(\sigma_{range}, \sigma_{robot}) \in \{0.1 \text{ m}, 1.0 \text{ m}\} \times \{0.01 \text{ m}, 0.1 \text{ m}, 1.0 \text{ m}\}$. For each set we simulated 1000 runs using the stochastic gradient descent algorithm and 1000 runs using the particle filter. Figure 6-3 shows the results for all these runs grouped in six plots according to parameter choices. Each plot shows the mean distance over time (solid line) with 1σ boundaries (dashed lines). The stochastic gradient descent results are plotted in red and the particle filter results are plotted in blue. In all six cases the particle filter outperforms the stochastic gradient descent. When the noise is low (Figures 6-3(a) and 6-3(c)), the particle filter takes on average 104 s for the robot to come to within 5 m of the sensor node. This is only 10 % more then the theoretical minimum, which is 95 s



(a) Panoramic picture of experimental site at Pandan Reservoir.

(b) Picture of AMOUR 6 and floating Wifi.

(c) Picture of PANDA.

Figure 6-5: (a) Experimental site at Pandan Reservoir in Singapore. (b) AMOUR 6 with acoustic modem attached with the WiFi buoy next to the robot. (c) PANDA node with floats and weight.

since the robot starts 95 m away and travels at 1 m/s.

6.8 Experimental Setup

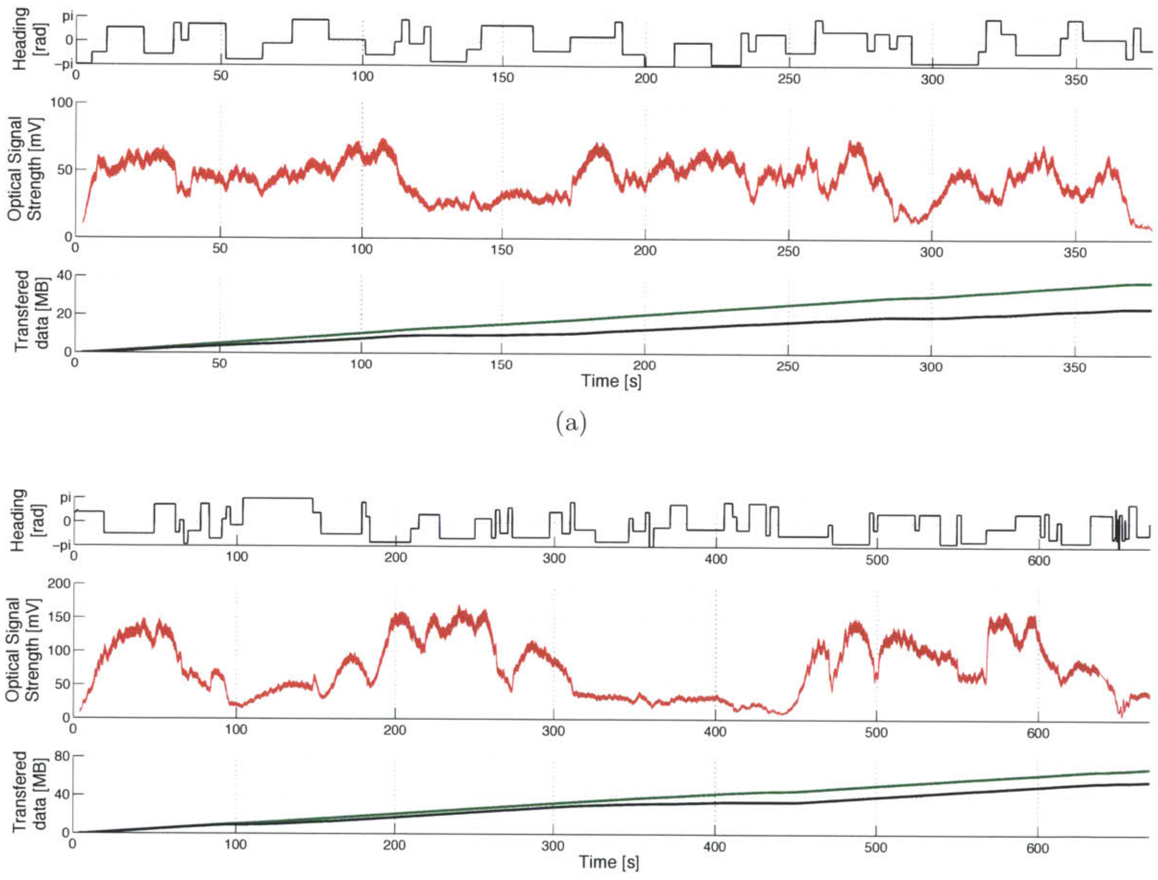
For these experiments we used the PANDA sensor node in combination with AuqaOptical II. The acoustic modem on the PANDA was configured to transmit an 18 byte long ranging beacon every 6 s that is received by the acoustic modem on the robot and provides it with a range measurement. The AquaOptical II modem was configured such, that each optical packet transmitted contained 576 bytes payload data plus 32 bytes of configuration data (i.e. source and destination address, packet size, 32-bit

CRC checksum, degree and seed used for the LT-Codes).

During the experiments two different hardware configurations were used. In the first configuration one acoustic and one optical modem were attached to the robot and it was configured with 5 thrusters allowing it to maintain attitude while traveling at a forward speed of 0.25 m/s. This configuration is shown in the water in Figure 6-4(a). One optical modem was attached to the PANDA node as shown in Figure 6-4(c). In the second configuration no optical modems were used. The robot carried one acoustic modem and had 6 thrusters attached. This allowed the robot to travel at a forward speed of 0.5 m/s. The robot and the PANDA can be seen in Figures 6-5(b) and 6-5(c).

We conducted 4 sets of experiments to demonstrate the system's ability to localize a sensor node using a robot and recover data from it. The first two sets of experiments used the gradient descent algorithms presented in Section 6.4. The third set of experiments was conducted to evaluate the performance of the acoustic particle filter algorithm presented in Section 6.5. The fourth and final set of experiments was conducted to evaluate the performance of the combined acoustic and optical particle filter algorithm presented in Section 6.6. In this chapter we did not focus on the return of the robot to the base station.

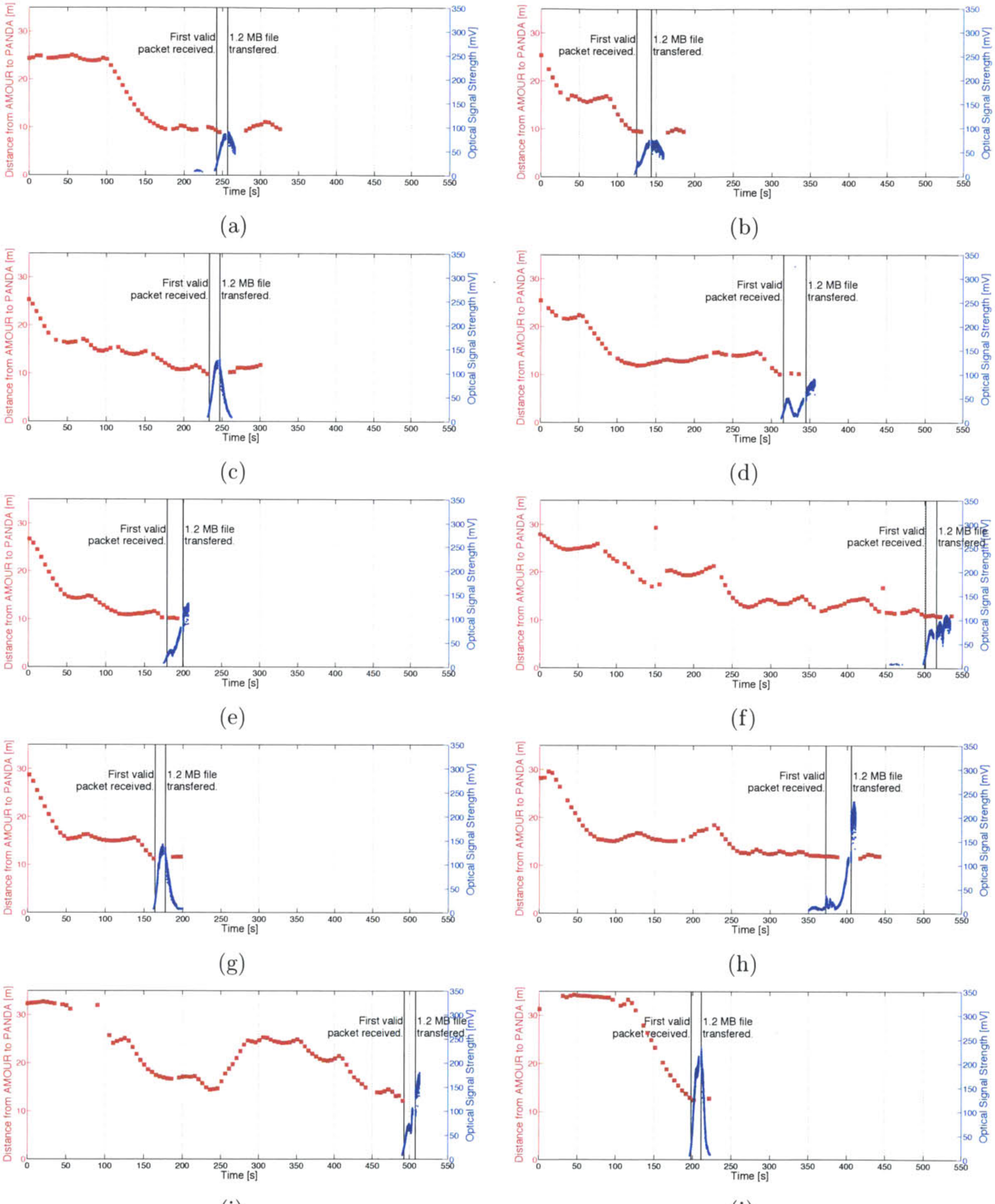
The first two sets of experiments were conducted from a dock at the Republic of Singapore Yacht Club (Figure 6-4(b)). The water depth was about 7 m and we estimated visibility at about 2 m. The PANDA with the optical modem was mounted on a tripod to guarantee that they should be pointing upright after being lowered to the ground. This setup can be seen in Figure 6-4(c). Our vehicle AMOUR carrying the acoustic and optical modems can be seen in Figure 6-4(a). It was tethered for data collection and security, but operated autonomously during the experiments. The robot speed was set at about 0.25 m/s to ensure safe operation and to keep distance changes at a reasonable rate between updates. Generating the LT-Codes as described in Section 6.4 requires substantial computation. Because of this we needed to reduce the number of packets transmitted from the optical modem on the PANDA to 392 packets a second. Including overhead this corresponds to a bit rate of 1.84 Mbps. The remainder of the optical channel (2.16 Mbps) was not utilized.



(a)

(b)

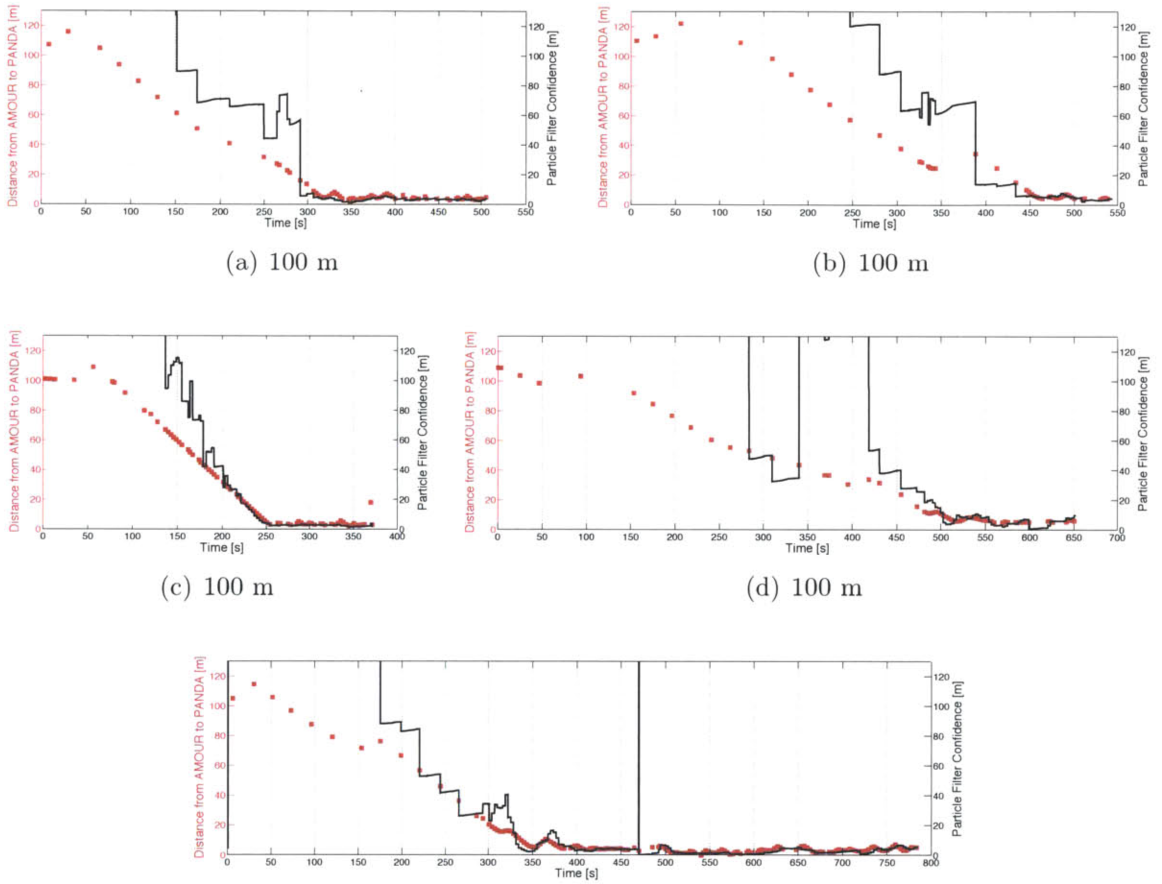
Figure 6-6: Results of two optical gradient descent experiment runs. The first run is plotted in (a), the robot operated at the water surface. The second run is plotted in (b), the robot operated at a depth of 1.5 m. At the beginning of each experiment the robot was manually steered close to the PANDA to establish an optical link. The optical gradient descent algorithm then controlled the robot to stay close to the PANDA. In all plots the x-axis indicates the time in seconds since the beginning of the experiment. The top graph for each experiment (black) shows the heading of the robot as computed by the optical gradient descent algorithm. The middle graph (red) shows the measured signal strength. The bottom graph shows the amount of data received in MB (green) and the amount of data received error-free in MB (black). Packet size was 576 bytes with a 4 byte CRC.



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Figure 6-7: Results of data muling experiments. Each graph (a)-(j) shows one experiment. The x-axis shows time in seconds since beginning of the experiment. The red curve (left y-axis) shows the distance between AMOUR and the PANDA node. Each red square corresponds to one range measurement received by AMOUR. The blue curve (right y-axis) shows the optical signal strength between PANDA and AMOUR. A link was established whenever there is non-zero signal strength. Two horizontal black lines mark the receipt of the first error free packet (left line) and the receipt of the final error free packet needed to decode the 1.2 MB test file (right line).

The first set of experiments consisted of manually placing the robot close to the PANDA node and using optical gradient descent to maintain a position close to the PANDA node. This experiment was conducted two times, one time with the robot at the water surface and the second time with the robot keeping a depth of 1.5 m under the water surface. Given a water depth of about 7 m, a height of the PANDA of



(a) 100 m

(b) 100 m

(c) 100 m

(d) 100 m

(e) 100 m

Figure 6-8: Results of acoustic particle filter experiments. Each graph (a)-(e) shows one experiment. The x-axis shows time in seconds since beginning of the experiment. The red curve (left y-axis) shows the distance between AMOUR and the PANDA node. Each red square corresponds to one range measurement received by AMOUR. The black curve (right y-axis) shows the confidence of the particle filter (square root of the determinant of the covariance of all particle positions, lower is better). The spike visible at second 420 of plot (e) was caused by a spurious range measurement. The GPS ground truth for these experiments is plotted in Figure 6-9.

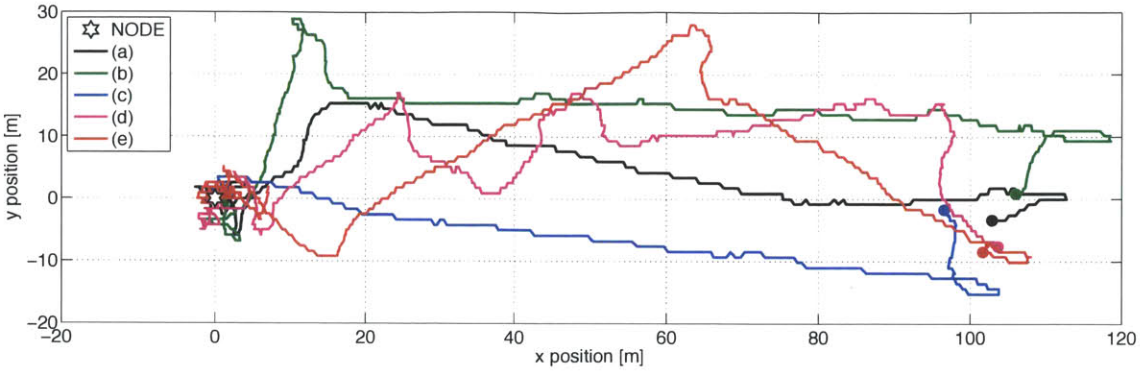
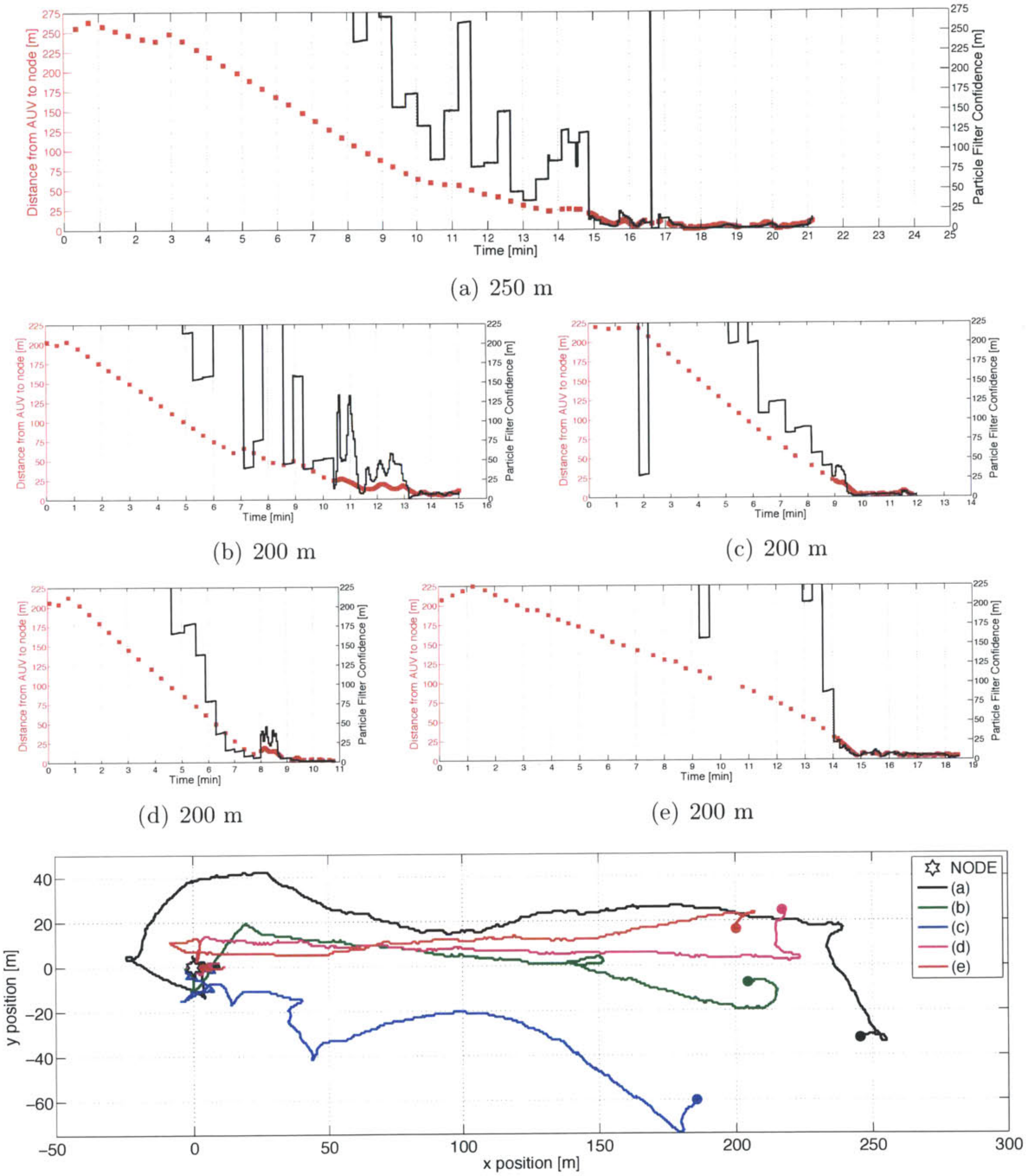


Figure 6-9: GPS ground truth for acoustic particle filter experiments shown in Figure 6-8. All 5 runs were conducted from a starting distance of about 100 m.

about 1 m, and a robot height of slightly below 1 m, the first and second experiment had a minimum distance of 5 m between the optical modems and 3.5 m, respectively.

In the second set of experiments we manually positioned the robot at a distance of about 25 m away from the PANDA, dove it to 2 m depth where we started the acoustic gradient descent algorithm. This experiment was conducted 12 times, of which two were aborted because the robot's tether got entangled with obstacles in the harbor (the robot would dive under the docks sometimes due to the random nature of stochastic gradient descent). We used the other 10 experiments for evaluation.

The third and fourth sets of experiments to measure the performance of the particle filters were conducted at the Pandan Reservoir in Singapore (Figure 6-5(a)). The reservoir covers an area of over 1 km^2 and has a depth of about 4 m close to the shore. During this set of experiments we used a floating buoy that carried a long-range WiFi. The buoy was tethered to the robot with a 5 m long Ethernet cable and additionally secured with a rope. This allowed for remote operation of the robot while it was able to move freely through the water without having a long tether that could get entangled in the many buoys that are present at the reservoir. To prepare the experiments the robot was used to transport the PANDA about 40 m off shore and drop it there. A weight attached to the bottom of the PANDA together with floats attached at the top ensured that it would sink to the bottom of the reservoir but remain upright (Figure 6-5(c)). A rope was permanently attached to the PANDA



(a) 250 m

(b) 200 m

(c) 200 m

(d) 200 m

(e) 200 m

(f) GPS ground truth

Figure 6-10: Results of acoustic particle filter experiments. Each graph (a)-(e) shows one experiment. The x-axis shows time in seconds since beginning of the experiment. The red curve (left y-axis) shows the distance between AMOUR and the PANDA node. Each red square corresponds to one range measurement received by AMOUR. The black curve (right y-axis) shows the confidence of the particle filter (square root of the determinant of the covariance of all particle positions, lower is better). The GPS ground truth for these experiments is plotted in (f).

that allowed us to recover it manually after the experiments. During each experiment the robot was manually positioned at a distance of at least 100 m away from the PANDA and the node localization algorithm based on a particle filter was started. At all times the robot traveled with a speed of 0.5 m/s at a depth of 1 m. At distances greater than 20 m the acoustic beacons were often corrupted by the noise of the robot's thrusters. To alleviate this problem we chose to send beacons every 3 s during this experiment instead of every 6 s as in the previous experiments. Further, if the robot did not receive a valid beacon for more than 20 s it stopped its thrusters significantly reducing the acoustic noise levels. This ensured that even at distances beyond 20m we would receive ranging beacons no less than twice per minute. The experiment was conducted at total of 12 times. The first 10 times were conducted to evaluate the acoustic particle filter presented in Section 6.5 and formed the third set of experiments. In 5 of these experiments the robot was started at a distance of 100 m from the node in 5 runs, at a distance of 200 m in 4 runs, and at distance of 250 m in the final run. The final two experimental runs formed the fourth and last set of experiments, conducted to evaluate the performance of the combined acoustic and optical particle filter presented in Section 6.6. All of these experiments were used for evaluation.

In summary, we conducted the following successful data muling experiments:

1. To evaluate the isolated performance of the optical gradient descent algorithm we performed two experiments from a dock at the Republic of Singapore Yacht Club dock with the robot hovering above the sensor node.

2. To evaluate the combined performance of the acoustic and optical gradient descent algorithm presented in Section 6.4 we conducted 12 experimental runs from the same dock at the Republic of Singapore Yacht Club. In these experiments the robot and to localize the sensor node from a distance of about 25 m and download a 1.2 MB large data file. 2 of these experiments had to be discarded because the robot's tether used for monitoring the experiment got entangled.

3. To evaluate the performance of the acoustic particle filter algorithms presented in Section 6.5 we conducted a total of 10 experiments at the Pandan Reservoir in Singapore. During these experiments the robot had to localize the sensor node from distances of 100 m, 200 m, and 250 m.

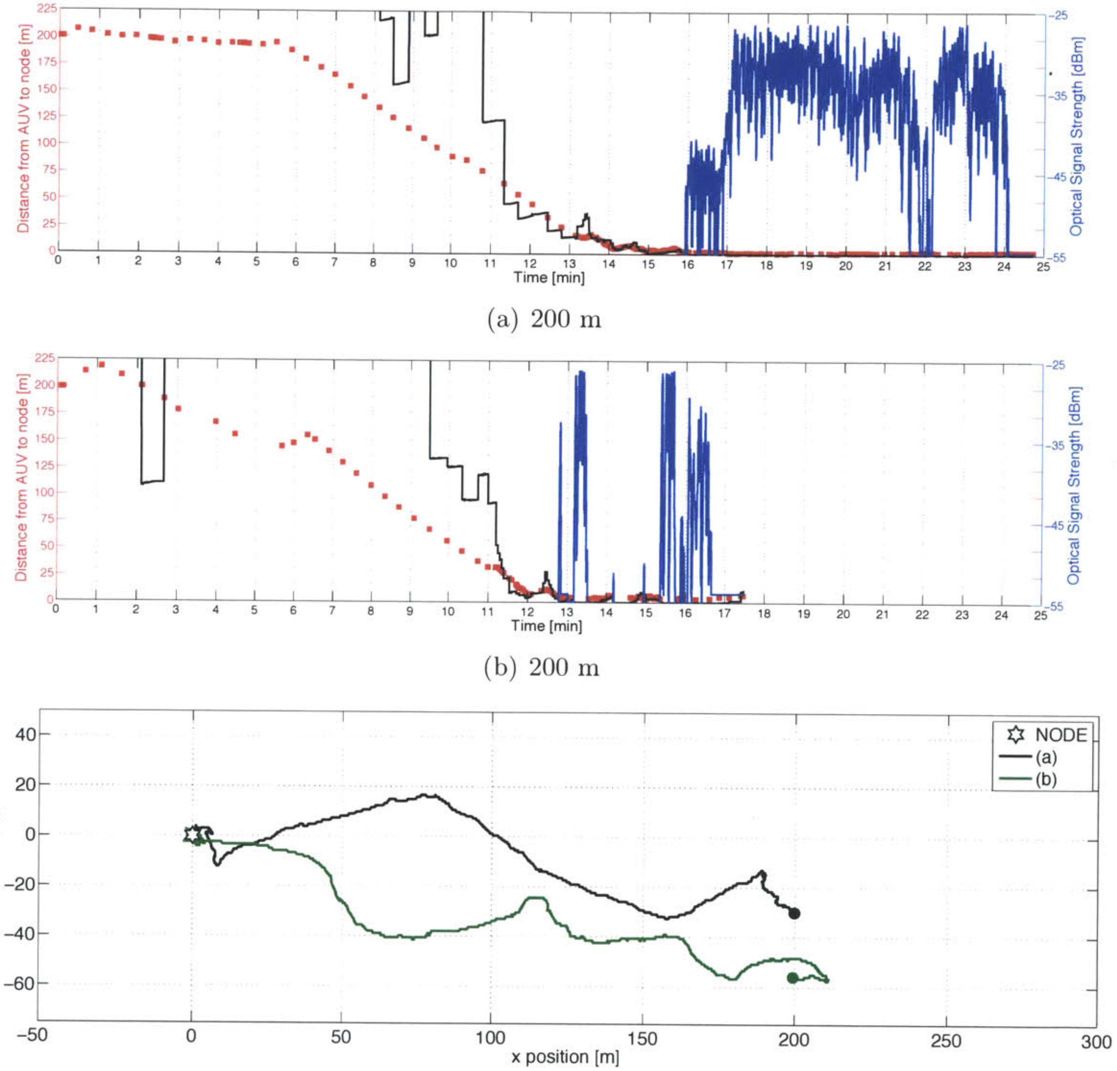
4. To evaluate the performance of the combined acoustic and optical particle filter algorithms presented in Section 6.6 we conducted 2 additional experiments at the Pandan Reservoir. During these experiments the robot had to localize the sensor node from a distance 200 m and establish optical communication.

6.9 Experimental Results

The results of the optical gradient descent experiments can be seen in Figure 6-6. In the first run the robot maintained position for over 6 min before it lost track of the optical signal. During this time the optical modem on the PANDA transmitted 52.6 MB of payload data and on AMOUR received 37.5 MB of which 23.87 MB were error-free packets (one packet was 576 bytes large). In the second run it maintained position successfully for 11 min after which we stopped the experiment. During this time the PANDA transmitted 93.4 MB of payload data and AMOUR received 69.1 MB of which 55.6 MB were error-free packets. The rate of error-free packets was higher in the second run because the robot was operating at a lower depth closer to the transmitter, which resulted in higher signal strength at the receiver.

The results of the second set of experiments can be seen in Figure 6-7. In all 10 experiments the robot successfully found the PANDA within 2.5 to 8 min and proceeded to download the 1.2 MB file within an additional 10 to 35 s.

Figures 6-8, 6-9 and 6-10 show the results of the third set of experiments in which an acoustic particle filter was used. In all 10 experiments the robot successfully found the PANDA, meaning that it came with a distance of 5 m or less, which is sufficient to establish optical communication. For the runs over a distance of 100 m the node was found within 4.2 to 9 min. For the runs over a distance of 200 m the node was



(a) 200 m

(b) 200 m

(c) GPS ground truth

Figure 6-11: Results of combined acoustic and optical particle filter experiments. Graphs (a) and (b) shows one experiment each. The x-axis shows time in seconds since beginning of the experiment. The red curve (left y-axis) shows the distance between AMOUR and the PANDA node. Each red square corresponds to one range measurement received by AMOUR. The black curve (also left y-axis) shows the confidence of the particle filter (square root of the determinant of the covariance of all particle positions, lower is better). The blue curve (right y-axis) shows the optical signal strength measured in [dBm]. The GPS ground truth for these experiments is plotted in (c).

found within 9 to 15 min. In the run over 250 m the robot found the node within 17.5 min. This is a significant improvement over stochastic gradient descent when considering that the robot was coming from a distance 4-10 times larger than in the second set of experiments.

Figure 6-11 shows the results of the fourth set of experiments in which a combined acoustic and optical particle filter was used. In both experiments the robot successfully come close enough to the PANDA to establish and maintain optical communication. In the first run the robot cambs within communication distance in 16 min and in the second run this took 13 min. In the first run the robot was able to download a total payload of 200 MB.

6.10 Discussion

The proposed data muling system using bi-modal acousto-optical communication allows for large scale data recovery and eliminates the need for precise localization of the node and robot. It allows quick in-situ deployment of nodes and successive autonomous data recovery. In all gradient descent experiments the robot successfully found the underwater sensor node within a few minutes using acoustic gradient descent and proceeded to download a 1.2 MB file within 10 to 35 s using the optical link. Further we demonstrated that we can use the optical signal strength to maintain the robot's position close to the position of the sensor node. If the PANDA had been able to generate LT-Codes at the full rate of 4 Mbps then our throughput would have been 3.2 times higher than measured. It should be noted that this was purely a limitation on the computational side and not a limitation of the optical or acoustic modem itself. Also, since we do not use error coding and correction, all packets with a single bit error were discarded. This amounted to 13.6 MB of 37.5 MB and 13.5 MB of 69.1 MB in payload data lost. With the expense of more computational resources this bandwidth can be almost entirely utilized.

Finally, we experimentally evaluated the use of a particle filter to locate the node using only acoustic ranging. Both in simulation and experimentally the particle filter performed better than stochastic gradient descent. Finally we evaluated the use of a particle filter to locate the node and maintain position using a combination of acoustic ranging and optical signal strength measurements. The robot was again able to locate the sensor node and we showed the systems capability to find the node form distances of up to 250 m and download up to 200 MB of data in a single run.

Chapter 7

Real-time Remote Control of the Robotic Mobile Nodes

In this chapter we describe algorithms that allow real-time remote control of underwater vehicles. This is a crucial functionality in underwater observatories in cases where the user desires to intervene into ongoing autonomous operations or desires to perform dexterous task, such as fine manipulation or sample collection, which are too complicated to be executed automatically. The capabilities demonstrated in this chapter mark only the beginning of what is possible with as system such as AquaOptical II and AMOUR VI, but they lay the ground work in this important sector.

7.1 Concept

Marine scientists would like to use underwater robot systems to improve their understanding of the underwater world. However, current underwater robot systems are limited to open waters where few obstacles exist and there is little need for real-time feedback and control. One reason for this limitation is that cluttered and dynamic environments prevent the use of a tether between the underwater robot and the user. Reefs, rocks, and other items will quickly tangle the tether, hindering and possibly damaging the vehicle or the environment. Our goal is to develop a system that allows real-time, high-bandwidth communication with an underwater robot system to

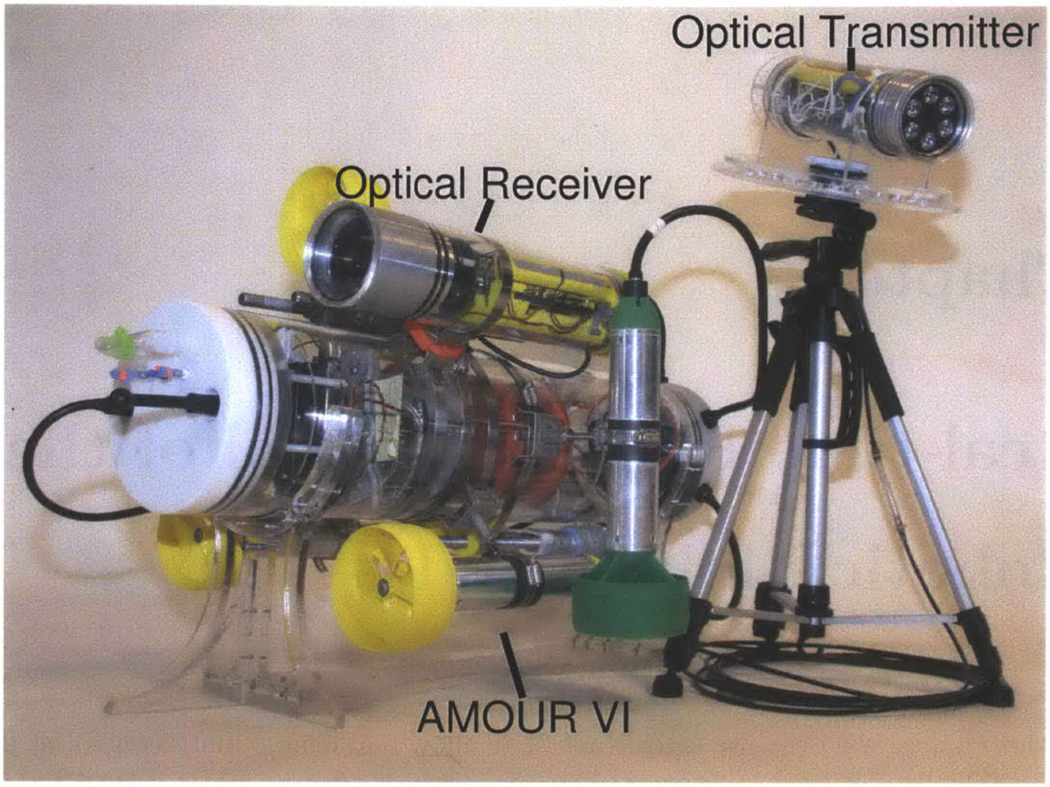


Figure 7-1: The optical transmitter mounted to a tripod (right) together with the optical receiver mounted to AMOUR.

enable operation in cluttered and dynamic waters such as in coral reef environments.

On land, radio communication usually allows systems to operate wirelessly. Unfortunately, radio does not work underwater because water absorbs most electromagnetic radiation. Acoustic modems are the most commonly used underwater communication system with ranges of several km. Acoustic communication, however, is extremely slow (only hundreds of bits per second) with high latency due to reflections and the relatively slow speed of sound underwater. Thus, it is not possible to dynamically control underwater vehicles remotely using acoustic communication in real-time.

Instead of using a tether or an acoustic modem, we developed a wireless underwater optical modem. In this chapter, we present the design and experimental results of a system to control our underwater robot (Autonomous Modular Optical Underwater Robot or AMOUR) in real-time using our optical modem link. Our optical modem achieves high bandwidth (megabits per second) and low latency while maintaining good coverage of the area of operation of the robot. Figure 7-1 shows a picture of

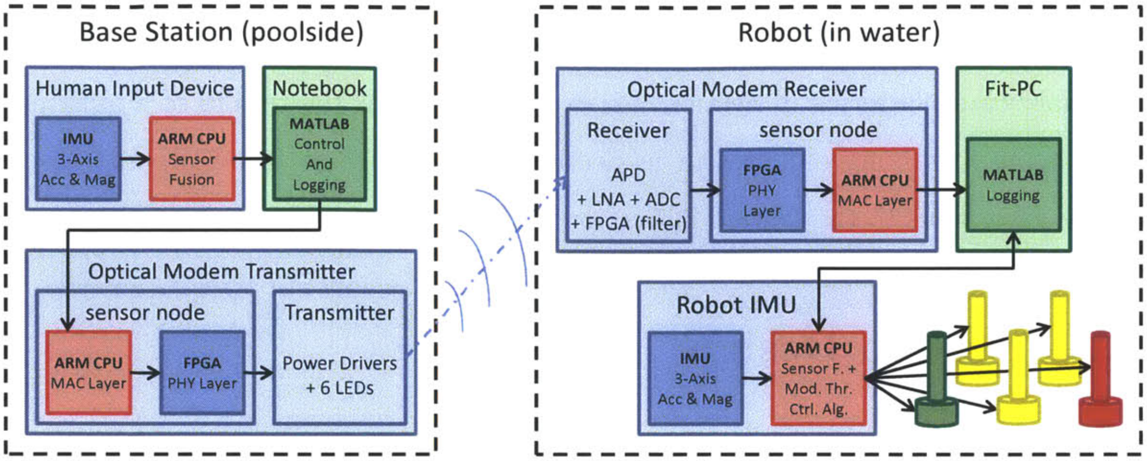


Figure 7-2: System overview showing the data path through all modules. The optional HID forwards data to a notebook computer. The computer runs a user interface for desired robot attitude, depth, and speed control and forwards this data to the optical modem transmitter. The transmitter encodes the signal using DPIM and transmits it optically. The optical modem receiver decodes the received pulse train into the desired robot state and forwards this information to the Fit-PC located inside the robot. The Fit-PC forwards this information to the robot's IMU. It also logs both the data from the optical modem as well as the robot's current position which it receives from the IMU. Finally, the IMU uses the Modular Thruster Control Algorithm [45] to compute thruster updates.

AMOUR and our optical communications system.

Our optical modem allows a land-based user to remotely operate the robot using our human input device (HID) in real-time. The system achieves real-time control due to the high speed and low latency optical link. We analyze the performance of our system in a pool. In nearly all positions and orientations over a 100 square meter area, our robot successfully receives optical commands from a single stationary transmitter.

We envision using this optical link to quickly update the mission goals and parameters of multiple autonomous robots. The optical link will also enable reception of high fidelity images and videos from the robots. Already, the optical modem has sufficient bandwidth available to allow the real-time operation of tens of robots in parallel with spare bandwidth for relay of images or video. The current system is uni-directional; however, we can easily add a transmitter on the robot to enable a bi-directional link allowing, for instance, the transmission of live video from the robot.

The rest of this chapter is organized as follows. We then outline the system design and hardware platforms in Section 7.2. Next, we present the results of a number of experiments verifying our system in Section 7.3. Finally, we conclude and discuss future extensions to the system in Section 7.4.

7.2 System Design

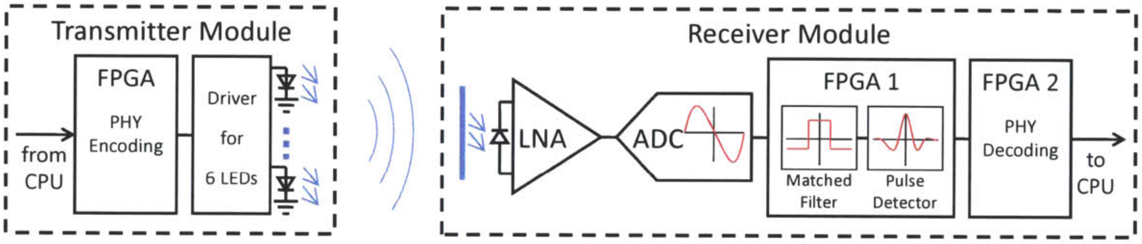


Figure 7-3: Optical Modem System Overview.

Our system allows control of an underwater robot via an optical link. The system consists of three high level components: (1) the base station, which provides an interface for the user to control the robot, (2) the optical modem, which forms the wireless communication link between the base station and the robot, and (3) our underwater robot, AMOUR, which is capable of motions in 6 degrees of freedom. The base station is further divided into a laptop computer running a specially designed user interface (UI) and a human input device (HID) which allows the user to directly control the robot's attitude. Figure 7-2 presents an overview of the system. In the following subsections, we present the core hardware modules and the software running on each module.

7.2.1 Hardware

The hardware consists of the AquaOptical I optical modem, our underwater robot AMOUR VI, and our human input device.

The AquaOptical modem is shown in Figure 3-7 and described in Section 3.2.1. Figure 7-3 outlines the simplified optical modem system used. The physical layer encoding scheme used on the optical modem is DPIM (Digital Pulse Interval Modulation).

The robot is described in Section 3.1 and the the human input device (HID) described in section 3.1.2. For the purpose of this experiment we configured AMOUR VI with 5 thrusters as shown in Figure 7-1. The three yellow thrusters provide thrust along the robot's main axis for yaw, pitch, and forward speed control. The red and the green thrusters provide thrust for roll and depth control. We mounted the optical modem receiver alongside the top yellow thruster. AMOUR's main body houses the battery, the battery management board, the IMU (Inertial Measurement Unit), and a small PC [31]. This entire system weighs approximately 19.5 kg and has a buoyancy of 3 kg. We manually weighted the robot to obtain neutral buoyancy.

7.2.2 Base Station and Optical Transmitter Software

The human input device is an optional input device. If in use, the HID forms the beginning of the data chain. The HID contains an IMU, which computes a rotation matrix describing the attitude of the HID device. The IMU updates the attitude at 400 Hz and forwards it over a serial link to a notebook computer at 80 Hz.

The notebook computer runs the user interface. It displays the HID's position and allows the user to set the robot's depth, speed, and maximum thruster limits. If the HID is disabled or not used, the user interface allows the setting of the robot's attitude and forms the first link in the data chain. The user interface runs at 80 Hz and forwards all information to the optical transmitter over a serial link at 80 Hz.

The optical transmitter runs a 400 Hz loop. On every iteration a new packet containing the newest desired robot position is queued into the transmitter queue. Every packet is 37 bytes long and is composed of a 4 byte header, 32 bytes of payload data, and a CRC byte. The payload data consists of 16 values of 2 bytes each, containing: (1) the rotation matrix describing the desired robot attitude (9 entries), (2) the robot's desired depth (1 entry), (3) the maximum allowed thruster speeds (1 entry), and (4) the desired translational robot speed (3 entries). The 15th value is reserved for future use and the 16th value contains an additional CRC check. For the

DPIM encoding, we chose a pulse length of 2 us with a minimum guard interval of 2 us and a difference between symbols of 2 us. This results in an average data rate of 200 Kbps. The maximum transmission time for a packet was 1.48 ms with a guard interval maintaining a transmission rate of 400 packets per second.

7.2.3 Optical Receiver and Robot Software

For the robot the optical receiver is the first link in the data chain. A loop running at 1000 Hz checks for received packets with correct CRC values. It then updates an internal desired robot state vector. This vector, along with the time passed since the last update, is forwarded over a serial link to the PC inside the robot at 25 Hz.

The PC checks that the desired robot state is not older than 1 sec. If it is older, the PC invalidates the state, otherwise the state is marked as valid. If the state is valid, the PC directly forwards the state to the IMU at 12.5 Hz. If the state is invalid, the PC modifies the state to set the maximum thruster speed to zero and forwards the modified state to the IMU. This is a safety feature to disable the robot in case the optical link fails. Once the PC receives a valid state (newer than 1 sec), it slowly ramps the maximum thruster speed to the value set in the desired robot state. The PC logs all data received from the optical receiver. It also logs the robot's attitude and thruster speeds which it receives from the robot's IMU at 80 Hz.

The IMU estimates the robot's position at 400 Hz and updates the thruster outputs at 80 Hz. The IMU uses the Modular Thruster Control Algorithm presented in [45] to orient and position the robot into the desired attitude.

7.3 Experiments

In this section, we describe a set of experiments we performed to characterize the operation of the optical modem by itself and in conjunction with the robot. First, we present ranging experiments we performed in air, in a pool, and in the Singapore Harbor. These ranging experiments were performed without use of the robot. Next,

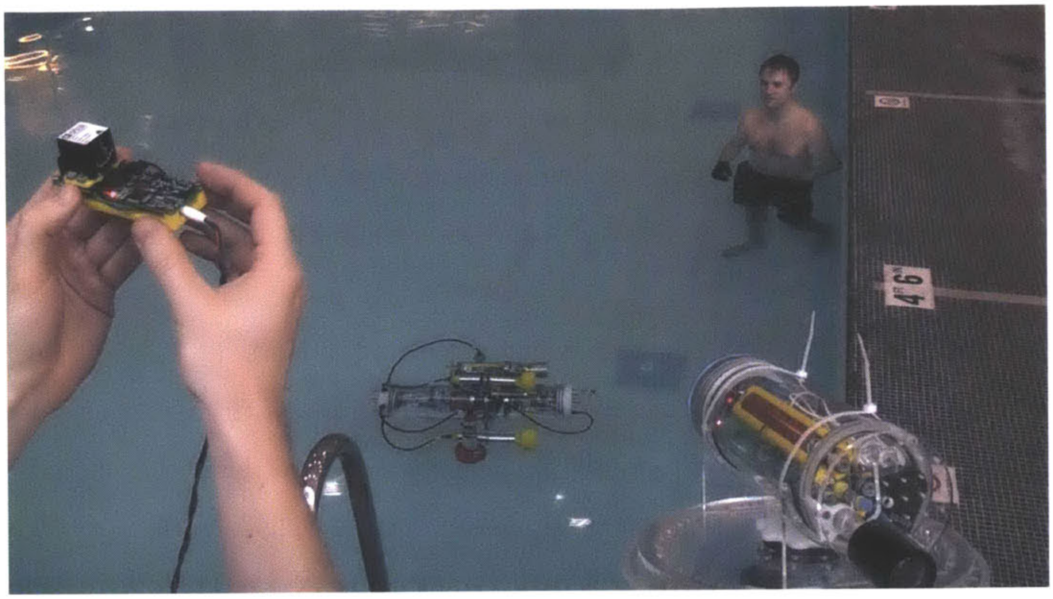


Figure 7-4: Picture of the robot in the pool. The HID is at left and the optical transmitter is at right.

we characterize the details of using the optical modem to control the robot in a pool. And finally, we demonstrate using the human input device over the the optical modem link to control the robot in the pool. Figure 7-4 shows a picture of the HID, robot, and optical transmitter from left to right.

These experiments show that we are able to obtain good optical communication ranges under a variety of conditions and are able to have full control of the robot in a pool while using only a single stationary transmitter. Additionally, our optical modem has a sufficiently high bit rate and low latency to allow easy and intuitive wireless control of the robot using our HID device.

7.3.1 Optical Modem Range in Air

We performed optical modem experiments under three different conditions to characterize the range of the optical modem. For other experimental characterizations of the optical modem see [44].

Our first experiment occurred at night along a lit street with a length of 200 m. We performed it in air, utilizing a bitrate of 1.2 Mbps. For distances up to 150 m the optical modem received 100 % of the packets. At the full distance of 200 m, the optical modem received 70 % of the packets.

Our second experiment took place in a swimming pool with normal lighting using a speed setting of 1.2 Mbps. The transmitter and receiver were both submerged at a depth of 1m. At 30 m, the full length of the pool, the optical modem received 100 % of the packets.

We performed the final experiment in the Singapore Harbor. The optical transmitter and receiver were attached along a rigid 10m pole at varying separations and then lowered into the water to a depth of approximately 2 m. This location had water with extremely low visibility, estimated to be 1.5 m. Using a bitrate of 600 Kbps, we received 100 % of the packets when the transmitter and receiver were separated by 7 m. After this range, the success rate dropped off quickly with only 40 % success at 8 m and virtually no correct packets at 9m separation. While the Singapore Harbor experiment achieved a shorter maximum range than the pool experiment, the maximum range obtained was over five times the maximum visibility distance.

7.3.2 Optical Modem Range in Pool

In this experiment, we attached the optical modem receiver to the robot and manually moved the robot at various angles throughout the pool to characterize the system at various locations in the pool. We fixed the optical transmitter location and orientation approximately in line with the right edge of the pool and 1 meter below the bottom edge of the pool as seen in Figure 7-6. The transmitter was elevated 2m above the water surface and was pointed at the center of the pool. We chose to position the transmitter outside of the water to achieve a good illumination of the entire pool area while simultaneously avoiding saturation of the receiver at close range. The pool is 13.7 m long and 7.3 m wide. The depth along the right side is 1.4 m and along the left side is 1.1 m.

Figure 7-6 shows the results of the experiments. We moved the robot along each edge of the pool as well as along the diagonals with the optical modem aimed in five different orientations, pointing: towards the transmitter, to the left as viewed from

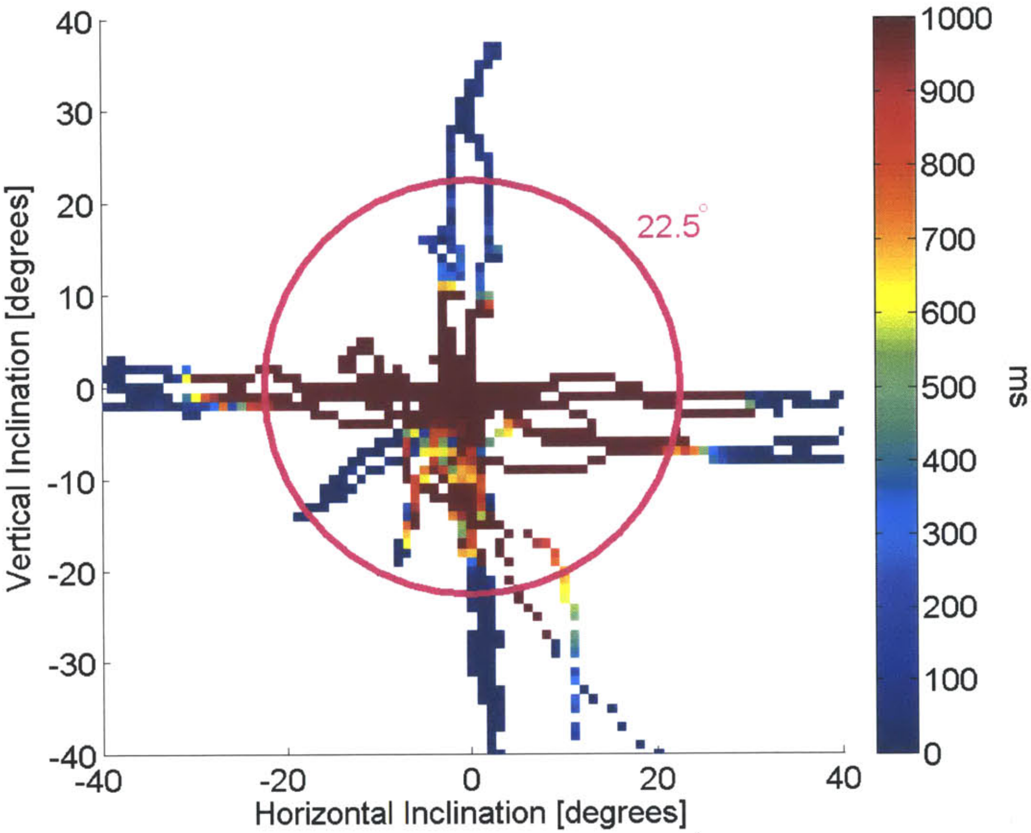


Figure 7-5: Reception in the center of the pool with the receiver pointed towards the ceiling. The x-axis corresponds to inclination of the receiver along the short side of the pool. Negative values indicate the receiver pointing towards the left side of the pool and positive values towards the right side. The y-axis corresponds to inclination of the receiver along the long side of the pool. Negative values indicate the receiver pointing towards the transmitter and positive values away from the transmitter (which is located towards the lower right). The origin corresponds to the receiver pointing straight up at the ceiling.

the transmitter, away from the transmitter, to the right, and down towards the pool floor. The optical transmitter sent packets to the robot at a rate of 400 Hz or a packet every 2.5 ms. Every 25 ms a monitor loop on the optical receiver side recorded the time since the last valid packet.

Figure 7-6 plots the received packet delays for each receiver orientation. For each orientation, we moved the robot along the boundary of the pool and across the diagonals. We discretized the pool and averaged delays within each bin. As packets are expected every 2.5 ms, any packet delay of less than 2.5 ms (blue-yellow) indicates near perfect receipt. Red bins indicate that the average delay was 5ms, corresponding to a success rate of about 50 %. For all of these experiments the receiver was able to receive a significant fraction of the packets sent. This is a somewhat surprising result given that the receiver pointed away from the transmitter for most of the experiment. This can be explained by the sensitivity of the receiver and the white, reflective surface of the pool.

Interestingly, when the receiver was aimed to the right, perpendicular to the transmitter, the receiver obtained the best average receipt rate. The upper right corner and lower edge of the pool tended to be areas of weak reception in most configurations. In these positions, the pool edge shadows the receiver.

Perhaps the most unexpected result is that when the optical receiver was aimed towards the transmitter (far left figure), we saw the worst reception in the upper left quadrant. We believe this is due to an over-saturation of the receiver given that it was aimed nearly directly at the transmitter, which also had a large, sunny window behind it.

The only orientation where the optical modem performed poorly was when it was oriented directly up towards the ceiling. In this configuration, the optical modem rarely received data. We believe that this was caused by the lack of reflective surfaces and overpowering of the receiver by the overhead lights in the pool. Figure 7-5 shows an analysis of the angle at which the optical modem could receive data while pointed up in the center of the pool. Red indicates poor or no reception and blue indicates good reception. The angle of the robot was manually adjusted to explore the range at which the optical system could receive data. The circle indicates an angle of 22.5 degrees. Around this angle the optical modem started receiving data.

7.3.3 Remote Robot Operation Using the Optical Modem

In this experiment, we used the HID to control the desired orientation of the robot. We transmitted the desired orientation to the robot over the optical communication link. Users of the system indicated that the system was very responsive and could not differentiate between a tethered configuration and the configuration using the optical

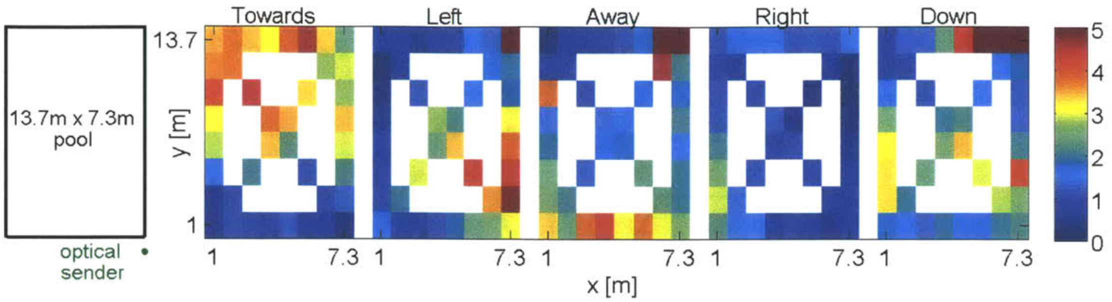
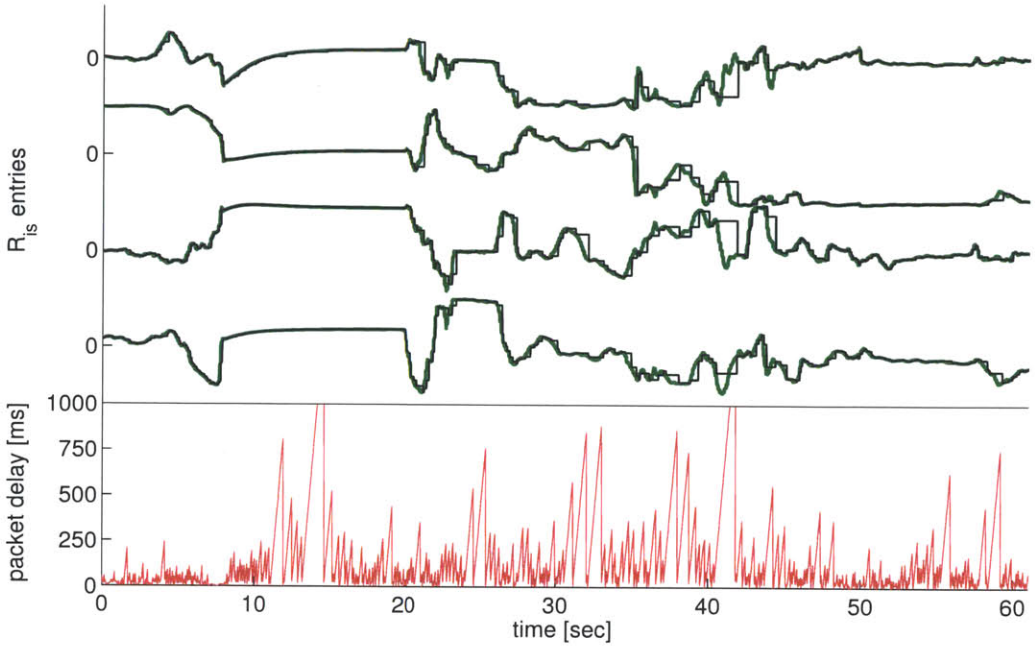


Figure 7-6: Average time since last valid packet received during IMU update.

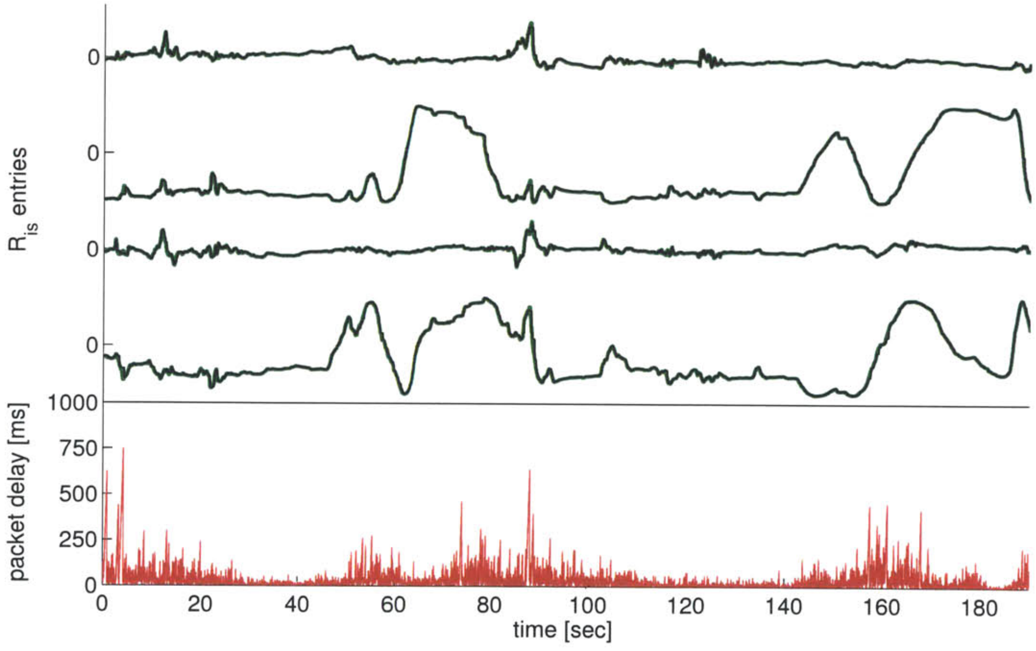
link.

Figure 7-7 shows the results of the HID performance during two segments of the experiment. In the left figure, the robot was in an area with poor reception seeing an average latency of 104.3 ms. In the right figure, it was in an area with good reception seeing an average latency of 43.4 ms. Over the course of the whole experiment the average latency was 51.4 ms. The top four lines in the figures are values from the 3x3 rotation matrix. These uniquely define the rotation matrix of the robot up to some sign ambiguities. The desired orientation as output from the HID on the transmitter side is in green and the command received via the optical link by the robot is in black. At the bottom of the plot the red line indicates the time since the last valid packet was received.

For most of the time, the transmitter and receiver had the same target orientations. However, in the middle of the left plot, there are periods where the optical link experienced high latency. During these periods, the received data deviates from the transmitted data for short periods of time, causing a step function in the data (such as seen around 40 sec). When this delay exceeded one second, which occurs only twice in the left plot, the thrusters were automatically disabled as a safety precaution. This prevents the robot from operating with old target orientation and speed information. Once a new transmission was received, operation resumed.



(a) Poor reception.



(b) Good reception.

Figure 7-7: Performance of the optical system while using HID. The top lines in the figures indicate the values transmitted (green) and received (black). The lower red lines plot the reception delay. In (a) the robot was in an area with poor reception and, in (b), an area with good reception.

7.4 Discussion

We showed that optical communication for underwater robot control allows data rates on the order of megabits per second and latency on the order of a millisecond. In comparison acoustic communication achieves data rates on the order of kilobits per second and latencies of multiple hundred milliseconds. The disadvantage of underwater optical communication is the reduced communication range of tens of meters when compared to ranges of multiple kilometers achieved with acoustic communication. A further disadvantage of underwater optical communication is that ambient light can saturate the receiver. Ambient light is a predominant problem at low depths, as in the case of the system presented in this chapter, which is designed to operate at depths of up to 100 m. We showed that our system can deal with ambient light in cases where the receiver is not directly pointed at a light source, e.g. the sun or overhead lamps in the pool. A further challenge of shallow waters is the increased turbidity due to surface currents and waves.

Chapter 8

Robust Real-time High Definition Underwater Video Streaming

In this chapter we present an application of AquaOptical II in the form of algorithms used to transmit live video streams using AquaOptical II. These algorithms can be used in underwater observatories to transmit video feeds from the stationary and mobile sensors back to the user if a series of optical links is available. As with remote control of vehicles, such functionality is desirable for any for of remote user intervention into an underwater observatory. It can, however, also be used simply for passive monitoring of fully autonomous systems.

8.1 Concept

In this chapter, we introduce a state-of-the-art underwater video streaming solution that is comprised of AquaOptical II, and a two-layer digital encoding scheme designed for error-resistant communication of high definition images. Specifically, we employ (255,239) Reed-Solomon (RS) [93] inner code for byte-level error correction, coupled with a systematic Luby Transform (LT) [84] outer code to mitigate packet-level losses. Our video transmission solution does not require feedback and can thus be used in a unidirectional communication channel with low infrastructural overhead. Our

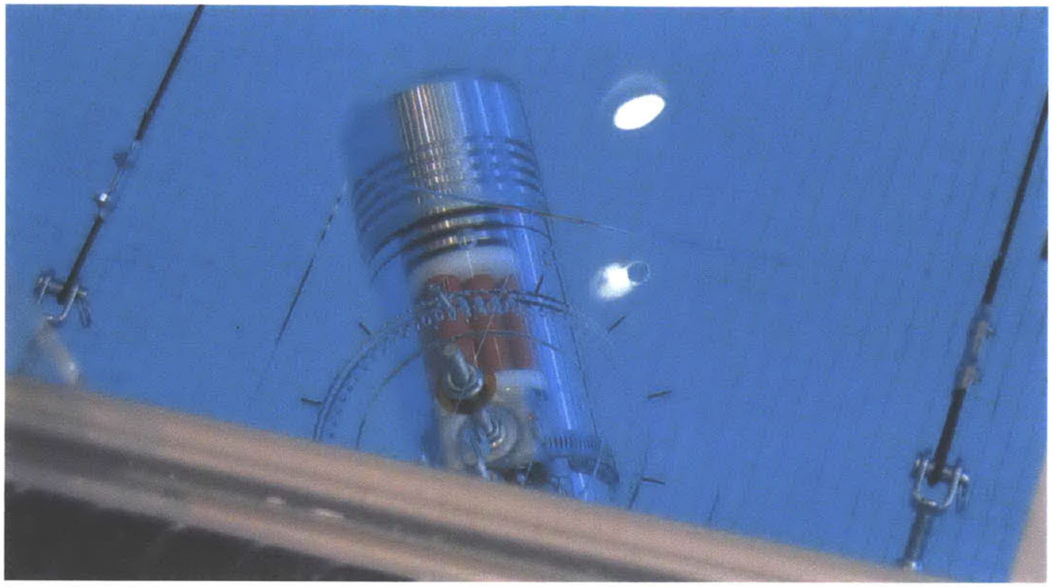


Figure 8-1: AquaOptical II modem mounted on an underwater test rig.

empirical assessments in clear water settings demonstrate the ability to stream video resolutions of up to 1288 × 964 pixels and frame rates of up to 15 frames per second (FPS) robustly at ranges of up to 40 m. We further show that this system can deliver 752 × 480 resolution video at 15 FPS with an average transmission latency of 100 ms and end-to-end latency of 170 ms, where the latter measure takes into account frame grabbing and image display delays.

This chapter contributes the following:

- a novel high-bandwidth underwater optical communication device;
- a two-layer digital encoding scheme which is highly robust to channel noise and also accommodates for unidirectional real-time streaming;
- extensive empirical results of the an end-to-end underwater video streaming solution.

Our video streaming solution is a key enabling technology for a wide range of underwater settings, including surveillance and robotics. We are especially interested in real time video transmission to enable such applications as video feed exchange

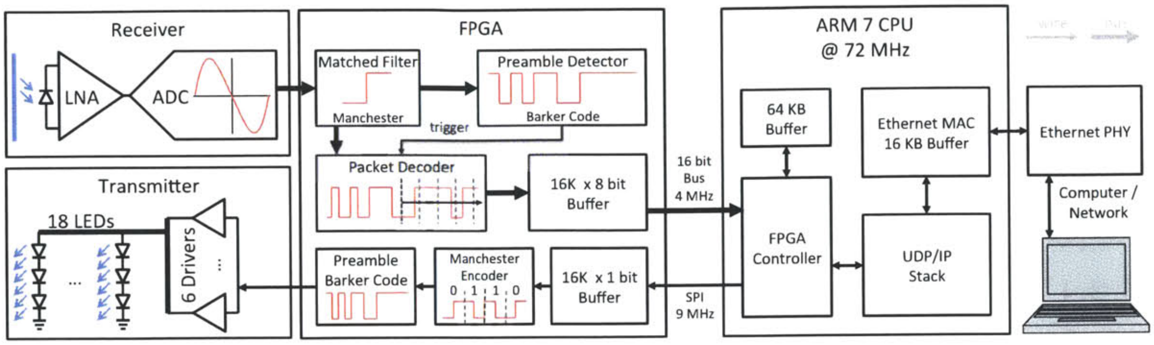


Figure 8-2: The AquaOptical II modem consists of a transducer compromising the `Receiver` and `Transmitter` units, an FPGA that implements the physical and link layer, and an ARM 7 CPU that implements the MAC layer and Ethernet communication.

among divers and live video delivery for surveillance networks. In addition, our high-bandwidth low-latency streaming solution plays a critical role in underwater robotics, including dexterous operations using Remotely Operated Vehicles (ROV), real-time tele-control in narrow caves and shipwreck settings, and off-board visual autonomous navigation setups. The presented video streaming solution further complements our work in chapter 7, which demonstrated wireless remote control of an underwater vehicle using free-space optical communication.

The remainder of this chapter is organized as follows: in Section 8.2 we present an overview of the AquaOptical II modem and the digital encoding algorithms. Section 8.3 describes our experimental methodology in evaluating this underwater video streaming solution. Experimental results are given in Sections 8.4 and 8.5. We present a brief analysis of the results in Section 8.6.

8.2 System Overview

This section provides an overview of our underwater video streaming solution between two AquaOptical II modems. The video feed is encoded on the transmitting computer in real time from images acquired by a digital camera. Once the data has been transmitted through the water medium via AquaOptical II, it is subsequently decoded on the receiving computer and is either analyzed by vision algorithms, or displayed to the user.

The AquaOptical II hardware and software was described in section 3.2.1. This section describes the video compression and two-layer encoding processes.

8.2.1 Video Frame Compression

In order to transmit high resolution video frames at fast frame rates, it is necessary to compress the video data before transmitting through the AquaOptical II modem. Inter-frame prediction-based video codecs, such as H.264 and Theora, can achieve very high compression ratios through the use of *predicted* image frames (P-frames) and *bi-predicted* frames (B-frames), which efficiently encode pixel content as relative changes to either previous frames, or previous and subsequent frames, respectively. When transmitting the compressed stream of frames over a noisy channel however, the loss of a single encoded frame will likely introduce significant visual artifacts in nearby frames as well. In addition, predictive frames often incur a significant latency during decoding, on the order of $\frac{1}{FPS}$, due to their inter-frame dependencies. Furthermore, these modern video codecs require a significant amount of computational power, and thus cannot sustain real-time software compression of high definition content. Specifically, our empirical testing using the `GStreamer` multimedia framework revealed that both H.264 and Theora codecs had difficulty encoding and decoding 1288×964 resolution images at 8 FPS on a high-end quad-core 2.5 GHz processor.

To achieve the goal of delivering real-time video content with minimal latency, we opted to use the comparably straight-forward Motion-JPEG (M-JPEG) codec, which operates by compressing individual frames separately using JPEG. Although this codec results in worse compression ratios compared to modern video codecs, its encoder and decoder both require minimal processing time, on the order of 5 ms in the worst case for 1288×964 pixel images. Therefore, using M-JPEG we were able to stream high quality video at fast rates of up to 15 Hz within the 4 Mbps bandwidth limit of the AquaOptical II modems. More importantly, the M-JPEG codec allows each received frame to be decoded immediately, and therefore avoids the inherent

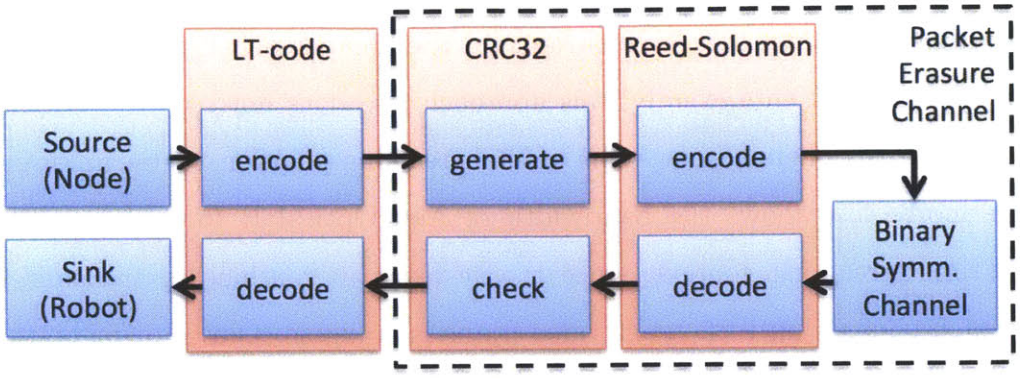


Figure 8-3: Two-layer error coding scheme employed for high quality real-time video transmission via unidirectional free-space optical communication.

inter-frame latencies found in modern video codecs.

8.2.2 Video Frame Transmission

Fig. 8-3 illustrates a two-layer digital encoding scheme that we employed on AquaOptical II to robustly transmit high-volume framed content in a potentially noise-laden communication channel, with minimal transmission latency.

AquaOptical II is driven by an oscillator clock source with a frequency stability of 50 parts per million (ppm). This consequently limits the maximum packet size in burst transmission to be a few hundred bytes, in order to avoid de-synchronization errors due to clock drifts. To accommodate this, each video frame is converted into a sequence of fixed-sized fragments by AquaOptical II prior to transmission. This has the benefit of restricting the scope of transmission errors, in that if a single fragment is lost or corrupted, only that fragment needs to be recovered, rather than having to recover the entire frame.

We employed Luby Transform (LT) codes, a form of digital fountain codes, to ensure robust transmission of each video frame. LT codes operate by splitting a large payload into N equal-sized blocks, and then generating a theoretically infinite stream of packets, which we call fragments, by convolving one or more of these blocks together. It can be shown that the reception of any subset of size $\geq (1+\epsilon)\cdot N$ of these fragments allows for the recovery of the original frame with high probability.

In addition to placing only a constraint on the number of fragments received, but not on which fragments exactly, this approach also does not require feedback from the receiver, and hence allows our video streaming solution to be unidirectional. Furthermore, this allows the bandwidth of AquaOptical II to be fully utilized in a real-time streaming context, since the system can continue to transmit further different fragments until the next video frame is available. Our implementation uses a variant called systematic LT codes [84], where the first N fragments transmitted are the single blocks from the source frame, and subsequent fragments are generated pseudo-randomly based on the fragment ID.

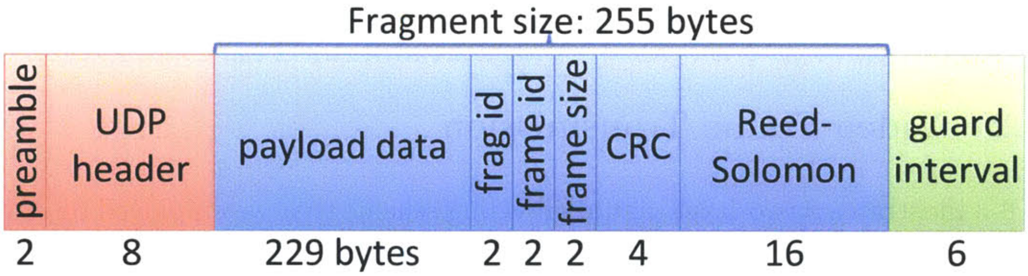


Figure 8-4: Structure of packets sent by the AquaOptical II modem.

The 255-bytes packet structure for AquaOptical II is shown in Fig. 8-4, and contains both the payload fragment data, as well as the fragment ID, the frame ID, and the total size of the frame. In addition, we generate a 4 byte cyclic redundancy check (CRC) from the fragment payload, and then encode the entire packet using (255,239) Reed-Solomon (RS) error correcting code, which can recover up to 8 bytes of error per packet. By accounting for the preamble, UDP header, packet data, and the guard interval, each packet takes a total of 547 μs to transmit using AquaOptical II. Our video streaming solution configured the LT encoder to send a new packet to the modem every 560 μs, thus resulting in 1785 fragments being transmitted per second.

AquaOptical II's coding scheme, as depicted in Fig. 8-3, implements a binary symmetrical channel, where noise-induced bit-flips occur with a probability p. This probability depends on the receiver's signal strength, which in turn is affected by factors such as transmission distance and water turbidity. By using a Reed-Solomon code in combination with a CRC we are turning this binary symmetrical channel into a packet erasure channel, in which entire packets are either lost (the errors cannot be recovered and the CRC fails) or successfully decoded with very high probability (where the CRC passes). The use of an LT code makes it highly likely that a frame can be recovered as soon as enough fragments have been received, thus keeping latency low.

8.3 Experiments

We evaluated our underwater transmission system by measuring the performance of its live video transmission in different settings. We deployed AquaOptical II modems in two indoor pools of 25 m and 50 m lengths, and streamed live video feeds at 0.5 m depth using a cabled test rig, as shown in Fig. 8-5. In these setups, the video source was obtained from natural pool scenes and real-time footage of the AMOUR robot, captured using a waterproof USB camera connected to a laptop computer. The encoded video fragments were sent out as UDP packets from a stationary AquaOptical II modem positioned against the pool wall. A second AquaOptical II modem, mounted on a movable cart positioned along the 1-D track, received these packets and forwarded them to the recipient computer for decoding, analysis, and logging.

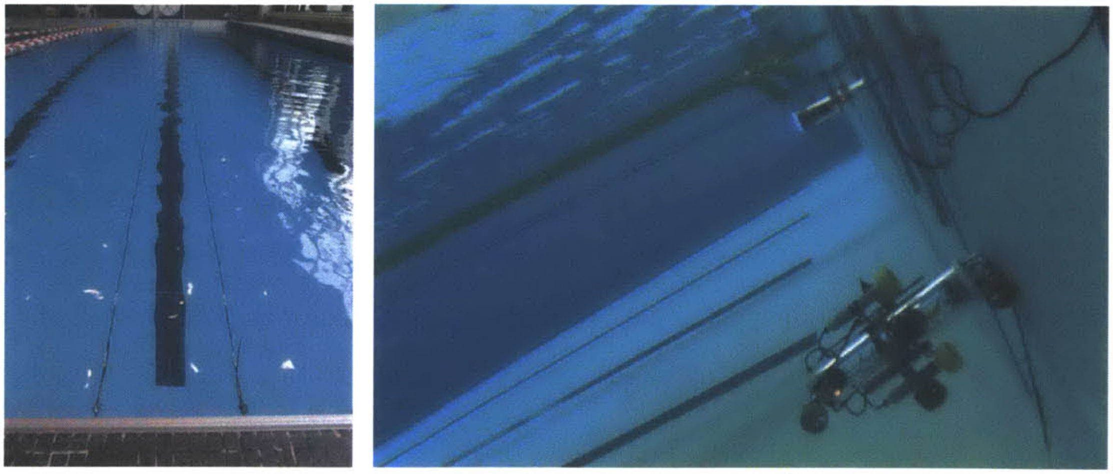
Our pool trials consisted of over 70 video streaming runs, corresponding to over 30,000 frames transmitted in total, while varying different system parameters. These parameters included the image resolution, video frame rate, JPEG compression quality, and modem-to-modem distance. We computed the following metrics on our collected data to thoroughly characterize the performance of the resulting video transmissions:

- *frame success rate*: percentage of successfully decoded frames;
- *frame latency*: time delay for transmitting a raw RGB image from one computer to another through the encoded optical channel;

- *peak signal-to-noise ratio (PSNR) [123]*: non-subjective measure of image quality between the compressed transmitted feed and the raw source image;

Furthermore, the collected video transmission data was also used to evaluate the performance of our digital encoding scheme for transmitting general purpose payloads over a potentially noisy free-space optical channel.

In addition to characterizing the transmission performance of individual video frames, we are also interested in studying the overall performance of live video feeds using the proposed underwater streaming solution. This is motivated by a number of applications, such as remote visual monitoring of underwater sensor stations, and visual-guided tele-operation of underwater robotic vehicles. To this end, we repeated our video streaming experiments in a controlled benchtop environment, where two AquaOptical II modems were positioned next to each other in free space. This is comparable to the underwater setup since we had empirically confirmed that the AquaOptical II hardware exhibited near identical latencies and packet drop rates



(a) Cabled test rig

(b) Acquired camera frame

Figure 8-5: Our experiment testbed consists of two AquaOptical II modems attached to carts and mounted facing each other on a cabled test rig in an indoor pool (a). Live video feed of the underwater AMOUR robot were streamed through the setup during our tests (b).

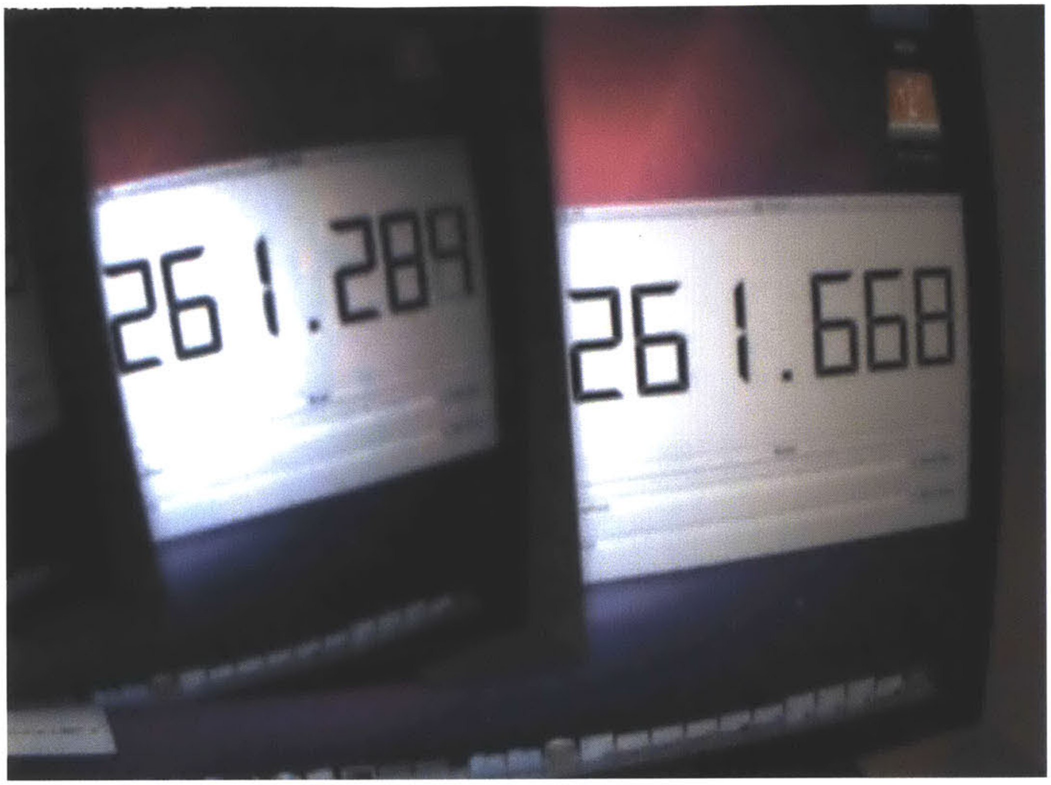
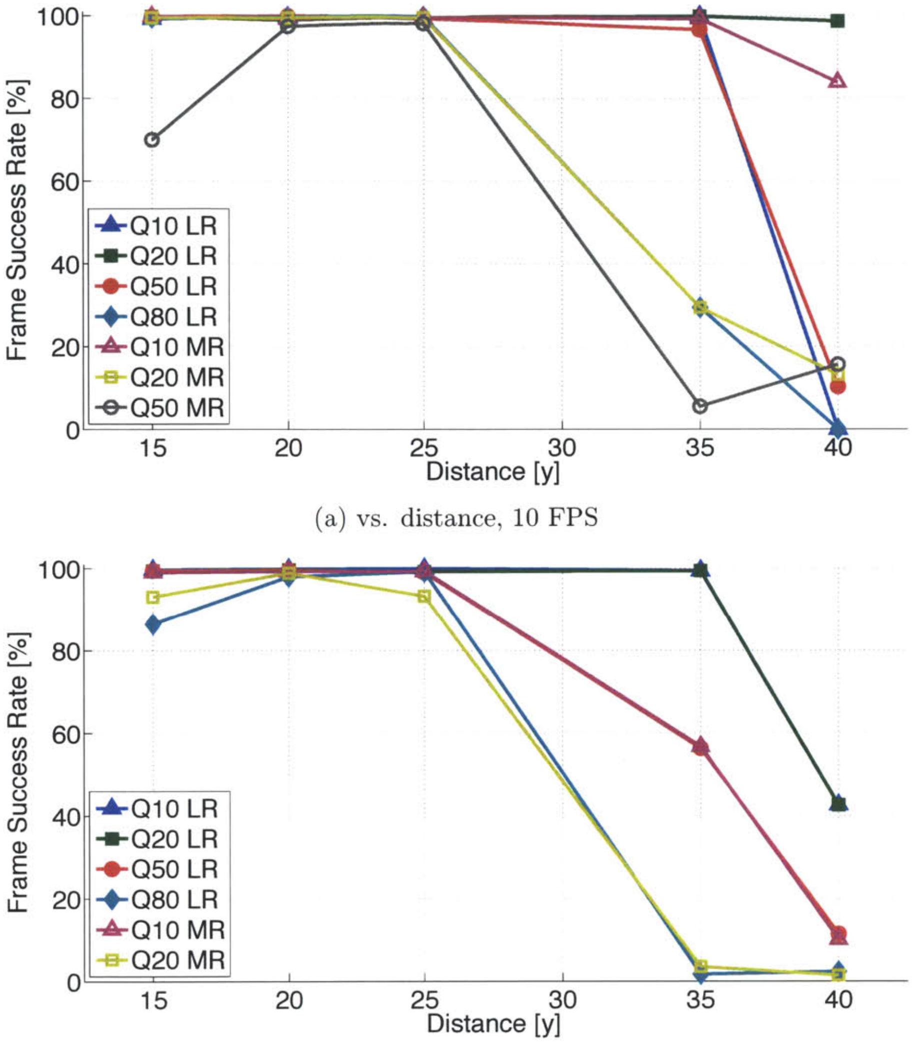


Figure 8-6: Sample acquired camera frame during our benchtop trial, used to determine the end-to-end video capture-transmission-display latency.

both underwater and above water. This close-range analogue enabled us to aim the camera at the recipient computer's monitor, which displayed both the latest decoded frame as well as a high-resolution stopwatch operating at the computer display refresh rate of 60 Hz, as shown in Fig. 8-6. Thus, the recorded camera frames revealed the *end-to-end video latency*, namely the time delay between capturing a camera frame and displaying on the recipient's monitor, which is one of the essential statistics of video streaming applications.

In our pool experiments we used the FireFly MV CMOS USB 2.0 camera (FMVU-03MTC), and in benchtop trials we also used FireFly, as well as the Flea 3 CCD Gigabit Ethernet camera (FL3-GE-13S2C), both commercially available from Point-Grey technologies[1]. Both cameras were equipped with global shutters, and produced raw Bayer-encoded images, which were then converted into RGB colorspace prior to applying the M-JPEG video codec. The FireFly MV camera was configured to grab

[1] www.ptgrey.com



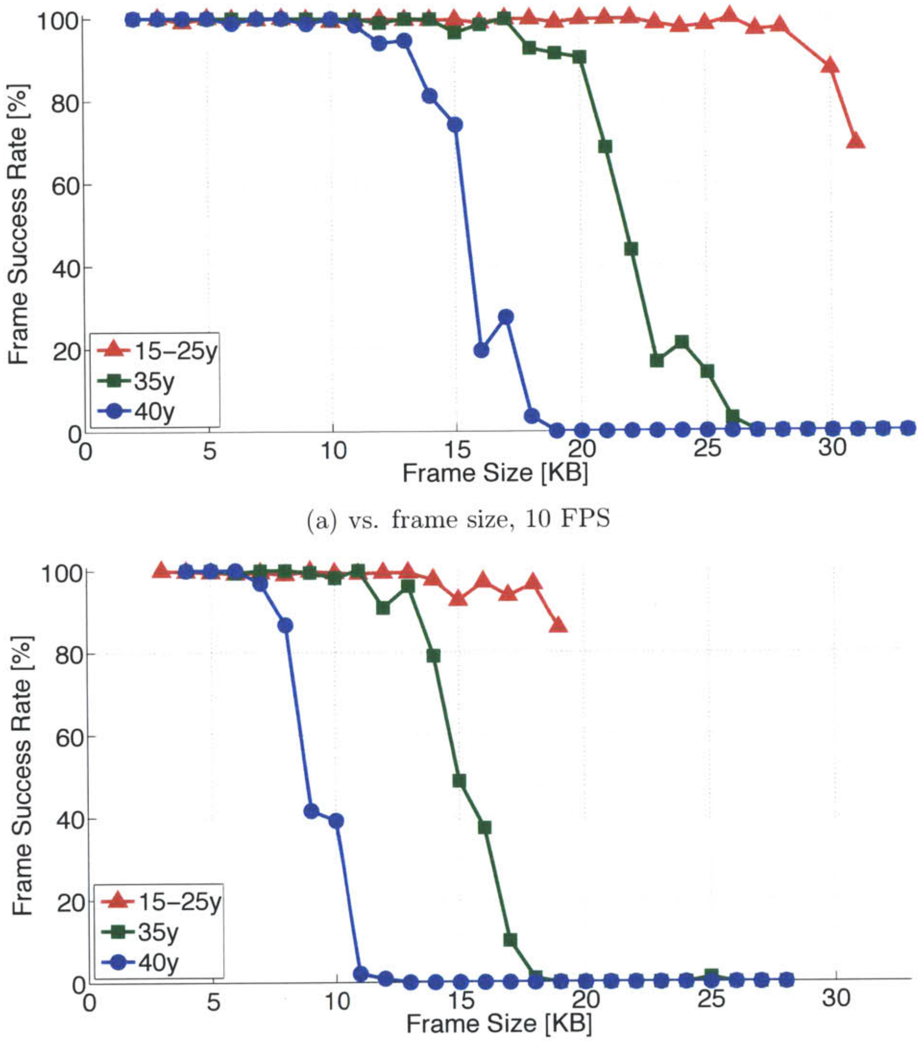
(a) vs. distance, 10 FPS

(b) vs. distance, 15 FPS

Figure 8-7: Frame reception success rates for different video qualities, plotted against modem-to-modem distance.

frames of 752 × 480 pixel resolution at 60 Hz capture rate, and the Flea 3 camera was configured to grab 1288 × 964 pixel images at 30 Hz. Using these sources, we carried out video streaming sessions at three different image resolutions:

- Low resolution (LR): 376 × 240 pixels
- Medium resolution (MR): 752 × 480 pixels
- High resolution (HR): 1288 × 964 pixels



(a) vs. frame size, 10 FPS

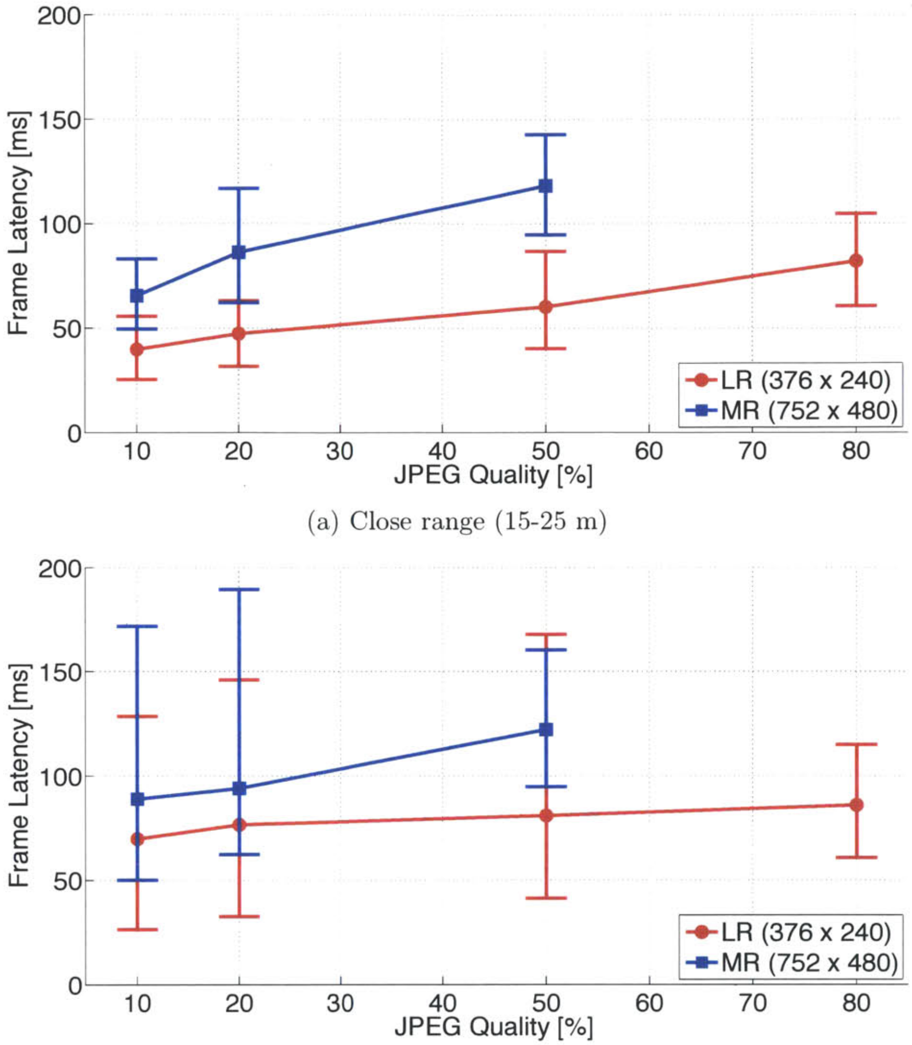
(b) vs. frame size, 15 FPS

Figure 8-8: Frame reception success rates at different modem-to-modem distances, plotted against frame size (in 1 KB bins).

In a final set of experiments the video transmission system was evaluated at three different sites in the field:

1. The Charles River in Cambridge, Massachusetts.
2. Boston Harbor waters near the Long Wharf.
3. Coastal waters near Gallops Island.

All three experiments were conducted by manually holding the modems underwater and aiming them at each other. Distances were measured using a taught rope

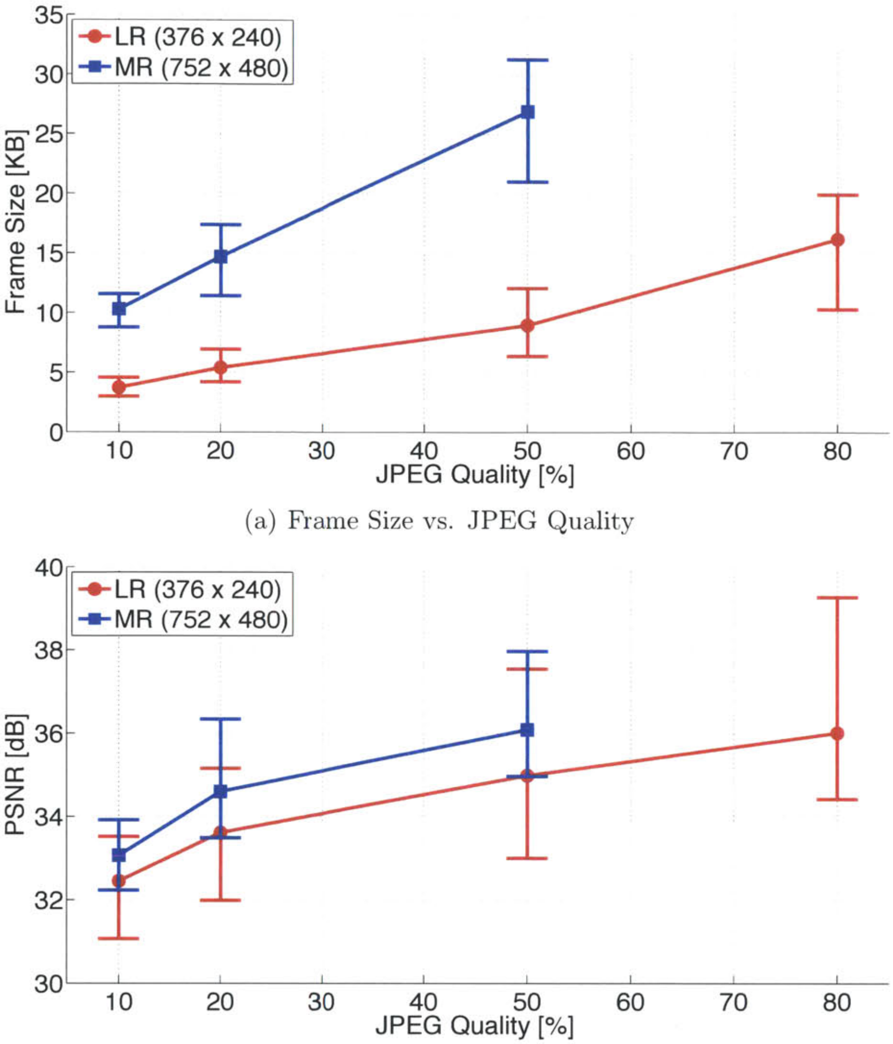


(a) Close range (15-25 m)

(b) Far range (35-40 m)

Figure 8-9: Frame latency at different ranges, as a function of JPEG quality. Vertical bars depict 5% and 95% percentiles.

with 1 m markings. In all three instances waves caused significant movement of the boats from which the experiments were conducted and thus made it hard to keep the modems level. In particular during the experiments at Gallops Island we experienced significant movement due to strong waves. This caused additional communication noise as the alignment of the modems would be significantly off at times.



(a) Frame Size vs. JPEG Quality

(b) PSNR vs. JPEG Quality

Figure 8-10: Image properties for the video streams used in the pool experiment, as a function of JPEG quality. Vertical bars depict 5% and 95% percentiles.

8.4 Experimental Results

The success rate of receiving JPEG frames in our video streaming pool trials are plotted as a function of modem distance in Fig. 8-7, and as a function of frame size in Fig. 8-8. For the received JPEG frames, Fig. 8-9 depicts the frame transmission latencies as a function of the JPEG compression quality, for both close ranges of within 25 m and far ranges of up to 40 m. This compression quality ratio is proportionally correlated to the size of each encoded frame, and separately to the PSNR image

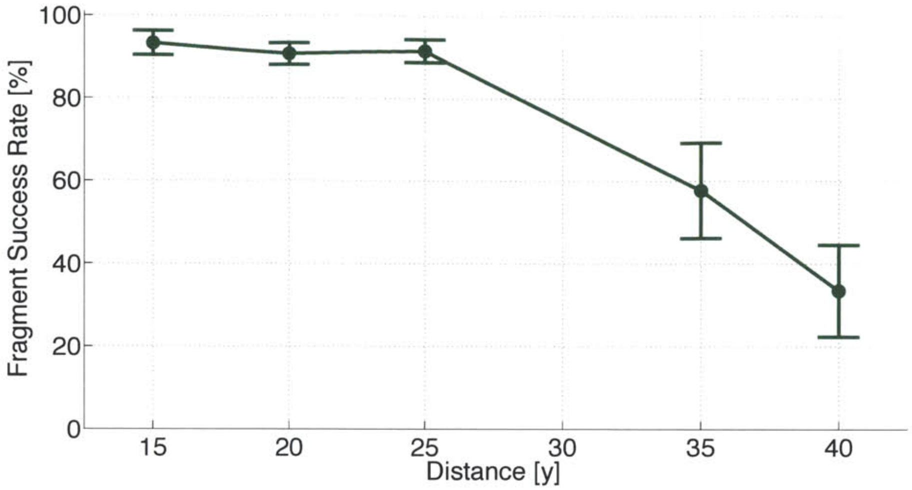
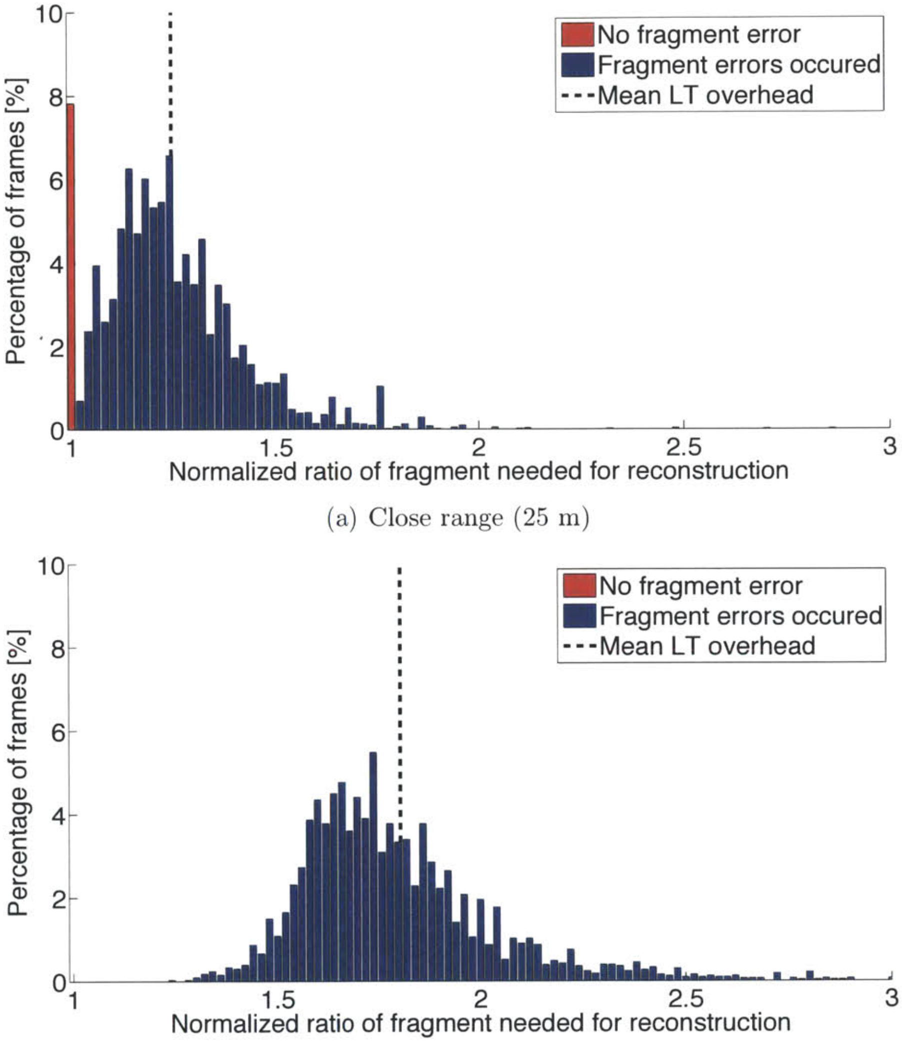


Figure 8-11: Fragment reception success rates for all pool trials, vs. distance. Vertical bars depict $\pm 1\sigma$.

qualities, as shown in Fig. 8-10.

Focusing on the performance of the two-layer RS / LT encoding scheme, Fig. 8-11 depicts the percentage of fragments successfully decoded as a function of modem distance. Fig. 8-12 illustrates the overhead required by the systematic LT outer code to reconstruct the frame. This overhead is measured as the ratio between the number of total fragments received before the frame could be decoded, normalized over the number of systematic fragments, N, for the source image frame. These overhead results are aggregated into a plot of the expected LT overhead as a function of the distance, in Fig. 8-13.

Looking at the packet level, Fig. 8-14 shows histograms of the time intervals between received fragment packets, both for near and far ranges. The *expected* bandwidth of our underwater optical transmission system at different ranges can be computed as the ratio of the averaged inter-packet interval, divided by the total volume of packet data transmitted during this duration. This expected bandwidth can also be determined by multiplying the fragment reception success rate, in Fig. 8-11, with AquaOptical II's maximum transmission bandwidth of 4 Mbps.



(a) Close range (25 m)

(b) Far range (35 m)

Figure 8-12: Ratio of the number of fragments received at a fixed range before a frame can be decoded, over the number of systematic fragments for the frame, representing LT overhead. The red bar on the left side of the plot represents instances where the frame was immediately decoded after successfully receiving the initial N systematic fragments.

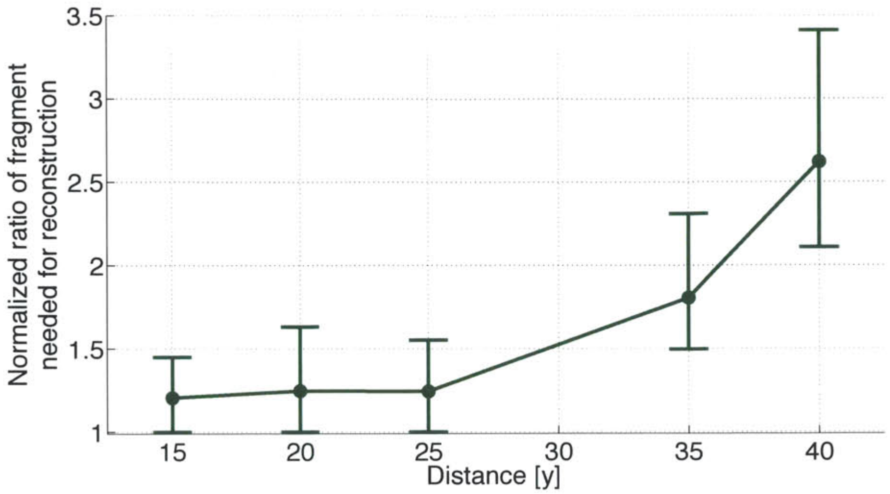
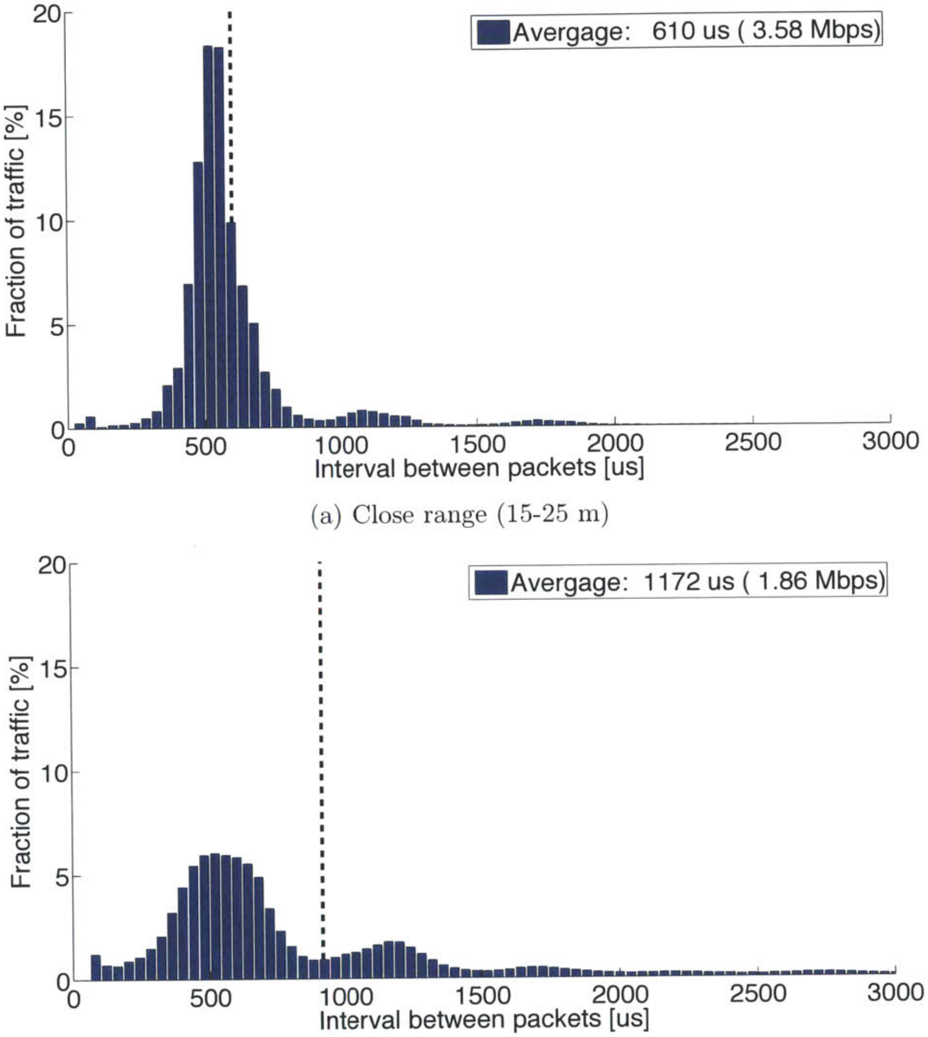


Figure 8-13: LT overhead as a function of modem ranges. Vertical bars depict 5% and 95% percentiles.

The end-to-end latencies measured during our benchtop tests are shown in Fig. 8-15. We have further determined that the gap between the frame transmission latency and the end-to-end latencies for our hardware setups are on average 60 ms for low-resolution and medium-resolution images, and 200 ms for high-resolution images. This latency increase accounts for the duration of grabbing a camera frame, converting it into RGB colorspace, and displaying the received RGB image on the recipient's monitor. These values are useful because they can be added to our latency results in Fig. 8-9 to obtain the corresponding expected end-to-end latencies for our pool trial sessions, since neither the frame grabbing delay nor the image display duration are dependent on the communication medium used to transmit these frames.

8.5 Field experiment Results

The success rate of receiving JPEG frames in all three video streaming field trials are plotted as a function of modem distance in Fig. 8-17. Focusing on the performance of the two-layer RS / LT encoding scheme, Fig. 8-16 depicts the percentage of fragments successfully decoded as a function of modem distance. In summary, we were able to transmit video frames with over 90 % success over distances of 3 m in the Charles



(a) Close range (15-25 m)

(b) Far range (35-40 m)

Figure 8-14: Histogram of time intervals between consecutive fragments received. Peaks in each histogram are centered around integer multiples of the transmitter's packet delay of 560 μs. The mean interval is marked by a vertical black dashed line.

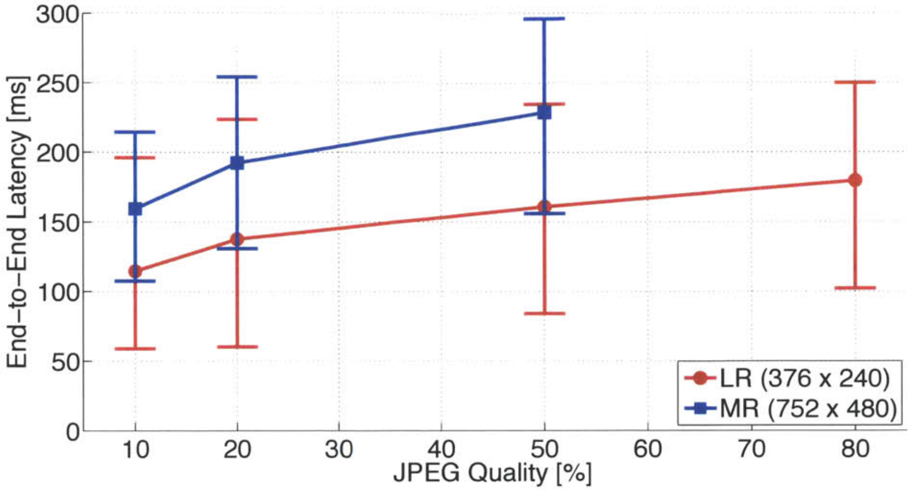
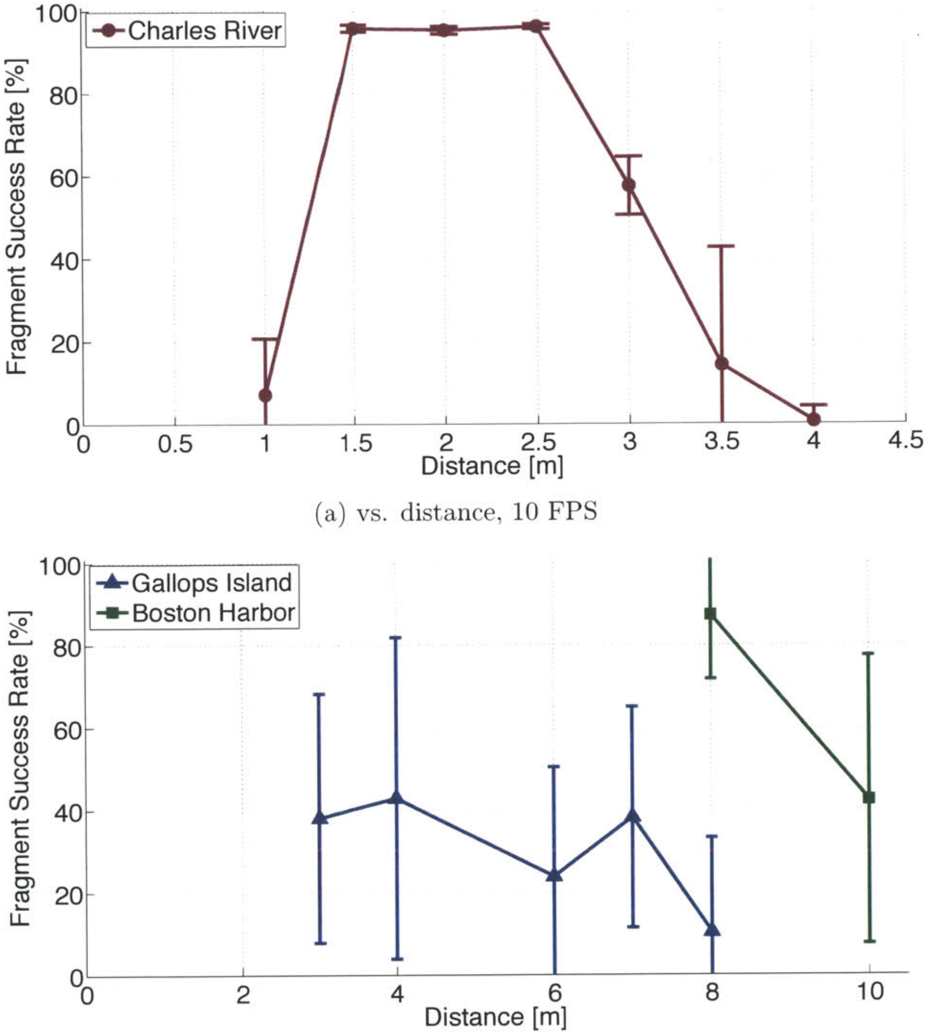


Figure 8-15: End-to-end latencies for capturing real-time video, streaming it via an unidirectional underwater free-space optical channel, and displaying the feed on a recipient computer. Vertical bars depict 5% and 95% percentiles.

River, 4 m near Gallops Island, and 8 m in the Boston Harbor near Long Wharf. Fragment success rates varied widely. For the Charles River experiments, which were more controller because of less wave action, we can see stable fragment reception of above 95 % at distances of 1.5 to 2.5 m and a constant decay of performance at increasing distances. In the case of the Boston Harbor experiments, were waves were stronger but holding the modem was still manageable, we can see acceptable fragment success rates of above 80 % at 8 m and above 40 % at 10 m. For the Gallops Island experiments, were we were fighting to keep the modems steady, the fragment success rate remained between 10 % and 43 %, but with a far stronger standard deviation when compared to the Charles River experiments.

8.6 Discussion

The parameter values used in our experiments are representative of typical video streaming settings, and were chosen to be of equal or better quality compared to similar studies conducted in the robotics literature. In particular, we observed anecdotally that images with JPEG compression qualities varying from 10% to 80% produced images that were subjectively rated as passably-visible to great quality, respectively. In



(a) vs. distance, 10 FPS

(b) vs. distance, 15 FPS

Figure 8-16: Fragment reception success rates for all experimental site trials, vs. distance. Vertical bars depict $\pm 1\sigma$.

addition, as shown in Fig. 8-10(b), these compressed images resulted in PSNR values above 30 dB, which is conventionally considered to be of decent quality [91, 123]. Similarly, our selection of frame rates and image resolutions provide competitive video accuracy compared to previous studies involving video streams [85, 128].

Fig. 8-7 revealed that all of the tested video settings have resulted in near perfect frame success rates within 25 m. In addition, configurations that yielded relatively small-sized frames have been shown to be robust at even greater distances, where the optical signal degradation were far more pronounced. Since the frame success rate is

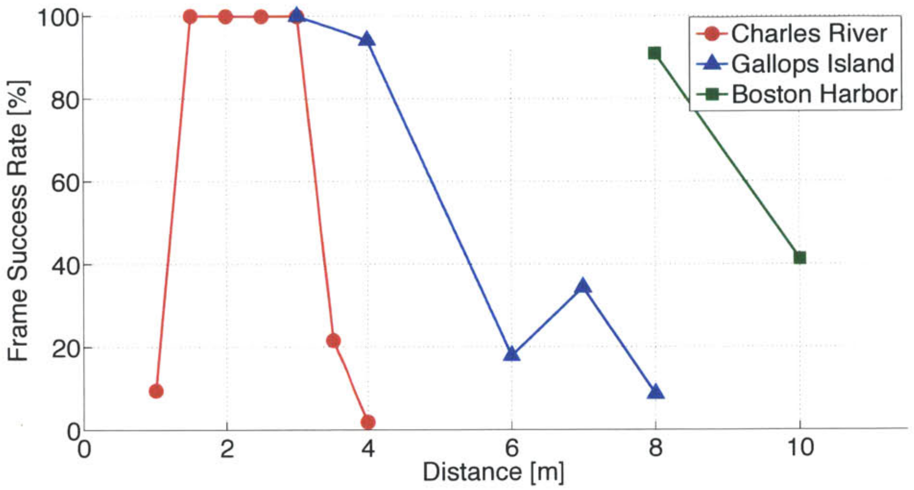


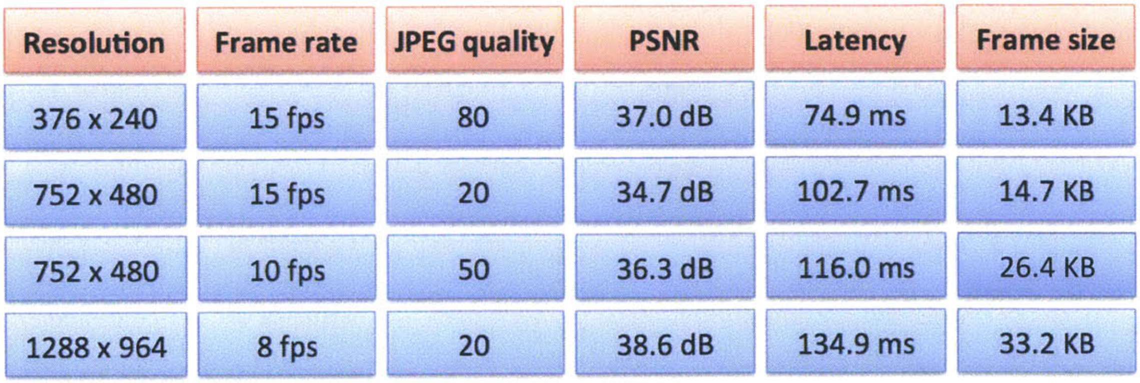
Figure 8-17: Frame reception success rates for all pool trials, vs. distance.

directly influenced by the underlying fragment success rate, it is not surprising that Fig. 8-7 and Fig. 8-11 depict similar-shaped curves. Nevertheless, the various frame success rate curves showed consistently steeper drop offs compared to the fragment success rate curve, which is attributed to the relatively strict maximum duration during which each video frame can be decoded, before the transmitter starts sending fragments corresponding to the subsequent frame.

By plotting the frame success rate as a function of the frame size in Fig. 8-8, we provide an indication of the expected transmission rates for framed data in general, and not necessarily just video frames. In particular, although our current system uses M-JPEG codec, which generates consistently-sized frames, the results in Fig. 8-8 can be used to predict the expected performance of variable-bitrate framed content such as videos encoded with inter-frame codecs like H.264 or Theora, or multi-modal content, for example high-bandwidth payload interleaved with low-bandwidth status and signaling meta-data in robot tele-operation applications.

Overall, our various video streaming tests showed similar performances at close ranges of up to 25 m, and also showed similar degradations as the distance between the two modems increased. This performance discrepancy can also be seen from the LT overhead plots, where for instance a noticeably more significant portion of frames were decoded using only the first N systematic fragments at close ranges (i.e. the red bar in Fig. 8-12(a)). The difference in the two ranges can further be attributed to the expected amount of packets dropped in Fig. 8-11, and in particular, in Fig. 8-14(a) the large peak around 560 us time interval indicates that a large percentage of fragments were received by the decoder at close ranges, whereas at farther distances (Fig. 8-14(b)) the distribution of fragments shifted to the second and third peaks, at twice and thrice the transmission interval, and consequently indicating the loss of one and two fragments, respectively.

A summary of the performance of our video streaming solution for several representative system configurations can be found in Fig. 8-18. These indicate that our system can achieve frame transmission latencies of under 135 ms, and corresponding end-to-end latencies between 170 ms and 400 ms. These suggest that our video streaming solution is capable of enabling a range of high-performance underwater applications based on various latency studies, including visual monitoring ([128], 300 ms), interaction with virtual environments ([81], 225 ms; [50], 313 ms), tele-surgery ([79], 250 ms), and tele-operated robotics ([95], 460 ms).



Resolution	Frame rate	JPEG quality	PSNR	Latency	Frame size
376 x 240	15 fps	80	37.0 dB	74.9 ms	13.4 KB
752 x 480	15 fps	20	34.7 dB	102.7 ms	14.7 KB
752 x 480	10 fps	50	36.3 dB	116.0 ms	26.4 KB
1288 x 964	8 fps	20	38.6 dB	134.9 ms	33.2 KB

Figure 8-18: Summary of underwater video streaming results for several representative video settings.

Chapter 9

Conclusion

Prior to this thesis, the recovery of data from underwater collection sites relied heavily on cables or directed human effort to pick up the data. Even large scale underwater observatory projects, such as NEPUTE [15], VENUS [16], and MARS [66], were based on a cabled system relying on humans to integrate new sensors into the system.

This thesis has made significant contributions towards the development of autonomous underwater observatories. We have presented a novel design for a modular robot, called AMOUR VI, that can be easily reconfigures in the field. Further we presented algorithms to estimate the robots configuration parameters and allow for model free control in six degrees of freedom. In addition, we have developed a novel underwater wireless communication device, called AquaOptical II, based on visible blue light. This device permutes data rates upwards of 4 Mbps at distances of up to 50 m and can be operated near the water surface, allowing for deployments in corral reefs and other sunlit areas.

Finally, we have integrated the systems presented in this thesis together with state of the art acoustic communications equipment to perform autonomous underwater large scale data retrieval, such as videos, audio, and image recordings. This is called underwater data muling, and our system performs data muling without the need for careful sensor placement or external localization of the sensor node or robot. All that is required is a rough estimate of the sensor location to within acoustic communication ranges, which can span from dozens to hundreds of kilometers. Using the algorithms presented in this thesis our robot is capable of navigating towards the sensor node using only the acoustic range measurements and then to maintain position using the optical signal strength measured by AquaOptical II. To our knowledge, the experiments presented in this thesis mark the first successful fully autonomous underwater data muling system deployment in the field. With demonstrated sensor node location from distances of 100 to 250 m and the demonstrated ability to download hundreds of Megabytes of data our system has pushed the envelope of what was possible in this field.

This thesis was dedicated to exploring the following thesis statement:

> *Underwater Data Retrieval can be effectively achieved using Data Muling with Combined Underwater Optical and Acoustic Communication.*

9.1 Contributions

This thesis makes the following contributions towards the development of autonomous underwater observatories:

- An novel agile hovering underwater robot that can be easily reconfigured by the end user in the field. This robot allows the user to "wrap" a set of thrusters around existing instrumentation and sensing equipment to provide precise actuation in six degrees of freedom. By allowing arbitrary configurations of thruster this robot spans the class of all robots who are purely actuated and controlled by thrusters.

- A control algorithm that spans the class of all vehicles with fixed thruster locations. This algorithm is versatile enough to allow up to six degrees of freedom control of arbitrary thruster configurations for the above robot and all other vehicles purely actuated and controlled by thrusters. It is only limited by the physical limitations imposed by the thruster configurations themselves, that is the number of degrees of freedom that a thruster configuration affords.

- An algorithm to estimate the thruster configuration parameters of the above controller. This allows the end user to simply attach the thrusters to the robot, after which our algorithms will estimate their locations and allow the user to control the robot without the need to provide configuration space parameters.

- An underwater wireless optical communication device that can communicate with data rates between 4 Mbps and 32 Mbps and distances of up to 50 m with ambient light present. The device can be connected to any ethernet device or network and will create a UDP bridge between the local network and any another networks connect to identical devices that are within range.

- An end-to-end model for estimating the optical signal strength for underwater wireless optical communication. This model allows predictions to be made about the expected signal strength, and these can be used to estimate the link quality or in particle filters for position estimation.

- Algorithms to perform underwater data muling using combined acoustic and optical communication. The algorithms presented allowed localization of the sensor node using only one way acoustic range measurements and position keeping of the robot above the sensor node by fusing the information from acoustic measurements and optical signal strength measurements with the estimates obtained from the IMU.

9.2 Lessons Learned

- Water is unforgiving to electronics! This is especially true for ocean water which will immediately destroy any electronic circuit it comes into contact with. This is a serious matter in an environment where water is omnipresent and an unseen human hair can cause a catastrophic failure of o-rings and significant equipment damage. Luckily this lessen was learned early on and most systems described in this thesis were implemented with a dual o-ring seal. The first o-ring usually serves to wipe away present contaminants while the second forms the seal. Further if the outer o-ring (the one in contact with water) fails, then often the inner o-ring with catch the leak.

- Easy of use when deploying is crucial to prevent errors even by the most advanced users. The necessity to open equipment in order to activate or start it in the field should generally be avoided. Such requirements to early versions of the modem have caused significant delays and damage to at least one modem unit because of operator error.

- A common communications link for all hardware is desirable. The decision to connect AMOUR's control system, the AquaOptical II modems, and the UNET 2 acoustic modems over ethernet significantly reduced the overhead required for operation and debugging of these system. Because we used the widely spread UDP datagram communication system it was very easy to access our hardware from any system or application connected to the network. What appears obvious in retrospect was not already implemented in previous versions of the hardware due to cost, size, and power considerations. However, this is rarely an issue anymore, as most micro controllers, processors, and computers are equipped with ethernet these days.

- Easy interfaces for debugging are a must. This was especially true for the optical modem where the addition of a software defined mode and raw signal forwarding over UDP made it possible to develop and test more complicated communications structures.

- Adaptability to the environment is of crucial importance. This observation was made in two separate instances. First, when deploying optical modems we notices significant differences in the absorption spectra of different waters. While blue light propagates best in pool waters, green light will penetrate farther in algae-rich ocean and costal waters. Our devices allowed us to easily swap the transmit LEDs to adapt to this situation, but what would have been even more desirable would be a system that can automatically pick from many different wavelengths. A similar experience was made using acoustic communications. The thrusters designed for our robot create a noticeable amount of noise and the UNET2 modems ability to monitor this noise and then actively adapt transmission settings to work around it proved crucial during our data muling experiments.

- The above acoustic communication noise issue highlights another important lessen learned: The ability to quickly debug. We designed most systems with very clear and accessible interfaces even to low level data. This was true the robot's IMU which was continuously reporting all system variables. It was also the case for the optical modem's "software defined radio" principle which allowed easy testing of parameters but also to listen in on established communication links with a third device. Finally, this ability was exhibited by the UNET2 modems. All of these systems would have been near-to-impossible to debug without the ability to monitor each and every state variable so that mistakes could be quickly identified. This is especially important in the field, where time is of even higher value than in the lab because of transportation and equipment rental costs.

9.3 Near-Term Improvements

Future improvements of the data muling system include the usage of the acoustic link to turn on and off both the optical modem and the acoustic beacons (or at least reducing their frequency) to save battery life while the robot is not in range. We also plan to extend the presented data muling system to three dimensions, which will allow for the nodes to be deployed at greater depths. The particle filter algorithm can be extended to utilize optical signal strength measurements in order to improve the robot's position keeping above the node. Further, we plan to extend experiments into the open ocean where the algorithm can be tested at distances of multiple kilometers.

A useful extension of our control algorithms would be to accommodate the docking of multiple robots together to form one larger robot. This could be achieved by simply sharing thruster position information and would allow to control the newly formed and larger vehicle using the same algorithm.

Regarding optical remote vehicle operation, the use of a single optical link to control many robots at once is a good candidate for future work. Using the current infrastructure, we have the bandwidth available to control tens of robots and the potential to expand the system to control over a hundred robots in the water. With the bi-directional system, the robots could communicate with high-speed links to enable groups of robots to autonomously and collaboratively perform tasks.

For future work we would like to further analyze the source of the prediction errors observed when the receiver was held at an angle greater than ±15 degrees. Additionally we would like to apply this model to other optical modems.

Further, we would like to augment the optical video transmission system with hardware-based video compression capabilities in order to further increase the range of frame resolution and quality afforded by our underwater video streaming solution. We are also investigating the integration of this video streaming solution along with a tele-control system for AUVs, to close the loop between the user and the robot. Finally, using a bi-directional communication channel would further allow us to dynamically optimize system parameters based on the current link quality and available bandwidth.

9.4 Future Work

The algorithms and systems presented in this thesis lay the groundwork for virtual presence and manipulations systems underwater. The agile AUV permits motions in six degrees of freedom, much like a human diver would move underwater. We see as a natural next step is the extension of the system with additional actuators and manipulators such as robotic arms. This will allow users to now only move in the environment but also interact with it. The wireless optical system creates the necessary backbone for low latency and fast data rate links to such a system. In particular, extending communications into full duplex mode, possibly with the use of multiple frequencies, will allow for quick and detailed control of the robot but also feedback to the user. Feedback to the user, such as haptic feedback, can be very useful during manipulation. Another useful extension to the communication system will be the development of reliable optical networks that can span large areas and efficiently relay information. We envision entire corral reefs and islands that have complete optical wireless network coverage allowing researchers to remotely operate these agile robots in real-time with direct high definition video feedback, much like a movable webcam or even a remote presence.

Bibliography

[1] Ravil Agishev, Barry Gross, Fred Moshary, Alexander Gilerson, and Samir Ahmed. Simple approach to predict apd/pmt lidar detector performance under sky background using dimensionless parametrization. *Optics and Lasers in Engineering*, 44(8):779–796, 2006.

[2] Ian F. Akyildiz, Dario Pompili, and Tommaso Melodia. Underwater acoustic sensor networks: Rcsearch challenges. *Ad Hoc Networks*, 3:257–279, 2005.

[3] Ambalux. Ambalux website, 2012.

[4] D. Anguita, D. Brizzolara, and G. Parodi. Building an underwater wireless sensor network based on optical: Communication: Research challenges and current results. In *Sensor Technologies and Applications, 2009. SENSORCOMM '09. Third International Conference on*, pages 476–479, june 2009.

[5] D. Anguita, D. Brizzolara, and G. Parodi. Optical wireless communication for underwater wireless sensor networks: Hardware modules and circuits design and implementation. In *OCEANS 2010*, pages 1–8, sept. 2010.

[6] D. Anguita, D. Brizzolara, and G. Parodi. Optical wireless communication for underwater wireless sensor networks: Hardware modules and circuits design and implementation. In *OCEANS 2010*, 2010.

[7] D. Anguita, D. Brizzolara, and G. Parodi. Vhdl modules and circuits for underwater optical wireless communication systems. *WTOC*, 9(9):525–552, September 2010.

[8] D. Anguita, D. Brizzolara, G. Parodi, and Qilong Hu. Optical wireless underwater communication for auv: Preliminary simulation and experimental results. In *OCEANS, 2011 - Spain*, pages 1–5, june 2011.

[9] K. J. Åström. Pid controllers : Theory, design and tuning. *Instrument Society of America*, 1995.

[10] K.J. Åström and T. Hägglund. *Advanced PID Control.* ISA-The Instrumentation, Systems, and Automation Society, 2006.

[11] G. Baiden and Y. Bissiri. High bandwidth optical networking for underwater untethered telerobotic operation. In *OCEANS 2007*, pages 1–9, 29 2007-oct. 4 2007.

[12] G. Baiden and Y. Bissiri. High bandwidth optical networking for underwater untethered telerobotic operation. In *OCEANS 2007*, 2007.

[13] Greg Baiden, Yassiah Bissiri, and Andrew Masoti. Paving the way for a future underwater omni-directional wireless optical communication systems. *Ocean Engineering*, 36(9-10):633–640, 2009.

[14] J.W. Bales and Chryssostomos C. High bandwidth, low-power, shot-range optical communication underwater. *Proc. 9th Int. Symp. on Unmanned Untethered Submersible Technology*, 1995.

[15] C. Barnes. Transforming the ocean sciences through cabled observatories. In *Aerospace conference, 2009 IEEE*, pages 1–2, march 2009.

[16] C.R. Barnes and V. Tunnicliffe. Building the world's first multi-node cabled ocean observatories (neptune canada and venus, canada): Science, realities, challenges and opportunities. In *OCEANS 2008 - MTS/IEEE Kobe Techno-Ocean*, pages 1–8, april 2008.

[17] Elizabeth Basha, Carrick Detweiler, Marek Doniec, Iuliu Vasilescu, and Daniela Rus. Using a multi-functional sensor network platform for large-scale applications to ground, air, and water tasks. In *Proceedings of the 6th Workshop on Hot Topics in Embedded Networked Sensors*, HotEmNets '10, pages 15:1–15:5, New York, NY, USA, 2010. ACM.

[18] Michael Bass, Casimer DeCusatis, Jay Enoch, Vasudevan Lakshminarayanan, Guifang Li, Carolyn Macdonald, Virendra Mahajan, and Eric Van Stryland. *Handbook of Optics, Third Edition Volume II: Design, Fabrication and Testing, Sources and Detectors, Radiometry and Photometry.* McGraw-Hill, Inc., New York, NY, USA, 3 edition, 2010.

[19] C.R. Benson, R.M. Dunbar, M.J. Ryan, E.H. Huntington, and M.R. Frater. Towards a dense high-capacity underwater acoustic network. In *Communication Systems (ICCS), 2010 IEEE International Conference on*, pages 386–389, nov. 2010.

[20] H. Brundage. Designing a wireless underwater optical communication system. Master's thesis, Massachusetts Institute of Technology, 2010.

[21] R. Camilli, A. Bowen, and N. Farr. Bright blue: Advanced technologies for marine environmental monitoring and offshore energy. In *OCEANS 2010 IEEE - Sydney*, pages 1–7, may 2010.

[22] M.A. Chancey. Short range underwater opical communication links. Master's thesis, North Carolina State University, 2005.

[23] N.A. Chaturvedi, A.K. Sanyal, and N.H. McClamroch. Rigid-body attitude control. *Control Systems, IEEE*, 31(3):30–51, june 2011.

[24] M. Chen, M. Zhou, and T. Li. The implementation of ppm in underwater laser communication system. In *Communications, Circuits and Systems Proceedings, 2006 International Conference on*, volume 3, pages 1901–1903, june 2006.

[25] M. Chitre, I. Topor, and Teong-Beng Koay. The unet-2 modem - an extensible tool for underwater networking research. In *OCEANS 2012 - Yeosu*, pages 1–9, May 2012.

[26] H.T. Choi, A. Hanai, S.K. Choi, and J. Yuh. Development of an underwater robot, ODIN-III. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 1, pages 836–841, 2003.

[27] B. Cochenour and L. Mullen. Channel response measurements for diffuse non-line-of-sight (nlos) optical communication links underwater. In *OCEANS 2011*, pages 1–5, sept. 2011.

[28] B. Cochenour, L. Mullen, and A. Laux. Phase coherent digital communications for wireless optical links in turbid underwater environments. In *OCEANS 2007*, pages 1–5, 29 2007-oct. 4 2007.

[29] B. Cochenour, L. Mullen, A. Laux, and T. Curran. Effects of multiple scattering on the implementation of an underwater wireless optical communications link. In *OCEANS 2006*, pages 1–6, Boston, MA, sept. 2006.

[30] ORION Executive Steering Committee et al. Ocean observatories initiative science plan. *Washington, DC*, 2005.

[31] CompuLab. fit-pc2 wiki, 2009.

[32] W.C. Cox. A 1mbps underwater communication system using a 405nm laser diode and photomultiplier tube. Master's thesis, North Carolina State University, 2008.

[33] W.C. Cox, K.F. Gray, J.A. Simpson, B. Cochenour, B.L. Hughes, and J.F. Muth. A mems blue/green retroreflecting modulator for underwater optical communications. In *OCEANS 2010*, pages 1–4, sept. 2010.

[34] W.C. Cox, J.A. Simpson, C.P. Domizioli, J.F. Muth, and B.L. Hughes. An underwater optical communication system implementing reed-solomon channel coding. In *OCEANS 2008*, pages 1–6, Sept. 2008.

[35] W.C. Cox, J.A. Simpson, and J.F. Muth. Underwater optical communication using software defined radio over led and laser based links. In *MILITARY COMMUNICATIONS CONFERENCE, 2011 - MILCOM 2011*, pages 2057–2062, nov. 2011.

[36] W.C. Cox, J.A. Simpson, and J.F. Muth. Underwater optical communication using software defined radio over led and laser based links. In *Military Communications Conference (MILCOM '11)*, pages 2057–2062, 2011.

[37] C. Detweiler, J. Leonard, D. Rus, and S. Teller. Passive mobile robot localization within a fixed beacon field. In *Proceedings of the 2006 International Workshop on Algorithmic Foundations of Robotics*, New York, aug 2006.

[38] C. Detweiler, I. Vasilescu, and D. Rus. An underwater sensor network with dual communications, sensing, and mobility. In *OCEANS Europe*, June 2007.

[39] C. Detweiler, I. Vasilescu, and D. Rus. An underwater sensor network with dual communications, sensing, and mobility. *OCEANS 2007 - Europe*, pages 1–6, June 2007.

[40] M. Dillinger, K. Madani, and N. Alonistioti. *Software defined radio: architectures, systems, and functions.* Wiley, 2003.

[41] M. Doniec, C. Detweiler, I. Vasilescu, and D. Rus. Using optical communication for remote underwater robot operation. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 4017–4022, oct. 2010.

[42] M. Doniec and D. Rus. Bidirectional optical communication with AquaOptical II. In *Communication Systems (ICCS), 2010 IEEE International Conference on*, pages 390–394, nov. 2010.

[43] Marek Doniec, Carrick Detweiler, Iuliu Vasilescu, Mandar Chitre, Matthias Hoffmann-Kuhnt, and Daniela Rus. Aquaoptical: A lightweight device for high-rate long-range underwater point-to-point communication. *Marine Technology Society Journal*, 44(4):55–65, July/August 2010.

[44] Marek Doniec, Iuliu Vasilescu, Mandar Chitre, Carrick Detweiler, Matthias Hoffmann-Kuhnt, and Daniela Rus. Aquaoptical: A lightweight device for high-rate long-range underwater point-to-point communication. In *OCEANS*, Biloxi, MS, Oct. 2009.

[45] Marek Doniec, Iuliu Vasilescu, Carrick Detweiler, and Daniela Rus. Complete se(3) underwater robot control with arbitrary thruster configurations. In *In Proc. of the International Conference on Robotics and Automation*, Anchorage, Alaska, May 2010.

[46] Marek W. Doniec, Iulian Topor, Mandar Chitre, and Daniela Rus. Autonomous, localization-free underwater data muling using acoustic and optical communication. In *ISER 2012, to appear in*, 2012.

[47] Gregory Dudek, Philippe Giguere, Chris Prahacs, Shane Saunderson, Junaed Sattar, Luz-Abril Torres-Mendez, Michael Jenkin, Andrew German, Andrew

Hogue, Arlene Ripsman, Jim Zacher, Evangelos Milios, Hui Liu, Pifu Zhang, Marti Buehler, and Christina Georgiades. Aqua: An amphibious autonomous robot. *Computer*, 40(1):46–53, jan. 2007.

[48] M. Dunbabin, P. Corke, I. Vasilescu, and D. Rus. Data muling over underwater wireless sensor networks using an autonomous underwater vehicle. *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 2091–2098, May 2006.

[49] M. Dunbabin, J. Roberts, K. Usher, G. Winstanley, and P. Corke. A hybrid AUV design for shallow water reef navigation. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 2105–2110, 2005.

[50] Stephen R. Ellis, Mark J. Young, Bernard D. Adelstein, and Sheryl M. Ehrlich. Discrimination of changes in latency during head movement. In *Proc. of the 8th Int. Conf. on Human-Computer Interaction (HCI '99)*, pages 1129–1133, 1999.

[51] N. Farr, A. Bowen, J. Ware, C. Pontbriand, and M. Tivey. An integrated, underwater optical / acoustic communications system. In *IEEE OCEANS Sydney 2010*, 2010.

[52] N. Farr, A. Bowen, J. Ware, C. Pontbriand, and M. Tivey. An integrated, underwater optical /acoustic communications system. In *OCEANS 2010 IEEE - Sydney*, pages 1–6, may 2010.

[53] N. Farr, A. Chave, L. Freitag, J. Preisig, S. White, D. Yoerger, and P. Titterton. Optical modem technology for seafloor observatories. In *OCEANS, 2005. Proc. of MTS/IEEE*, pages 928 –934 Vol. 1, 2005.

[54] N. Farr, A.D. Chave, L. Freitag, J. Preisig, S.N. White, D. Yoerger, and F. Sonnichsen. Optical modem technology for seafloor observatories. In *OCEANS 2006*, pages 1–6, sept. 2006.

[55] N. Farr, J. Ware, C. Pontbriand, T. Hammar, and M. Tivey. Optical communication system expands cork seafloor observatory's bandwidth. In *OCEANS 2010*, pages 1–6, sept. 2010.

[56] J. Farrell, B. Goldenthal, and K. Govindarajan. Connectionist learning control systems: submarine depth control. In *Decision and Control, 1990., Proceedings of the 29th IEEE Conference on*, volume 4, pages 2362–2367, 5-7 1990.

[57] T. Fossen and T. Johansen. A survey of control allocation methods for ships and underwater vehicles. In *Proceedings of the 14th IEEE Mediterranean Conference on Control and Automation*, Ancona, Italy, June 2006.

[58] C. Gabriel, M. Khalighi, S. Bourennane, P. Leon, and V. Rigaud. Channel modeling for underwater optical communication. In *GLOBECOM Workshops, 2011 IEEE*, pages 833–837, dec. 2011.

[59] Chris Gaskett, David Wettergreen, and Alexander Zelinsky. Reinforcement learning applied to the control of an autonomous underwater vehicle. In *In Proc. of the Australian Conference on Robotics and Automation (AUCRA'99)*, pages 125–131, 1999.

[60] 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(1):234–265, 2009.

[61] J.W. Giles and I.N. Bankman. Underwater optical communications systems. part 2: basic design considerations. In *Military Communications Conference. MILCOM 2005. IEEE*, volume 3, pages 1700–1705, oct. 2005.

[62] J. Guo, F.C. Chiu, and Chieh-Chih Wang. Adaptive control of an autonomous underwater vehicle testbed using neural networks. In *OCEANS '95. MTS/IEEE. Challenges of Our Changing Global Environment. Conference Proceedings.*, volume 2, pages 1033–1039, 9-12 1995.

[63] R.M. Hagem, D.V. Thiel, S.G. O'Keefe, A. Wixted, and T. Fickenscher. Low-cost short -range wireless optical fsk modem for swimmers feedback. In *Sensors, 2011 IEEE*, pages 258–261, oct. 2011.

[64] A Hanai, H. T Choi, S. K Choi, and J Yuh. Experimental study on fine motion control of underwater robots. *Advanced robotics: the international journal of the Robotics Society of Japan*, 18(10):963–978, 2004.

[65] Frank Hanson and Stojan Radic. High bandwidth underwater optical communication. *Appl. Opt.*, 47(2):277–283, 2008.

[66] R.G. Henthorn, B.W. Hobson, P.R. McGill, A.D. Sherman, and K.L. Smith. Mars benthic rover: In-situ rapid proto-testing on the monterey accelerated research system. In *OCEANS 2010*, pages 1–7, sept. 2010.

[67] K. Ishii, T. Ura, and T. Fujii. A feedforward neural network for identification and adaptive control of autonomous underwater vehicles. In *Neural Networks, 1994. IEEE World Congress on Computational Intelligence., 1994 IEEE International Conference on*, volume 5, pages 3216–3221, 27 1994.

[68] S. Jaruwatanadilok. Underwater wireless optical communication channel modeling and performance evaluation using vector radiative transfer theory. *Selected Areas in Communications, IEEE Journal on*, 26(9):1620–1627, December 2008.

[69] D. Kedar and S. Arnon. Non-line-of-sight optical wireless sensor network operating in multiscattering channel. *Applied Optics*, 45(33):8454–8461, 2006.

[70] D.B. Kilfoyle and A.B. Baggeroer. The state of the art in underwater acoustic telemetry. *Oceanic Engineering, IEEE Journal of*, 25(1):4–27, jan 2000.

[71] Shunji Kishimoto. High time resolution x-ray measurements with an avalanche photodiode detector. *Review of Scientific Instruments*, 63(1):824–827, 1992.

[72] Teong Beng Koay, P.J. Seeking, M. Chitre, Soo Pieng Tan, and M. Hoffmann-Kuhnt. Advanced panda for high speed autonomous ambient noise data collection and boat tracking - system and results. In *OCEANS 2006 - Asia Pacific*, pages 1–9, may 2006.

[73] V. S. Kodogiannis, P. J. G. Lisboa, and J. Lucas. Neural network modelling and control for underwater vehicles. *Artificial Intelligence in Engineering*, 10(3):203–212, 1996.

[74] Jongan Lee, Mootaek Roh, Jinsung Lee, and Doheon Lee. Clonal selection algorithms for 6-dof pid control of autonomous underwater vehicles. In Leandro de Castro, Fernando Von Zuben, and Helder Knidel, editors, *Artificial Immune Systems*, volume 4628 of *Lecture Notes in Computer Science*, pages 182–190. Springer Berlin / Heidelberg, 2007.

[75] Ke Li, Chien-Chung Shen, and Guaning Chen. Energy-constrained bi-objective data muling in underwater wireless sensor networks. In *Mobile Adhoc and Sensor Systems (MASS), 2010 IEEE 7th International Conference on*, pages 332–341, nov. 2010.

[76] J. Lorentz and J. Yuh. A survey and experimental study of neural network auv control. In *Autonomous Underwater Vehicle Technology, 1996. AUV '96., Proceedings of the 1996 Symposium on*, pages 109–116, 2-6 1996.

[77] Feng Lu, Sammy Lee, Jeffrey Mounzer, and Curt Schurgers. Low-cost medium-range optical underwater modem: short paper. In *Proceedings of the Fourth ACM International Workshop on UnderWater Networks*, WUWNet '09, pages 11:1–11:4, New York, NY, USA, 2009. ACM.

[78] M. Luby. Lt codes. In *Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on*, pages 271–280, 2002.

[79] M.J.H. Lum, J. Rosen, Hawkeye King, D.C.W. Friedman, T.S. Lendvay, A.S. Wright, M.N. Sinanan, and B. Hannaford. Teleoperation in surgical robotics - network latency effects on surgical performance. In *Proc. of the IEEE Int. Conf. on Engineering in Medicine and Biology Society (EMBC '09)*, pages 6860–6863, 2009.

[80] Philips Lumileds. LUXEON® Rebel and LUXEON® Rebel ES data sheet, 2012.

[81] I. Scott MacKenzie and Colin Ware. Lag as a determinant of human performance in interactive systems. In *Proc. of the INTERACT / CHI conference on human factors in computing systems (INTERCHI '93)*, pages 488–493, 1993.

[82] J. Mitola. The software radio architecture. *Communications Magazine, IEEE*, 33(5):26–38, may 1995.

[83] David Moore, John Leonard, Daniela Rus, and Seth Teller. Robust distributed network localization with noisy range measurements. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, SenSys '04, pages 50–61, New York, NY, USA, 2004. ACM.

[84] T.D. Nguyen, L-L. Yang, and L. Hanzo. Systematic luby transform codes and their soft decoding. In *Proc. of the IEEE Workshop on Signal Processing Systems (SiPS '07)*, pages 67–72, 2007.

[85] Barry O'Brien and Jesse Kovach. Underwater wireless optical communication channel modeling and performance evaluation using vector radiative transfer theory. *IEEE Journal on Selected Areas in Communications*, 26(9):1620–1627, 2008.

[86] Kyu-Hyun Oh, Jung-Yeob Kim, Ill-Woo Park, Jungho Lee, and Jun-Ho Oh. A study on the control of AUV's homing and docking. In *9th IEEE Conference on Mechatrinics and Machine Vision in Practice*, 9, pages 45–52, Thailand, 2002.

[87] Edwin Olson, John Leonard, and Seth Teller. Robust range-only beacon localization. In *Proceedings of Autonomous Underwater Vehicles, 2004*, 2004.

[88] G. Pang, T. Kwan, H. Liu, and Chi-Ho Chan. Optical wireless based on high brightness visible leds. In *Industry Applications Conference, 1999. Thirty-Fourth IAS Annual Meeting. Conference Record of the 1999 IEEE*, volume 3, pages 1693–1699, 1999.

[89] C. Pelekanakis, M. Stojanovic, and L. Freitag. High rate acoustic link for underwater video transmission. In *OCEANS 2003*, pages 1091–1097, 2003.

[90] Inc. Penguin Automated Systems. Penguin automated systems, inc. website, 2012.

[91] P. Piñol, M. Martinez-Rach, O. Lopez, M.P. Malumbres, and J. Oliver. Analyzing the impact of commercial video encoders in remotely teleoperated mobile robots through ieee 802.11 wireless network technologies. In *Proc. of the 5th IEEE Int. Conf. on Industrial Informatics*, pages 425–430, 2007.

[92] C. Pontbriand, N. Farr, J. Ware, J. Preisig, and H. Popenoe. Diffuse high-bandwidth optical communications. In *OCEANS 2008*, pages 1–4, sept. 2008.

[93] I. S. Reed and G. Solomon. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304, 1960.

[94] M.A.J. Rivera, B.E. Lawton, and C. Chyrssostomidis. Reef missions: Engaging students in science and the marine environment using an autonomous underwater vehicle. In *OCEANS 2011*, pages 1–7, sept. 2011.

[95] Bill Ross, John Bares, David Stager, Larry Jackel, and Mike Perschbacher. An advanced teleoperation testbed. In *Field and Service Robotics*, volume 42 of *Springer Tracts in Advanced Robotics*, pages 297–304, 2008.

[96] Felix Schill, Uwe R. Zimmer, and Jochen Trumpf. Visible spectrum optical communication and distance sensing for underwater applications. In *AGRA*, 2004.

[97] R.C. Shah, S. Roy, S. Jain, and W. Brunette. Data mules: modeling a three-tier architecture for sparse sensor networks. In *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, pages 30–41, may 2003.

[98] L.T. Sharpe, Stockman A., Jagla W., and Jägle H. A luminous efficiency function, v*(λ), for daylight adaptation. *Journal of Vision*, 5(11), December 2005.

[99] J.A. Simpson, W.C. Cox, J.R. Krier, B. Cochenour, B.L. Hughes, and J.F. Muth. 5 mbps optical wireless communication with error correction coding for underwater sensor nodes. In *OCEANS 2010*, pages 1–4, sept. 2010.

[100] J.A. Simpson, W.C. Cox, J.R. Krier, B. Cochenour, B.L. Hughes, and J.F. Muth. 5 mbps optical wireless communication with error correction coding for underwater sensor nodes. In *OCEANS 2010*, 2010.

[101] J.A. Simpson, B.L. Hughes, and J.F. Muth. A spatial diversity system to measure optical fading in an underwater communications channel. In *OCEANS 2009, MTS/IEEE Biloxi - Marine Technology for Our Future: Global and Local Challenges*, pages 1–6, oct. 2009.

[102] Jean-Jacques E. Slotine and H. Asada. *Robot Analysis and Control*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1992.

[103] J.H. Smart. Underwater optical communications systems part 1: variability of water optical parameters. In *Military Communications Conference, 2005. MILCOM 2005. IEEE*, volume 2, pages 1140–1146, oct. 2005.

[104] Peter P. Smyth, Philip L. Eardley, Kieran T. Dalton, David R. Wisely, Paul McKee, and David Wood. Optical wireless: a prognosis. In *Proc. SPIE*, volume 2601, pages 212–225, 1995.

[105] M. Stojanovic. Recent advances in high-speed underwater acoustic communications. *Oceanic Engineering, IEEE Journal of*, 21(2):125–136, apr 1996.

[106] Sebastian Thrun. Probabilistic robotics. *Commun. ACM*, 45(3):52–57, March 2002.

[107] M. Tivey. A low power, low cost, underwater optical communications system. *Ridge 2000 Events*, pages 27–29, April 2004.

[108] Y. Tsuchida, Hama N., and Takahata M. An optical telemetry system for underwater recording of electromyogram and neuronal activity from non-tethered crayfish. *J of Neuroscience Methods*, 2004.

[109] J. Vaganay, P. Baccou, and B. Jouvencel. Homing by acoustic ranging to a single beacon. In *OCEANS 2000*, volume 2, pages 1457–1462, 2000.

[110] J. Vaganay, M. Elkins, D. Esposito, W. O'Halloran, F. Hover, and M. Kokko. Ship hull inspection with the HAUV: US navy and NATO demonstrations results. In *OCEANS 2006*, pages 1–6, 2006.

[111] P. van de Ven, C. Flanagan, and D. Toal. Identification of underwater vehicle dynamics with neural networks. In *OCEANS '04. MTTS/IEEE TECHNO-OCEAN '04*, volume 3, pages 1198–1204, 9-12 2004.

[112] Pepijn W.J. van de Ven, Colin Flanagan, and Daniel Toal. Neural network control of underwater vehicles. *Engineering Applications of Artificial Intelligence*, 18(5):533–547, 2005.

[113] J. Vander Hook, P. Tokekar, E. Branson, P. Bajer, P. Sorensen, and V. Isler. Local search strategy for active localization of multiple invasive fish. In *Proc. International Symposium on Experimental Robotics*, 2012.

[114] I. Vasilescu. *Using Light Underwater: Devices, Algorithms and Systems for Maritime Persistent Surveillance.* PhD thesis, MIT, February 2009.

[115] I. Vasilescu, C. Detweiler, M. Doniec, D. Gurdan, S. Sosnowski, J. Stumpf, and D. Rus. Amour v: A hovering energy efficient underwater robot capable of dynamic payloads. *International Journal of Robotics Research (IJRR)*, 2010.

[116] I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke. Data collection, storage, and retrieval with an underwater sensor network. In *SenSys*, pages 154–165, 2005.

[117] I. Vasilescu, K. Kotay, D. Rus, L. Overs, P. Sikka, M. Dunbabin, P. Chen, and P. Corke. Krill: An exploration in underwater sensor networks. *Embedded Networked Sensors, 2005. EmNetS-II. The Second IEEE Workshop on*, pages 151–152, May 2005.

[118] I. Vasilescu, P. Varhavskaya, K. Kotay, and D. Rus. Autonomous modular optical underwater robot (amour) design, prototype and feasibility study. In *Proc. IEEE ICRA 2005)*, pages 1603–1609, Barcelona, Spain, April 2005.

[119] I. Vasilescu, P. Varshavskaya, K. Kotay, and D. Rus. Autonomous modular optical underwater robot (amour) design, prototype and feasibility study. *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 1603–1609, April 2005.

[120] Iuliu Vasilescu, Carrick Detweiler, and Daniela Rus. Aquanodes: an underwater sensor network. In *WuWNet '07: Proceedings of the second workshop on Underwater networks*, pages 85–88, New York, NY, USA, 2007. ACM.

[121] Iuliu Vasilescu, Carrick Detweiler, and Daniela Rus. Color-accurate underwater imaging using perceptual adaptive illumination. In *Proc. of Robotics: Science and Systems (RSS 2010)*, 2010.

[122] J. Verne. 20,000 leagues under the sea. 1870. *Trans. Anthony Bonner. New York: Bantam*, 1962.

[123] Stephen Welstead. *Fractal and Wavelet Image Compression Techniques*. SPIE Publications, 1999.

[124] J. Yuh. Design and control of autonomous underwater robots: A survey. *Auton. Robots*, 8(1):7–24, 2000.

[125] S. Zhao, J. Yuh, and H.T. Choi. Adaptive DOB control of underwater robotic vehicles. In *OCEANS, 2001. MTS/IEEE Conference and Exhibition*, volume 1, pages 397–402, 2001.

[126] M. Zhuang and D.P. Atherton. Automatic tuning of optimum pid controllers. *Control Theory and Applications, IEE Proceedings D*, 140(3):216–224, may 1993.

[127] JG Ziegler and NB Nichols. Optimum settings for automatic controllers. *trans. ASME*, 64(11), 1942.

[128] Li Zuo, Jian Guang Lou, Hua Cai, and Jiang Li. Multicast of real-time multiview video. In *Proc. of the 2006 IEEE Int. Conf. on Multimedia and Expo*, pages 1225–1228, 2006.