

ABSTRACT

Title of Thesis: **AUTONOMOUS ESTIMATION AND GUIDANCE
OF AN AMPHIBIOUS QUADROTOR FOR
BISTATIC UNDERWATER LASER IMAGING**

Nathan Toombs
Master of Science, 2022

Thesis Directed by: **Professor Derek Paley**
**Department of Aerospace Engineering and
Institute for Systems Research**

Underwater object classification by unmanned underwater vehicles (UUVs) is a critical task that is made difficult in shallow waters with concentrated particulate matter. Bistatic laser imaging is a current area of research that is more effective than traditional optical methods, but it requires separation of the laser receiver from the UUV-mounted laser emitter. This work explores the prospect of performing bistatic laser imaging with the receiver mounted to a quadrotor unmanned aerial vehicle (UAV). To facilitate the imaging application, estimation and guidance algorithms are developed to autonomously locate and track a UUV-mounted laser with an amphibious UAV. The UAV is equipped to carry a receiver payload in safe above-water flight and water landings. To represent the received laser measurements, laser intensity models are developed based on the distributions of the decollimated lasers used in the imaging application. The UAV autonomy is validated both in a reduced-order simulation environment and with the hardware testbed.

AUTONOMOUS ESTIMATION AND GUIDANCE OF AN
AMPHIBIOUS QUADROTOR FOR BISTATIC UNDERWATER
LASER IMAGING

by

Nathan Toombs

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2022

Advisory Committee:
Professor Derek Paley, Chair/Advisor
Professor Michael Otte
Professor Mumu Xu

© Copyright by
Nathan Toombs
2022

Dedication

To my wife Katelyn and my daughter Kendrie, without whom I wouldn't be who I am today, and without whom I couldn't imagine tomorrow.

Acknowledgments

First, I would like to express my sincere gratitude to my advisor, Prof. Derek Paley, for his continuous support during my time at the University of Maryland. He has been an excellent advisor and professor, giving me aid and instruction in my courses and in preparing this thesis. His guidance was invaluable for me to determine what direction to take in my research, and to help clarify my thoughts and my work. I also appreciate his patience and understanding with my unique circumstances and my proclivity to finish tasks later than anticipated.

Special thanks to all of my fellow lab members at the CDCL. PhD candidates Sydrak Abdi, Rachel Suitor, and Anthony Thompson have played key roles in helping me through class assignments, serving as sounding boards for my ideas and struggles, and providing moral support through the era of COVID. David Peterson has been a support from the early stages of this thesis. From proof-of-concept to hardware design to exchanging ideas and finally to providing feedback on this document, he has played a significant role in my thesis' progression. Other lab members that have been of noteworthy support are Michael Hinckley, Kruti Bhingradiya, Alexander Donkels, Wei-Kuo Yen, and Wei Cui.

There are many other UMD faculty and staff members whose help I am grateful for. Ivan Pensky has been generous, and forgiving, in allowing me use of the Brin Family

Drone Lab and other resources and in ensuring that I follow safety guidelines in my hardware testing. Dr. David Akin has been helpful in coordinating with me to use the Neutral Buoyancy Research Facility, and I also appreciate Steve Hand and Dmitry Akmal for helping me ensure that my experiments were conducted safely. Additionally, I am grateful to the staff at the UMD Aerospace Office for their continued support for me and my fellow students.

Many thanks to my associates at NAWCAD, including Linda Mullen, Derek Alley, DJ Cooper, and Lauren Boulay. I appreciate their support not only in helping define my research problem, but also in designing and delivering components for the experimental testbed. I am also greatly appreciative of NAWCAD Grant No. N004212120002, which has generously funded my research for this thesis.

Lastly, this endeavor would not have been possible without the support of my wife, Katelyn, and of my other family members. They have continuously pushed me to be my best self, and have helped me immensely over the past two years. In many of my busiest days and nights, Katelyn single-handedly cared for our baby and kept our home running while working as well, and I am indebted to her aid and support.

Table of Contents

Dedication	ii
Acknowledgements	iii
Table of Contents	v
List of Figures	vii
List of Abbreviations	ix
Chapter 1: Introduction	1
1.1 Motivation	1
1.2 Relation to Previous Ongoing Work	4
1.3 Technical Approach	5
1.4 Contributions	6
1.5 Outline of Thesis	7
Chapter 2: Experimental Testbed	8
2.1 Unmanned Aerial Vehicle	8
2.2 Remotely Operated Vehicle	12
2.3 Laser and Receiver	14
2.4 System Interface	15
Chapter 3: Unmanned Vehicle Planar Dynamics and Controls	17
3.1 Unmanned Aerial Vehicle Model	17
3.2 Remotely Operated Vehicle Model	23
Chapter 4: Laser Measurement Modeling	29
4.1 Laser Intensity Models	31
4.2 Laser Detectability and Effective Divergence	37
Chapter 5: Estimation and Guidance Algorithms	41
5.1 Vehicle State Estimation	41
5.2 Laser Intensity Estimation	43
5.3 Autonomous Quadrotor Guidance	50
Chapter 6: Simulation and Experimental Results	55

6.1	Simulation Environment	55
6.2	Simulation Results	56
6.3	Experimental Setup	64
6.4	Experimental Results	66
Chapter 7: Conclusion		72
7.1	Summary of Contributions	72
7.2	Suggestions for Ongoing and Future Work	73
Bibliography		74

List of Figures

2.1	Image of the amphibious quadrotor UAV	10
2.2	UAV software and interfaces diagram	12
2.3	Image of the BlueROV2 Heavy	13
2.4	Images of the ROV-mounted laser and the UAV-mounted receiver	15
2.5	System interface diagram	16
3.1	Reference frames for the planar UAV and ROV models	18
3.2	Free-body diagram showing the forces and moments experienced by the UAV	19
3.3	2D cascaded PID controller that provides UAV motor thrust outputs from commanded position or velocity	20
3.4	Simulation results of the UAV dynamics and controls	23
3.5	Free-body diagram showing the forces and moments experienced by the ROV	25
3.6	Cascaded PID controller that provides ROV motor thrust outputs from commanded position or velocity and attitude	26
3.7	Simulation results of the ROV dynamics and controls	28
4.1	A laser profile from the side, and a cross-section of the profile showing the laser-illuminated area	30
4.2	The Gaussian beam waist ($w(z)$) shown as related to the laser divergence	32
4.3	Comparison between super-Gaussian intensity model and diffused laser intensity from ThorLabs Engineered Diffuser	33
4.4	Gaussian laser model with increasing order (n); as the order increases, the transient region between minimum and maximum intensity shortens	34
4.5	The receiver frame ($\mathcal{S} = (S, \mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3)$) and the laser frame ($\mathcal{L} = (L, \mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3)$) with respective unit vectors; note that the laser and receiver point along their third axis ($\mathbf{l}_3, \mathbf{s}_3$)	36
4.6	It is shown that the receiver can be in the LIA without detecting the laser (a); the relative size of the LDA (dotted) compared to the LIA (solid) (b)	38
4.7	The unaltered laser models compared to the expected detectable models.	38
4.8	A comparison between the super-Gaussian models with experimental data of varying receiver FOV	39
5.1	Position coordinates r and θ of the LIA centerpoint P relative to the UAV-mounted receiver S	47

6.1	Animated UAV, ROV, laser, and receiver FOV from the simulation	56
6.2	Simulation results of the concentric search algorithm	57
6.3	Simulation results using the gradient ascent algorithms for a Gaussian laser model and a super-Gaussian laser model	58
6.4	Simulation results of the EKF on a flat-top laser	59
6.5	Simulated results of the EKF when the receiver remains in the flat-top area	60
6.6	Simulated results of the search algorithm with position hold guidance and a moving ROV	61
6.7	Simulated results of the search pattern with gradient ascent on a flat-top laser	62
6.8	Simulated results of the EKF estimation and guidance over a stationary flat-top distributed laser	63
6.9	Simulated results of the EKF estimation and guidance over a moving flat-top distributed laser	63
6.10	Simulated results of using ROV tracking to guide the UAV	64
6.11	Image of the Neutral Buoyancy Research Facility water tank	65
6.12	Hardware implementation of the concentric search algorithm, starting near the position coordinate $(0, 0)$	67
6.13	Laser intensity measurements made in the LDA in hardware testing	68
6.14	Results from the gradient ascent estimation algorithm in handheld testing	69
6.15	Results from the extended Kalman filter estimation algorithm in handheld testing	71

List of Abbreviations

ADC	Analog-to-Digital Converter
DVE	Degraded Visual Environment
DVL	Doppler Velocity Log
EKF	Extended Kalman Filter
ESC	Electronic Speed Controller
FOV	Field-of-View
FXTI	Fathom-X Tether Interface
IMU	Inertial Measurement Unit
LDA	Laser Detectable Area
LIA	Laser-Illuminated Area
NBRF	Neutral Buoyancy Research Facility
NED	North-East-Down
QGC	QGroundControl
ROS	Robot Operating System
ROV	Remotely Operated Underwater Vehicle
SBL	Short Baseline
SONAR	Sound Navigation and Ranging
SWaP	Size Weight and Power
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
USV	Unmanned Surface Vessel
UUV	Unmanned Underwater Vehicle
VIO	Visual-Inertial Odometry

Chapter 1: Introduction

1.1 Motivation

The motivation for this thesis stems from the need for high-resolution imaging in shallow, degraded visual environments (DVEs). In marine environments, various imaging techniques are used to visualize and understand unknown objects. Currently, sound navigation and ranging (SONAR) devices are commonly used in maritime applications. The technology is mature, having been developed in a wide breadth of applications for decades, and it operates over long distances in water because of the speed and range of sound propagation. However, SONAR imaging systems do not provide the kind of resolution that optical devices (e.g. cameras) operate with. SONAR also suffers from multi-path signals in shallow water, and the more robust systems that can provide a better resolution are too large for small, remotely operated underwater vehicles (ROVs) that would be used in these scenarios [1]. More robust SONAR systems can provide higher resolution, but size constraints make them unattractive for small ROVs.

For these reasons, small ROVs typically operate with visible camera-based sensors. While they can produce high-resolution imagery in nominal conditions, cameras perform poorly in DVEs, such as turbid coastal waters and harbors [2]. The performance degradation in these environments is due to light-scattering particulates in the water. To find a solution

to this issue, laser imaging systems are actively being developed [1, 3–5]. For example, there are defense-related systems that operate either below water or through the air-water interface that take advantage of lasers for enhanced resolution to detect and classify underwater mines [6], [7]. However, these systems are large, bulky, and expensive. Further development has been directed toward creating systems of a lower size, weight, and power (SWaP) to be used on small ROV platforms by allocating some of the required equipment to a separate platform.

One of the key factors in the large SWaP of these laser imaging systems is their pseudo-monostatic nature. A monostatic device acts both as the transmitter and receiver, often used to describe traditional radar or sonar systems. However, it is often the case that different devices are required for the transmitter and receiver. A pseudo-monostatic device is one in which the distance between the transmitter and receiver is much smaller than the range to the target, such as with laser imaging systems where the emitter and receiver are both located on the same ROV [8]. Pseudo-monostatic laser imaging systems tend to be large, both because there are two devices mounted to the same vehicle (each with their own weight and power requirements) and because the devices are mounted as far from each other as possible.

Multistatic sensing involves using multiple transmitters or receivers. For example, bistatic systems have a single transmitter and receiver, but the distance between them is comparable to the distance to the target. Adjusting the geometry between the transmitter, receiver, and target can increase performance, especially if the transmitter and receiver are on moving platforms [2]. A bistatic laser imaging system would have the advantage of reducing SWaP of the ROV by moving the receiver to another platform. It was also

found that the separate emitter and receiver can be positioned to provide a higher-quality image than with using a pseudo-monostatic system [2]. Additionally, it was shown that the laser receiver can be located above water, without suffering issues with the laser signal travelling across the air-water interface [2], such that the vehicle carrying the receiver does not need to be confined to the water domain.

In initial testing, the laser receiver was mounted above a test facility water tank. For outdoor testing, the receiver was mounted on an unmanned surface vessel (USV) [9]; however, the instability of the water's surface can cause the receiver to lose sight of the laser. This work explores the possibility of mounting the laser receiver on a multirotor UAV. Multicopters or multirotor drones are simple aerial vehicles that are able to hover in place and vertically take-off and land. While their endurance is less than similarly sized fixed-wing aerial vehicles, their ability to hover makes them advantageous for this application. A hovering multirotor UAV would be able to maintain level, stable flight for the laser receiver to increase the likelihood of laser detection.

A challenge that arises with using a UAV-mounted receiver is the localization of the ROV in order to acquire the laser signal. Knowledge of the ROV position relative to the UAV is needed in order to place the receiver within the range and line-of-sight of the emitter. With a USV, an acoustic underwater localization device can be used to get range and heading from the USV to the ROV. However, the air-water interface between the ROV and UAV rules out this approach. Also, while estimating the relative position of the ROV can be helpful, the objective is to position the receiver such that it detects the laser signal, so the ideal position of the receiver may not be directly above the ROV. The objective is more directly met by localizing the receiver within the spatial gradient of the

laser to position it in an ideal location.

1.2 Relation to Previous Ongoing Work

Multistatic sensing with robot agents has long been a topic of interest. Multistatic radar has been used to locate and track small UAVs [10]. Robot networks have been used to distribute multistatic emitters and receivers, both with UAVs [11, 12] and with UUVs [13, 14]. However, little research has been conducted in multistatic laser imaging beyond the motivating application.

Also related to this topic is spatial gradient estimation and source localization. More typical examples of spatial gradients are phenomena like environmental pressure gradients, temperature gradients, chemical spills or gas leaks, and more. Mobile robots have been used to estimate gradients and identify sources in these applications, from individual agents [15], to unmanned ground vehicle (UGV) swarms [16], to UAV swarms [17], and even cross-domain applications [18]. While lasers are not typically considered as being represented by a spatial gradient, estimating gradients from other electromagnetic fields has been researched [15, 19].

Localization with respect to lasers and other optical devices is important because of their increased usage for communication, such as in the bistatic laser imaging application. LEDs have been used for combined communication and localization [20], even using independent robots for the emitter and receiver [21]. Research has also been conducted on using lasers to communicate across an air-water interface, with the intended use case being for communication between UAVs and UUVs [22]. This work's efforts toward

improving the localization of the receiver within the laser field can benefit research in these areas.

1.3 Technical Approach

To test UAV-aided bistatic laser imaging, a quadrotor UAV is designed taking into account the unique requirements of the application. The UAV needs to carry a laser receiver payload weighing about 1.5 kg long enough for calibration and testing of the imaging system. The UAV design also considers the operational environment, i.e. the need to fly safely above water, by implementing an amphibious design. Also to be designed are a communication network between the UAV, the ROV, and a user interface to allow for autonomous coordination.

Autonomous flight capabilities, including a means to estimate the UAV's position, are formulated to determine the relative position between the laser emitter and receiver. The UAV autonomy is designed using estimation and control theory, using the laser as a means to localize the emitter and ROV. In order to estimate the position of the receiver within the laser field, laser intensity models are developed to represent the laser types used in hardware testing. An existing Gaussian beam intensity model is analyzed for far field use with lasers decollimated by a diverging lens. For lasers diffused into a flat-top distribution, a super-Gaussian distribution is examined. Although the laser passes through an air-water interface before reaching the receiver, it is assumed that the refraction at the surface is negligible, as well as the attenuation through water. Another simplifying assumption is that the ROV and UAV remain level, such that the emitter and receiver are

both aligned with the vertical axis. The estimation and guidance algorithms are designed for use with these laser models.

Before implementing any autonomous behaviors in hardware, they are tested in a reduced-order simulation. Vehicle dynamics are simplified to the plane defined by the forward and vertical axes. Additionally, laser models are imported to produce simulated results.

1.4 Contributions

The contributions of this thesis are:

1. **A testbed to experimentally replicate cross-domain coordination for bistatic laser imaging, including a custom amphibious quadrotor UAV.** The UAV carries a laser receiver payload for an adequate flight time for testing the application. The UAV is equipped with sensors and computing ability to perform on-board position estimation and autonomous flight in indoor and outdoor situations, as well as novel buoyant landing gear to allow for safe water landing and skimming maneuvers.
2. **A super-Gaussian laser intensity model that captures both the divergence and distribution of Gaussian and flat-top lasers.** The model describes the intensity measurement as a function of the axial and radial distances from the laser emitter to the receiver. It is appropriate for the far field intensity of lasers that have been decollimated with either a diverging lens or an engineered diffuser.
3. **Estimation and guidance algorithms to autonomously locate and track the upwelling laser with the UAV.** An optimization-inspired gradient ascent algorithm

commands the UAV to climb the estimated spatial laser gradient. An extended Kalman filter (EKF) uses available sensor measurements to estimate the relative position of the laser axis in the plane of the receiver. Guidance algorithms were developed to perform a constant-velocity search and to use state estimates to position the UAV and receiver in the detectable region of the laser. An autonomy framework was suggested to use available estimates and measurements to switch between guidance behaviors.

4. **Simulation and hardware results.** 2D dynamic models for a quadcopter UAV and a fully-actuated ROV were developed. 2D control algorithms were implemented for both the UAV and ROV models. Results for the performance of the estimation and guidance algorithms were produced in the simulation environment, and additionally with the hardware testbed.

1.5 Outline of Thesis

Chapter 2 describes the hardware testbed developed for this work. The quadrotor UAV design is shown, as well as the communication and coordination framework between the UAV, ROV, and groundstation. Chapter 3 presents the 2D dynamics and controls for the simulated UAV and ROV. Chapter 4 presents the development of the laser intensity models used in simulation and in algorithm development. Chapter 5 describes the estimation and guidance algorithms developed for the UAV autonomy. Chapter 6 presents the simulation and hardware results from the aforementioned algorithms, and Chapter 7 summarizes and concludes the thesis.

Chapter 2: Experimental Testbed

This application of bistatic laser imaging requires a unique and complex system of hardware and software components and interfaces. While the end-goal of the application is to conduct imaging of unclassified objects in open-water, this testbed was designed for performing initial testing indoors.

2.1 Unmanned Aerial Vehicle

2.1.1 Hardware Design

Before a design was made, commercial UAVs were considered, but none were found to adequately meet the constraints of the application. Instead, the UAV hardware was custom-designed with the unique aspects of the application and testing in mind. The minimum components for a quadrotor UAV are the frame, drive (motors, propellers, electronic speed controllers, and batteries), a flight controller with an inertial measurement unit (IMU) for state estimation, and an RC receiver. To implement autonomous behaviors and to facilitate data transmission from the receiver to the ground station, the UAV also requires a computing device with wireless connectivity, as well as a means of localizing the UAV indoors. The primary goal of the UAV is to carry the laser receiver payload,

which is expected to have a mass of about 1.5 kg. Because the testing of the bistatic laser system requires time for adjusting and calibration before the images are produced, flight longevity was an important aspect to consider. Most indoor research drones are designed to fly for just a few minutes, which would be insufficient for this project. In order to increase flight time, additional battery capacity can be added, but this comes with the problem that increasing total mass requires stronger rotors (motors and propellers). Generally, rotor efficiency is best when the propeller diameter is maximized, but larger propellers require a larger airframe. Not only does this increase mass, but a larger frame and propellers pose more of a safety risk when flying in a confined, indoor environment.

To keep the safety risk at a manageable level, the maximum length from propeller to propeller across the diagonal was chosen at 1 m. Using an optimization algorithm¹ previously developed by the author, the motors, propellers, and batteries were chosen to keep with this constraint while also performing adequately for the application. With a payload, the UAV is designed to fly for about 15-25 min, depending on the flight conditions.

Another unique aspect of the project is that the UAV would be flying almost exclusively above water. If there were to be a malfunction or any other situation where the UAV would be required to land immediately, the drone would almost certainly be submerged. Waterproofing the entire UAV would be unreasonable at this stage of research, especially as the interfaces between the drone and payload are being developed; also, waterproofing would likely mean designing and manufacturing a sealed enclosure, which would be an immense sink of resources and time. Instead, landing gear floats were designed to

¹<https://github.com/natetoombs/Multirotor-Sizing-Methodology>



Figure 2.1: Image of the amphibious quadrotor UAV

support the weight of the UAV on the surface of the water (see Figure 2.1). The floats, made of closed-cell foam that does not absorb water, are placed at the ends of each of the four arms of the UAV to maximize stability during landing and in the presence of disturbances. Additionally, the rigid foam floats can be used as landing gear for landing on the ground/on solid surfaces.

2.1.2 Software Design

The computing components on the UAV are the flight controller and the companion computer. The flight controller is a CC3D REVO F4 board running the ROSflight firmware, which is an open-source autopilot developed by a team of university researchers [23]. The flight controller has a built-in IMU, which provides a state estimate for the on-board PID angular rate controller. The flight controller is connected to the RC receiver; these components, connected to the quadrotor drive parts, are enough for manual flight of the vehicle.

The autonomous functions are implemented in the companion computer. The single-board computer selected is an NVIDIA Jetson TX2, which is connected to the flight controller via a USB connection. ROS (Robot Operating System) is used as the framework to connect the various software and hardware components. ROSflight has a node that runs on the computer to receive sensor data and send commands from and to the flight controller. The IMU data is used in an error-state EKF estimator, which produces a state estimate. This state estimate is used by the cascaded PID controller. The estimator and controller come from an open-source software package named ROScopter², developed by the same university lab that developed ROSflight.

An Intel Realsense T265 tracking camera was chosen to provide visual-inertial odometry (VIO) for the position estimate. The T265 camera connects with the computer via USB, and connects to ROS with a driver developed by the team at Intel RealSense. The measurements are also sent to the estimator. The command messages sent to the controller come either from a Wi-Fi connection to the groundstation computer, or from the onboard autonomy node that implements the methods described in Chapter 5. The autonomy uses measurements from the laser receiver, which is connected to the companion computer via USB and has a driver that converts the receiver serial output to ROS messages. The controller takes high-level commands (position or velocity), and it outputs throttle and angular rate commands to the flight controller.

²<https://github.com/byu-magicc/roscopter>

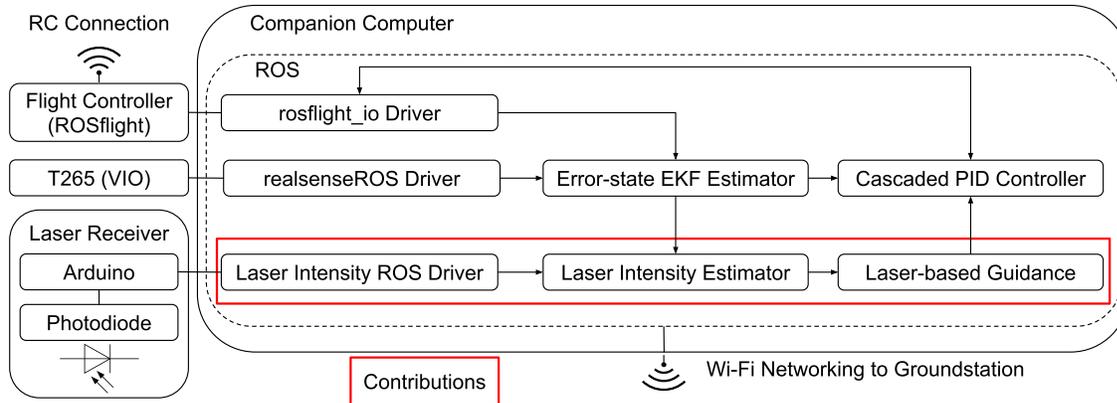


Figure 2.2: UAV software and interfaces diagram

2.2 Remotely Operated Vehicle

The ROV used in this testbed is the BlueRobotics BlueROV2 Heavy. Much like a UAV, the ROV principally consists of a frame, drive (motors, propellers, electronic speed controllers (ESCs), and a battery), a flight controller, a companion computer, and a means to communicate remotely. Some additional components that are important for the ROV are a camera, lights, buoyancy foam, and ballast weights. The electronic components are contained in two sealed compartments that are rated to a depth of 100 meters. There are a total of 8 thrusters (motors and propellers), which are used to provide full six-degree-of-freedom control. There is a heavy-duty tether that connects the ROV to the Fathom-X Tether Interface (FXTI), which connects the tether to an ethernet connection to the groundstation computer.

The flight controller on the ROV is a Pixhawk 1. While this is typically used for a UAV, it works well for an ROV because it controls the motors, reads sensors, and communicates to a groundstation, in a very similar way to how a UAV does. The



Figure 2.3: Image of the BlueROV2 Heavy

Pixhawk runs the ArduSub³ firmware, an open-source project for controlling ROVs. The flight controller also connects to a Raspberry Pi 3 companion computer, which runs the ArduSub-Raspbian operating system specifically built for communicating with an ROV. This companion computer relays information, commands, and the video stream through the tether to the groundstation computer. Onboard sensors include an IMU, a magnetometer, and a barometer, which can be used by the Pixhawk controller in a variety of control modes.

The ROV is also equipped to be used with an underwater acoustic positioning system. The Waterlinked GPS is a short baseline (SBL) system that uses four acoustic receivers to calculate the 3D position of a transmitter mounted to the ROV. The position estimate is then sent to the groundstation computer. While the system works well in open

³<https://www.ardusub.com/>

water, it proved inaccurate when used at the indoor test facility, likely due to multipath reflections from the metal struts in the tank walls.

2.3 Laser and Receiver

The testbed uses a mock laser system to represent the relevant aspects of the bistatic laser imaging application. Instead of a laser encoded with data from the imager [2], a green, low-power (0.045 W) laser is used. The laser is enclosed in a water-tight tube, along with either a lens or a diffuser, and a mirror to direct the laser upwards. The tube is attached to the top of the ROV (see Figure 2.4(a)). The laser is connected to a top-side power source through the tether that goes to the ROV.

The receiver consists of a photodiode, a light filter, a source transformation circuit, an Arduino Mini Pro, a USB connection, and a 9-volt battery for a power source. Light reaches the photodiode through a 532 nm filter, which matches the color of the laser as to reduce the ambient light noise. In another effort to reduce ambient light noise, the photodiode field-of-view (FOV) is restricted to 10 degrees. As the photodiode receives light, it acts as a current source, increasing the output current based on the amount of green light received. The source transformation circuit converts the current source to a voltage source, which can be read by the analog-to-digital converter (ADC) on the Arduino. The Arduino reads the voltage signal and sends it as a serial signal over the USB output. The receiver components are housed in a water-tight enclosure, which is mounted to the UAV (see Figure 2.4(b)).

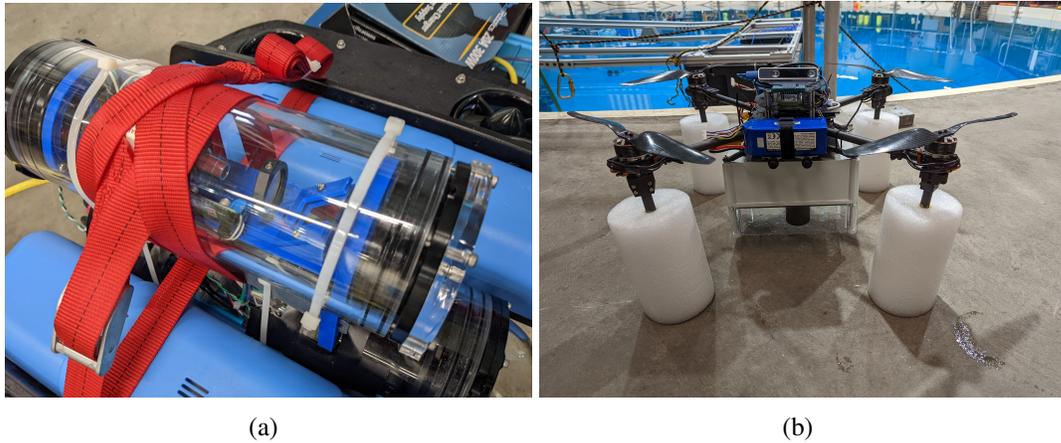


Figure 2.4: Images of the ROV-mounted laser (a) and the UAV-mounted receiver (b)

2.4 System Interface

A complete system diagram is shown in Figure 2.5. The groundstation computer serves as the central hub for UAV/ROV communication. Via ethernet, it connects to the FXTI to receive data and video from the ROV, and also to the Waterlinked GPS system. The data from the ROV autopilot, which uses the MAVLink protocol, are converted to ROS messages with a ROS software node. This data is then read by the ground control program QGroundControl (QGC), which displays important information about the ROV, shows the video feed, and allows for autonomous mission planning if there is a position measurement. A USB joystick connected to the computer also interfaces with the ROV through QGC to give manual control of the ROV.

Because the MAVlink data have been converted to ROS messages, depth data from the ROV can be used to calculate the distance between the laser emitter and receiver for use in the EKF. The messages are sent through the ROS framework over the Wi-fi connection to the UAV. If the ROV position is estimated and is accurate, then its position

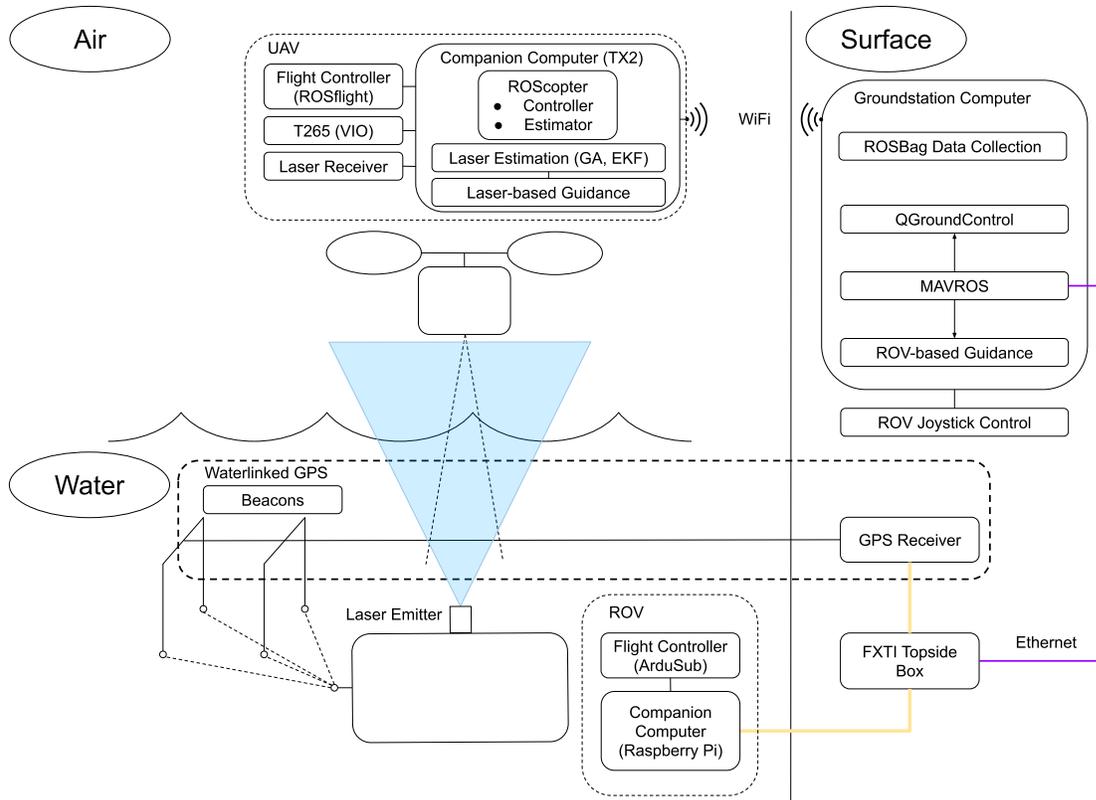


Figure 2.5: System interface diagram

can be used as a command directly to the UAV controller.

Chapter 3: Unmanned Vehicle Planar Dynamics and Controls

The following models were developed for 2D simulation of the vehicles involved in the bistatic laser imaging application.

3.1 Unmanned Aerial Vehicle Model

3.1.1 Quadrotor UAV Kinematics and Dynamics

Figure 3.1 shows the reference frames used for the UAV and ROV. Both frames are represented in the north-east-down (NED) coordinate system. The inertial frame is defined as $(\mathcal{I} = (I, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3))$. The drone frame $(\mathcal{D} = (D, \mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3))$ is represented by a rotation (θ_D) about the inertial frame \mathbf{e}_2 axis (see Figure 3.1; \mathbf{e}_2 is orthogonal to \mathbf{e}_1 and \mathbf{e}_3 , and would be coming out of the page). The rotation from the north (lateral) and down (vertical) coordinates of the drone frame to the inertial frame is given by the following rotation matrix:

$${}^{\mathcal{I}}R^{\mathcal{D}} = \begin{bmatrix} \cos(\theta_D) & \sin(\theta_D) \\ -\sin(\theta_D) & \cos(\theta_D) \end{bmatrix} \quad (3.1)$$

The states of the drone are its position, orientation, velocity, and angular velocity. The position vector $(\mathbf{r}_{D/I})$ is represented in the inertial frame with the coordinates x_D and z_D being the horizontal and vertical positions, respectively. The attitude is described by the rotation from the inertial frame, given by the pitch angle θ_D . As is standard in many

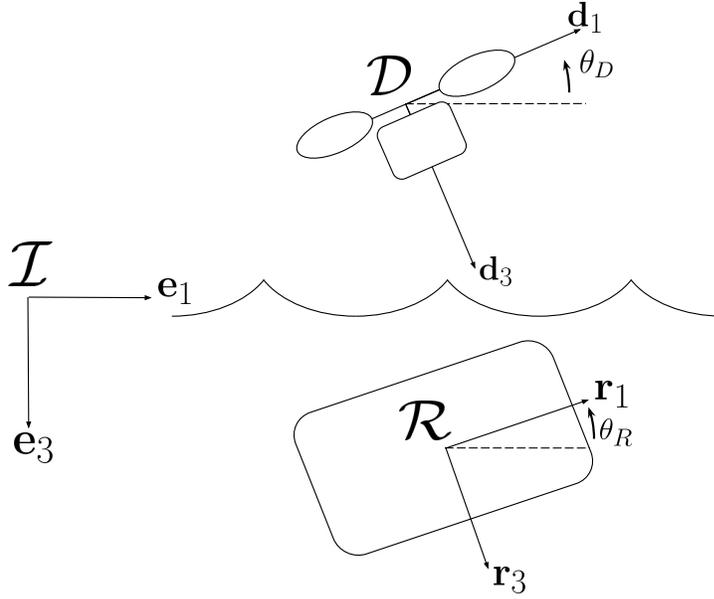


Figure 3.1: Reference frames for the planar UAV and ROV models

aerodynamic applications, the inertial velocity vector (${}^{\mathcal{I}}\mathbf{v}_{D/I}$) is represented in the drone body frame, where u_D is the velocity in the drone frame north direction \mathbf{d}_1 and w_D is the velocity in the drone frame down direction \mathbf{d}_3 . The angular velocity (${}^{\mathcal{I}}\omega^{\mathcal{D}}$) is represented by the derivative of the orientation angle ($\dot{\theta}_D$).

The forces and moments induced by the drone are dependent on the rotor thrusts (T_1 and T_2 ; see Figure 3.2). Only two rotors are used because they are sufficient to show the quadrotor dynamics in two dimensions. The effects of these thrusts can be expressed as a total thrust T and a moment M . Their relationship to the drone thrusts is

$$\begin{bmatrix} T \\ M \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ -d & d \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} \quad (3.2)$$

with d being the symmetrical distance from each motor to the center of the drone. Inverting this equation yields the mixing equation used to convert from a desired thrust and moment to the desired motor thrusts.

The other forces experienced by the drone are due to gravity and aerodynamic drag

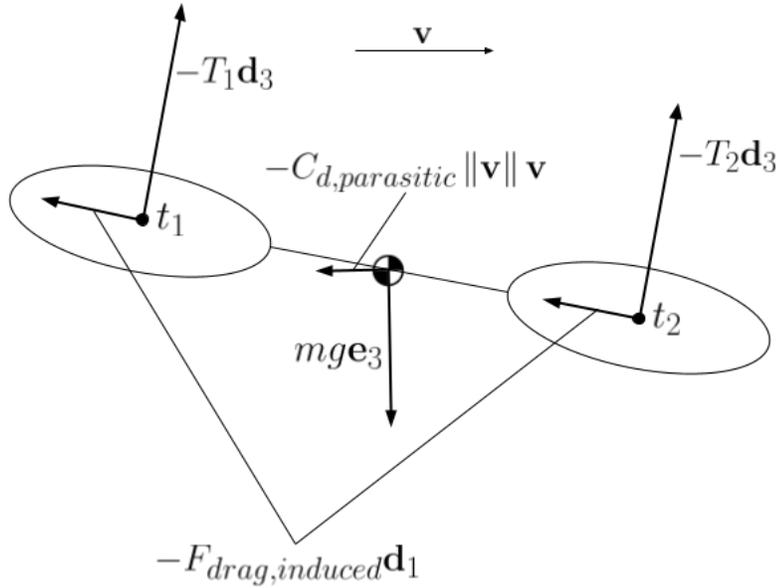


Figure 3.2: Free-body diagram showing the forces and moments experienced by the UAV (see Figure 3.2). Because the induced drag ($F_{drag,induced}$) has a significantly greater effect than the parasitic drag ($F_{drag,parasitic}$) at the low speeds of this project, and because the parasitic drag acts in the same direction as the induced drag while the drone is at hover, the parasitic drag is neglected. The induced drag is proportional to the speed of the rotors, but because the rotor speed is not modeled here, the drag can instead be represented as proportional to the motor thrust [24]. The drag is modeled as

$$F_{drag,induced} \approx -TC_d u_D, \quad (3.3)$$

where C_d is a non-dimensional drag constant. The equations of motion are as follows:

$$\frac{\mathcal{I}d}{dt} [\mathbf{r}_{D/I}]_{\mathcal{I}} = [\mathcal{I}\mathbf{v}_{D/I}]_{\mathcal{I}} = \mathcal{I}R^{\mathcal{D}} [\mathcal{I}\mathbf{v}_{D/I}]_{\mathcal{D}} = \mathcal{I}R^{\mathcal{D}} \begin{bmatrix} u_D \\ w_D \end{bmatrix}_{\mathcal{D}}, \quad (3.4)$$

$$\mathcal{I}\omega^{\mathcal{D}} = \dot{\theta}_D, \quad (3.5)$$

$$\frac{\mathcal{I}d}{dt} [\mathcal{I}\mathbf{v}_{D/I}]_{\mathcal{D}} = \frac{1}{m} \left({}^{\mathcal{D}}R^{\mathcal{I}} \begin{bmatrix} 0 \\ mg \end{bmatrix} + \begin{bmatrix} 0 \\ -T \end{bmatrix} + \begin{bmatrix} -TC_d u_D \\ 0 \end{bmatrix} \right) - \begin{bmatrix} \dot{\theta}_D w_D \\ -\dot{\theta}_D u_D \end{bmatrix}, \quad (3.6)$$

$$\frac{\mathcal{I}d}{dt} \mathcal{I}\omega^{\mathcal{D}} = \ddot{\theta}_D = \frac{1}{J} M, \quad (3.7)$$

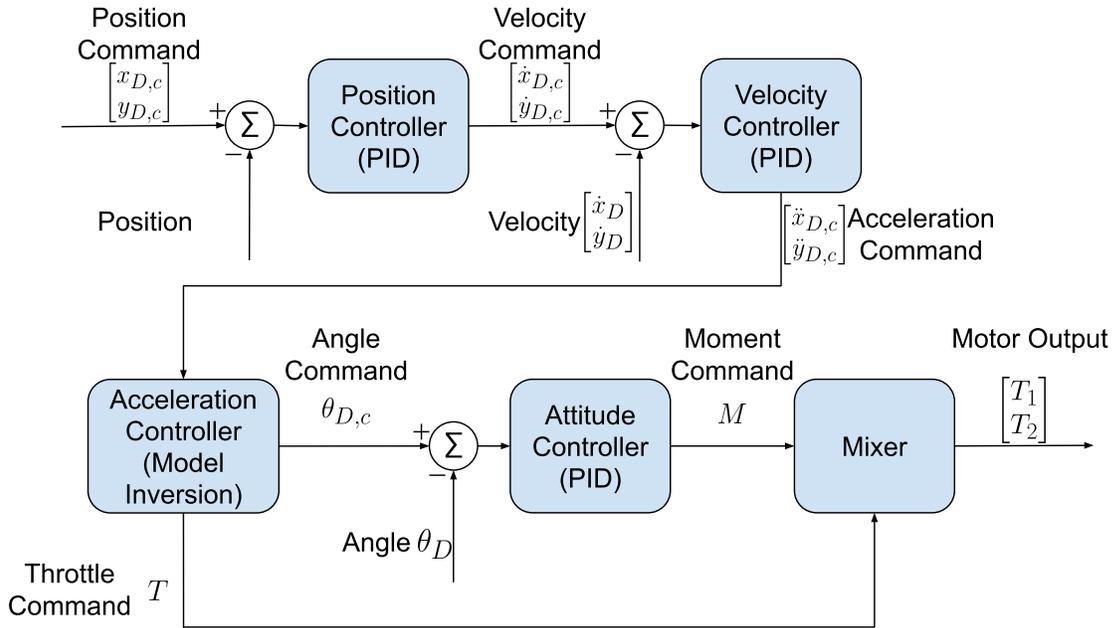


Figure 3.3: 2D cascaded PID controller that provides UAV motor thrust outputs from commanded position or velocity

where J is the moment of inertia of the UAV.

3.1.2 Quadrotor UAV Control

To have flexibility in the control input, the drone uses a cascaded PID controller (see Figure 3.3). This allows the controller input to be a position command or a velocity command, both of which are useful to this research. If a position command is specified, the error between the command and the state is the input for the position controller, which calculates a PID output for each of the position states (x_D and z_D). The output is a velocity command for each translational state in the inertial frame.

If a position command is instead specified from the UAV guidance, then the position controller can be omitted by giving the velocity command directly to the velocity controller. The velocity controller operates similar to the position controller, regardless of whether

the command comes from the UAV guidance or from the position controller. The output is an acceleration command for each translational state in the inertial frame.

Because the 2D quadrotor UAV can only directly apply a force in the body frame down axis (\mathbf{d}_3) and a moment about the east axis (\mathbf{d}_2), it is underactuated; in other words, the UAV can only indirectly control its motion in the body frame north direction (\mathbf{d}_1). For example, a hovering UAV can move up and down by changing the net thrust, and it can rotate by creating a differential between the front and rear thrusts, but it can't directly control its movement forward and back.

In order to achieve lateral movement, the quadrotor takes advantage of its dynamics to pitch in the desired direction. With an increase in thrust, the UAV can maintain altitude while the thrust vector accelerates it laterally. The desired angle and thrust for this motion are calculated in the acceleration controller, which uses model inversion to get these controls from the acceleration command.

To calculate the thrust and angle commands, a simplified dynamic model that neglects air drag is used. This model is used to calculate the thrust and pitch angle that must be computed to reach the commanded accelerations $\ddot{x}_{\mathcal{D},c}$ and $\ddot{z}_{\mathcal{D},c}$:

$$\frac{\mathcal{I}d}{dt} [\mathcal{I}\mathbf{v}_{\mathcal{D}/I}]_{\mathcal{I},desired} = \begin{bmatrix} \ddot{x}_{\mathcal{D},c} \\ \ddot{z}_{\mathcal{D},c} \end{bmatrix} = \frac{1}{m} \left(\begin{bmatrix} 0 \\ mg \end{bmatrix} + \mathcal{I}R^c \begin{bmatrix} 0 \\ -T_c \end{bmatrix} \right). \quad (3.8)$$

T_c and $\mathcal{I}R^c$ are the thrust and rotation that must be computed to satisfy the dynamics. The computed pitch angle $\theta_{\mathcal{D},c}$ defines the rotation from the computed frame \mathcal{C} :

$$\mathcal{I}R^c = \begin{bmatrix} \cos(\theta_{\mathcal{D},c}) & \sin(\theta_{\mathcal{D},c}) \\ -\sin(\theta_{\mathcal{D},c}) & \cos(\theta_{\mathcal{D},c}) \end{bmatrix}. \quad (3.9)$$

Because the UAV current orientation can be different from the computed orientation, the thrust command will be modified until the current and computed orientations align.

Next, T_c and $\theta_{D,c}$ are computed. Let $T_c = ma_c$, where a_c is the magnitude of the acceleration from the computed rotor thrust. From manipulating Equation 3.8, the acceleration is found as:

$$a_c = \sqrt{(g - \ddot{z}_{D,c})^2 + (\ddot{x}_{D,c})^2}. \quad (3.10)$$

By plugging Equation 3.10 into Equation 3.8, the computed pitch angle is found to be

$$\theta_{D,c} = \arcsin\left(\frac{\ddot{x}_{D,c}}{a_c}\right). \quad (3.11)$$

From the computed inputs T_c and $\theta_{D,c}$, the input commands must be calculated. The computed pitch $\theta_{D,c}$ will be used as the command for the attitude controller, but the difference between the computed orientation and the current orientation of the UAV must be considered in the thrust command T . To ensure that the thrust doesn't drive the UAV away from its desired acceleration, the computed thrust is projected onto the current body frame thrust axis:

$$T = \frac{T_c}{\cos(\theta_D)} \quad (3.12)$$

As the UAV orientation approaches the computed orientation, then the thrust command approaches the computed thrust.

The last remaining controller is the attitude controller, which takes as an input the error in the desired pitch angle and the actual angle. It uses PID control to output a moment command (M). The moment command and the thrust command are run through the mixer to determine the motor outputs:

$$\begin{bmatrix} T1 \\ T2 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2d} \\ \frac{1}{2} & -\frac{1}{2d} \end{bmatrix} \begin{bmatrix} T \\ M \end{bmatrix}. \quad (3.13)$$

Figure 3.4 shows simulated results of the UAV being commanded to follow some position commands. Note that the lateral position response has an initial delay before

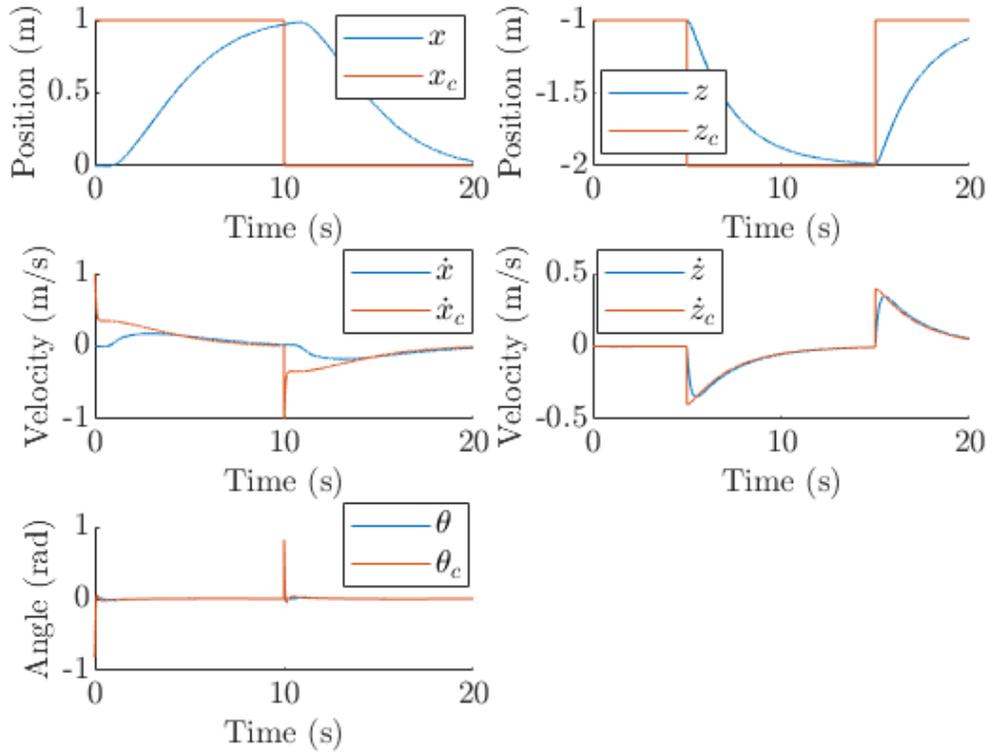


Figure 3.4: Simulation results of the UAV dynamics and controls

reducing error; this is because the UAV must first change its attitude before it can track laterally. This delay is not a factor in the altitude dynamics.

3.2 Remotely Operated Vehicle Model

3.2.1 ROV Kinematics and Dynamics

Like the UAV frame, the ROV frame ($\mathcal{R} = (R, \mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3)$) is represented by a rotation (θ_R) about the inertial frame \mathbf{e}_2 axis (see Figure 3.1). The rotation from the north (lateral) and down (vertical) coordinates of the drone frame to the inertial frame is given

by the following rotation matrix:

$$\mathcal{I}R^{\mathcal{R}} = \begin{bmatrix} \cos(\theta_R) & \sin(\theta_R) \\ -\sin(\theta_R) & \cos(\theta_R) \end{bmatrix} \quad (3.14)$$

The states of the ROV are its position, orientation, velocity, and angular velocity. The position vector ($\mathbf{r}_{R/I}$) is represented in the inertial frame with the coordinates x_R and z_R being the horizontal and vertical positions, respectively. The orientation is the rotation from the inertial frame, given by θ_R . As with the UAV, the velocity vector (${}^{\mathcal{I}}\mathbf{v}_{R/I}$) is represented in the ROV body frame, where u_R is the velocity in the ROV frame north direction and w_R is the velocity in the ROV frame down direction. The angular velocity (${}^{\mathcal{I}}\omega^{\mathcal{R}}$) is represented by the derivative of the orientation ($\dot{\theta}_R$).

The forces and moments induced by the ROV are dependent on the motor thrusts (T_1, T_2, T_3 , and T_4 ; see Figure 3.5). The effects of these thrusts can be expressed as a total thrust \mathbf{T} and a moment M . The total thrust \mathbf{T} is then broken into the ROV frame north and down directions, i.e. T_{north} and T_{down} . Their relationship to the ROV thrusts is

$$\begin{bmatrix} T_{down} \\ M \\ T_{north} \end{bmatrix} = \begin{bmatrix} -1 & -1 & 0 & 0 \\ -d & d & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} \quad (3.15)$$

with d being the symmetrical distance between the upward-facing motors and the center of the ROV. Assuming that T_3 and T_4 are equal and opposite, the above equation can be manipulated to convert from desired thrusts and moment to the desired motor thrusts.

The other forces experienced by the ROV are due to gravity, drag, and buoyancy. Due to the higher density of water, the ROV experiences not just a drag force along its direction of motion (\mathbf{F}_{drag}), but also a considerable drag moment resisting its angular velocity (M_{drag}). The buoyant force (F_{buoy}) acts in the opposite direction of gravity, and

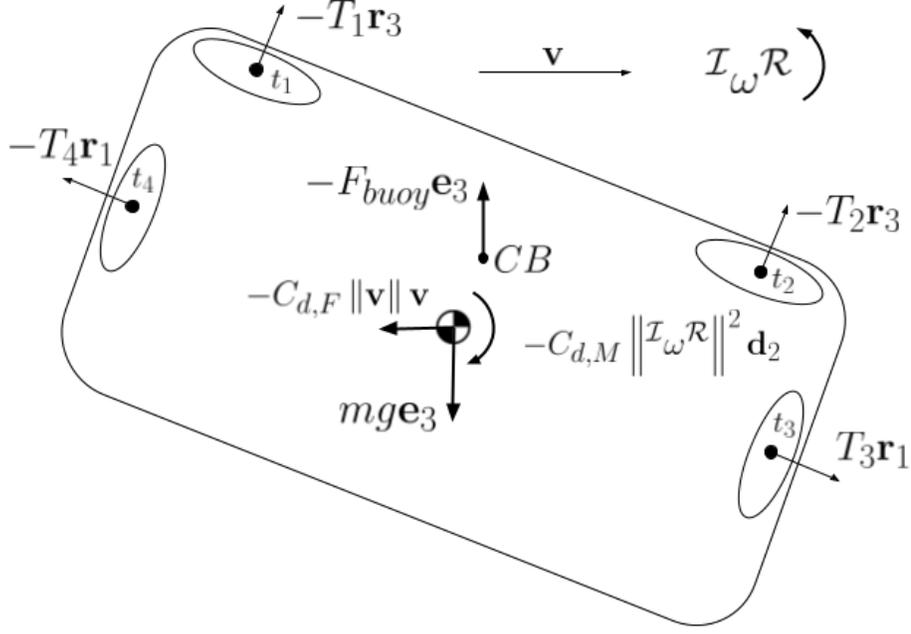


Figure 3.5: Free-body diagram showing the forces and moments experienced by the ROV. It also adds a self-righting moment to the ROV because the center of buoyancy (CB) is above the center of mass of the ROV. The equations of motion are as follows:

$$\frac{\mathcal{I}d}{dt} [\mathbf{r}_{R/I}]_{\mathcal{I}} = [\mathcal{I}\mathbf{v}_{R/I}]_{\mathcal{I}} = {}^{\mathcal{I}}R^{\mathcal{R}} [\mathcal{I}\mathbf{v}_{R/I}]_{\mathcal{R}} = {}^{\mathcal{I}}R^{\mathcal{R}} \begin{bmatrix} u_R \\ w_R \end{bmatrix}_{\mathcal{R}}, \quad (3.16)$$

$$\mathcal{I}\omega^{\mathcal{R}} = \dot{\theta}_R, \quad (3.17)$$

$$\frac{\mathcal{I}d}{dt} [\mathcal{I}\mathbf{v}_{R/I}]_{\mathcal{R}} = \frac{1}{m} \left({}^{\mathcal{R}}R^{\mathcal{I}} \begin{bmatrix} 0 \\ mg - F_{buoy} \end{bmatrix} + \begin{bmatrix} 0 \\ -T \end{bmatrix} + [\mathbf{F}_{drag}]_{\mathcal{R}} \right) - \begin{bmatrix} \dot{\theta}_R w_R \\ -\dot{\theta}_R u_R \end{bmatrix}, \quad (3.18)$$

$$\frac{\mathcal{I}d}{dt} \mathcal{I}\omega^{\mathcal{R}} = \ddot{\theta}_R = \frac{1}{J} (M - M_{drag}), \quad (3.19)$$

3.2.2 ROV Control

Similar to the drone controller, the ROV controller uses cascaded control to allow for either position or velocity inputs (see Figure 3.6). However, because the ROV is fully actuated, the attitude control is decoupled from the position control. This allows the ROV to hold any orientation while still moving to a desired position (or velocity).

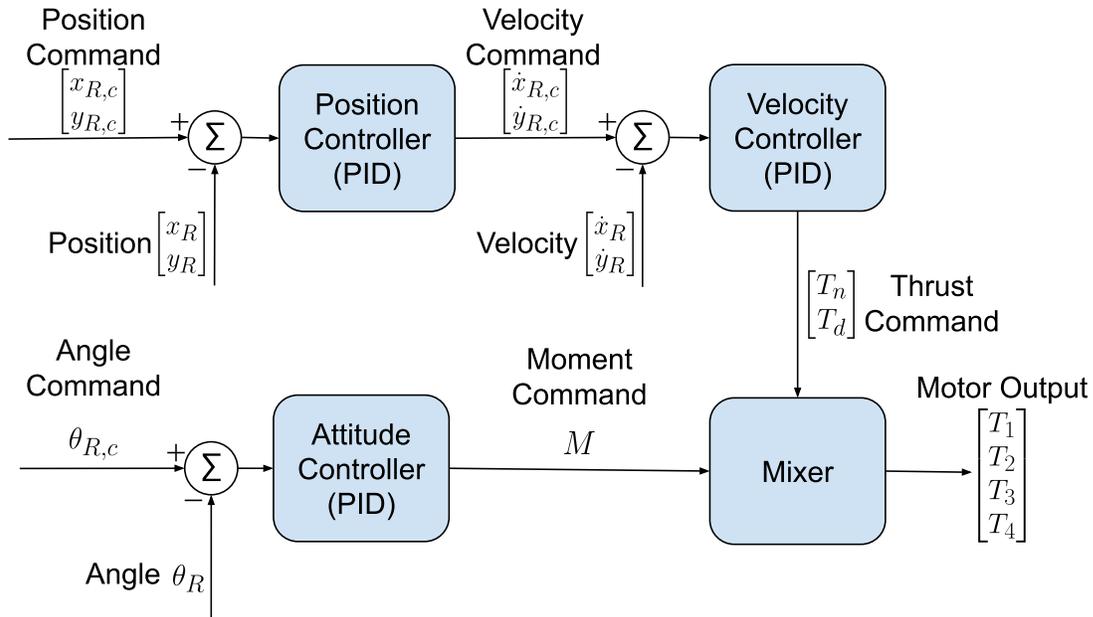


Figure 3.6: Cascaded PID controller that provides ROV motor thrust outputs from commanded position or velocity and attitude

As before, a given position command is compared to the actual position to determine the error, which error passes through the PID position controller to produce a velocity command. Either this velocity command or a direct velocity command can be used to calculate the velocity error for the PID velocity controller. Instead of outputting an acceleration command, the velocity controller directly produces the thrust command in each the north and down directions.

The decoupled attitude control allows any desired angle to be commanded. The error between the commanded and actual angles is the input to the PID attitude controller, which produces a moment command. The thrust commands and moment command are

through the mixer to determine the motor outputs:

$$\begin{bmatrix} T1 \\ T2 \\ T3 \\ T4 \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} & -\frac{1}{2d} & 0 \\ -\frac{1}{2} & \frac{1}{2d} & 0 \\ 0 & 0 & \frac{1}{2} \\ 0 & 0 & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} T_{down} \\ M \\ T_{north} \end{bmatrix} \quad (3.20)$$

An issue with using a PID controller on an attitude is that because the attitude coordinate is between $-\pi$ and π , the controller may attempt to reduce error by rotating more than 180 degrees. For example, if the current orientation is at $-3\pi/4$ and the desired is at $3\pi/4$, the error is $3\pi/2$, and the controller will output a moment that will rotate counter-clockwise over the entire error instead of rotating clockwise by $\pi/2$. To ensure that the ROV always rotates in the correct direction, Algorithm 1 is implemented.

Algorithm 1 Smallest Turn Logic

Input: Commanded angle θ_c , current angle θ

while $\theta_c - \theta > \pi$ **do**

$\theta_c \leftarrow \theta_c - 2\pi$

end while

while $\theta_c - \theta < -\pi$ **do**

$\theta_c \leftarrow \theta_c + 2\pi$

end while

Figure 3.7 shows simulated results of the ROV being commanded to follow some position and angle commands. Note that while the lateral and vertical positions are tracked, the angle is able to change arbitrarily.

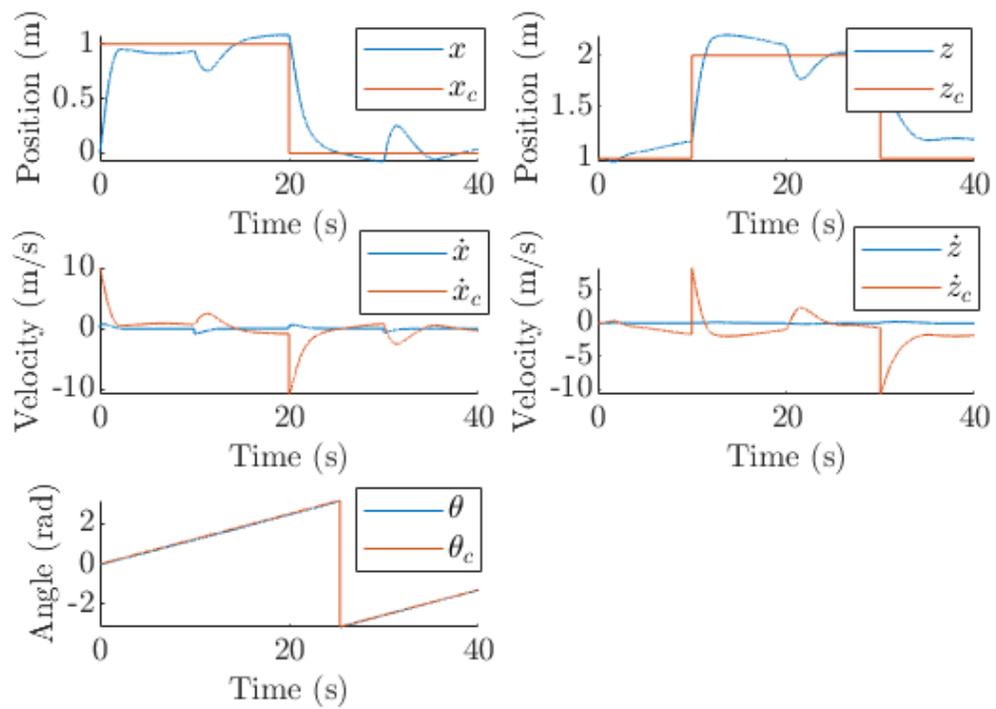


Figure 3.7: Simulation results of the ROV dynamics and controls

Chapter 4: Laser Measurement Modeling

Lasers typically have a very small divergence, such that they can travel long distances compared to typical lights (flashlights, headlights, etc.). This is because lasers are highly collimated, where all of the light rays exiting the laser device travel nearly parallelly to each other. However, in the application of bistatic laser imaging, a collimated laser beam would be nearly impossible to locate with a UAV flying above because of the small area that the laser illuminates in the horizontal plane of the laser receiver. This region of the horizontal receiver plane where the laser light is seen is heretofore called the laser-illuminated area (LIA; see Figure 4.1). To increase the ease of positioning the receiver within the LIA, the upwelling laser is decollimated using a lens or diffuser. By passing through the decollimating object, the laser divergence increases from nearly zero to an angle specified by the object's engineering.

The laser models examined here are designed to describe the spread of a laser after passing through a diverging apparatus. The distribution of a laser that passes through a decollimating lens can be approximated with a Gaussian beam model, while that of a laser passing through a diffuser can be approximated with a flat-top beam model. These models are developed because understanding them can offer information on how to track the ROV, and they provide a framework for simulating the system as well. For the development of

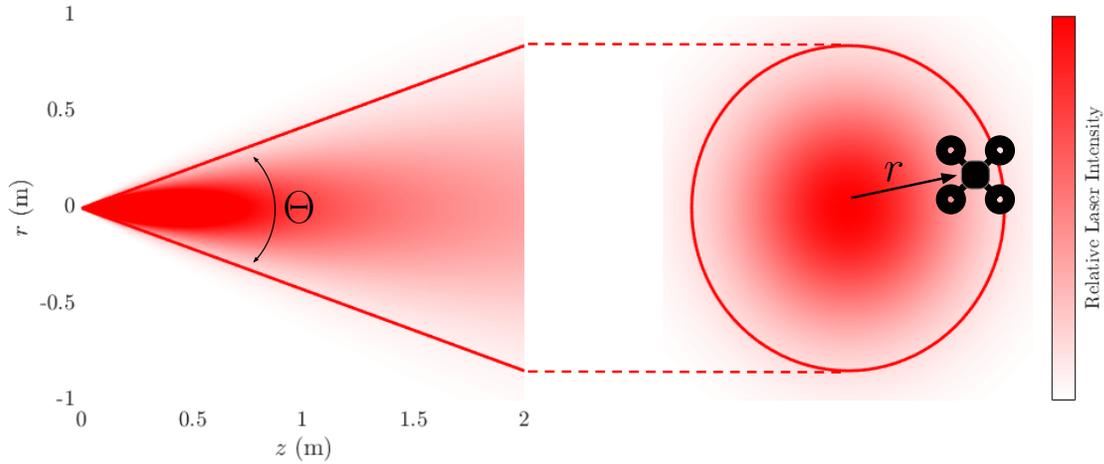


Figure 4.1: A laser profile from the side (left), and a cross-section of the profile showing the laser-illuminated area at $z = 2$ meters (circled, right); the divergence angle Θ and radial distance r are depicted

the laser intensity models, it is assumed that the UAV-mounted receiver is always level and pointing straight down, and that the ROV-mounted emitter is level and pointing straight up. Additionally, laser attenuation in the water and refraction at the air-water interface are neglected.

As with many physical phenomena, lasers follow the inverse-square law; that is,

$$\text{intensity} \propto \frac{1}{\text{distance}^2}. \quad (4.1)$$

This law can be explained by observing that as light travels, it spreads such that its area increases according to the divergence angle. Assuming that the laser power is the same at all distances, the intensity is equal to the power divided by the area, which area increases proportional to the square of the distance.

4.1 Laser Intensity Models

4.1.1 Gaussian Intensity Model

The Gaussian intensity model represents a laser beam whose intensity in the transverse plane is given by a Gaussian function [25]. While the model is more typically applied to lasers of small divergences, it is shown to apply as well to a beam with a wide divergence, such as one intentionally spread with a diverging lens.

The model is derived starting with a two-dimensional Gaussian function

$$f(x, y) = A \exp \left(- \left(\frac{(x - x_0)^2}{2\sigma_X^2} + \frac{(y - y_0)^2}{2\sigma_Y^2} \right) \right) \quad (4.2)$$

Assuming that (i) the mean is at zero ($x_0 = y_0 = 0$) and (ii) the variance is the same in each direction ($\sigma_X = \sigma_Y = \sigma$), and changing the coordinate to a radius ($r^2 = x^2 + y^2$), the equation simplifies to

$$f(r) = A \exp \left(- \frac{r^2}{2\sigma^2} \right). \quad (4.3)$$

The volume under the distribution is $2\pi A\sigma^2$, which is set equal to the power P of the laser, so A is solved for as

$$A = \frac{P}{2\pi\sigma^2}. \quad (4.4)$$

The intensity model can then be written as

$$I(r) = \frac{P}{2\pi\sigma^2} \exp \left(- \frac{r^2}{2\sigma^2} \right), \quad (4.5)$$

where I is the laser intensity and r is the radial distance from the laser receiver to the laser axis. However, σ varies with the axial distance from the emitter to the receiver (z). Instead of using σ , the $1/e^2$ width of the laser is generally used [26]. This width w , also called the beam waist, is the width where the laser intensity is 0.135 times the peak intensity, or

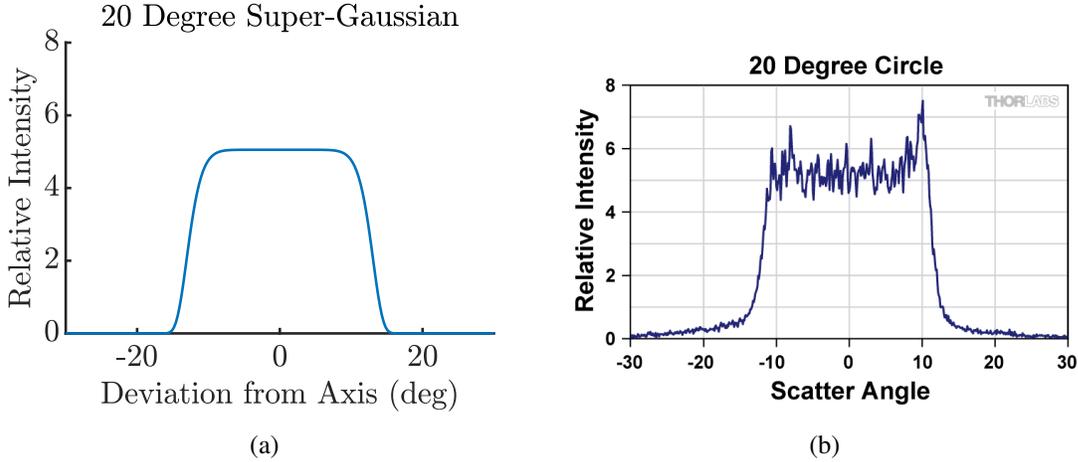


Figure 4.3: Comparison between super-Gaussian intensity model (a) and diffused laser intensity from ThorLabs Engineered Diffuser ⁵ (b)

distribution [27] (see Figure 4.3). A super-Gaussian function takes the following form:

$$f(r) = A \exp\left(-\frac{1}{2} \left(\frac{r}{\sigma}\right)^n\right), \quad (4.8)$$

where n is a positive, even integer indicating the order of the super-Gaussian. For example, a super-Gaussian with order $n = 2$ is equivalent to the standard Gaussian function. As the order of the super-Gaussian increases, the distribution grows closer to approximating a rectangular shape (see Figure 4.4).

4.1.2 Flat-top Intensity Model

The equation for intensity is then

$$I(r, z) = I_0(z) \exp\left(-\frac{1}{2} \left(\frac{r}{\sigma}\right)^n\right). \quad (4.9)$$

The on-axis intensity I_0 is different from before, but can be solved for in a similar manner.

The volume under the distribution is

$$V = \int_0^{2\pi} \int_0^{\infty} I(r, z) r dr d\theta = 2\pi I_0(z) \frac{1}{n} \left(4^{\frac{1}{n}}\right) \sigma^2 \Gamma\left(\frac{2}{n}\right), \quad (4.10)$$

⁵https://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=1660

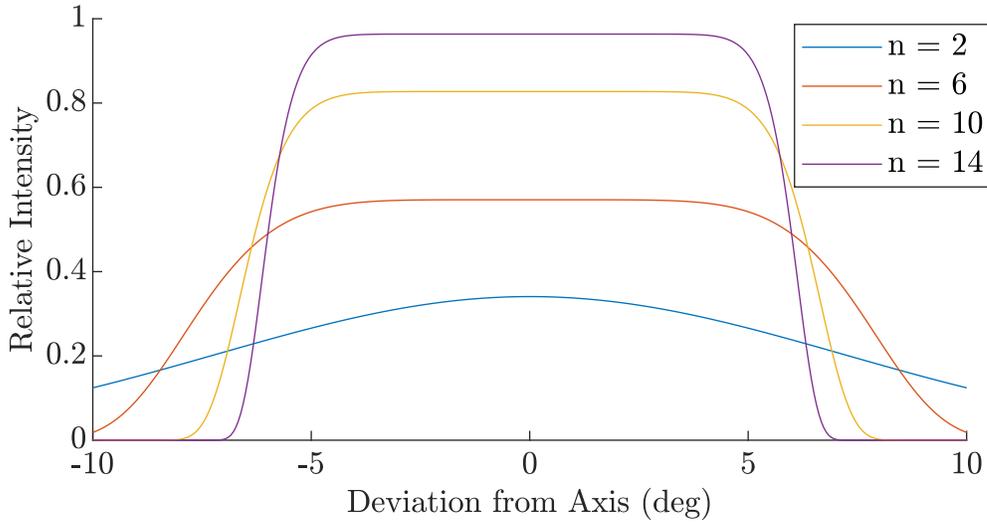


Figure 4.4: Gaussian laser model with increasing order (n); as the order increases, the transient region between minimum and maximum intensity shortens

where $\Gamma(\cdot)$ is the gamma function. Because n is positive and real, the following definition of the gamma function can be used:

$$\Gamma(z) = \int_0^{\infty} x^{z-1} e^{-x} dx. \quad (4.11)$$

Equation 4.10 calculates the volume under the two-dimensional symmetrical Gaussian, generalizing to any order n where n is a positive, even integer. Plugging in $n = 2$, the equation is equivalent to the previously stated volume under the standard two-dimensional Gaussian ($V = 2\pi A\sigma^2$). Then, the more general equation for the on-axis peak intensity (I_0) is found by setting the total volume to be the power P of the laser and solving for A :

$$I_0(z) = \frac{nP}{2\pi \left(4^{\frac{1}{n}}\right) \sigma^2 \Gamma\left(\frac{2}{n}\right)}. \quad (4.12)$$

There is another key difference between the Gaussian and flat-top laser models that must be accounted for here. As previously mentioned, the divergence of a Gaussian laser is the angle made between the waist lengths of the laser (see Figure 4.2). This waist is made to the $1/e^2$ width, where the intensity is at 0.135 times the peak value. However, the

divergence of flat-top lasers is the angle over which the intensity is approximately equal to the peak intensity (see Figure 4.3). As such, there must also be a modification made to the choice of standard deviation, which is a function of the width of the flat top of the laser.

Let the standard deviation σ be where the super-Gaussian laser intensity drops to 0.95 times the peak intensity:

$$0.95 = \exp\left(-\frac{1}{2}\left(\frac{r}{\sigma}\right)^n\right). \quad (4.13)$$

Choosing r as the radius to the edge of the divergence of the flat-top laser ($r = z \sin\left(\frac{\Theta}{2}\right)$) and solving for σ yields

$$\sigma = 0.1^{-\frac{1}{n}} z \sin\left(\frac{\Theta}{2}\right). \quad (4.14)$$

r is the radius to the edge of the divergence of the flat-top laser ($r = z \sin\left(\frac{\Theta}{2}\right)$).

Plugging Equation 4.12 and Equation 4.14 into Equation 4.9 yields the complete flat-top intensity model.

4.1.3 Determining Radial Distance and Axial Distance from Laser Source

The radial displacement from the laser mean r and the axial distance from the emitter to the receiver z must be determined to calculate the received laser intensity at any given location. Let a generic laser reference frame be defined as $\mathcal{L} = (L, \mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3)$, and the laser receiver frame be defined as $\mathcal{S} = (S, \mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3)$ (see Figure 4.5). Let $\mathbf{r}_{S/L}$ be the vector to the UAV-mounted receiver from the origin of the laser. The axial distance z is then equivalent to the scalar projection of $\mathbf{r}_{S/L}$ on the laser unit vector \mathbf{l}_1 :

$$z = \|\mathbf{r}_{S/L}\| \cos \theta = \mathbf{r}_{S/L} \cdot \mathbf{l}_1, \quad (4.15)$$

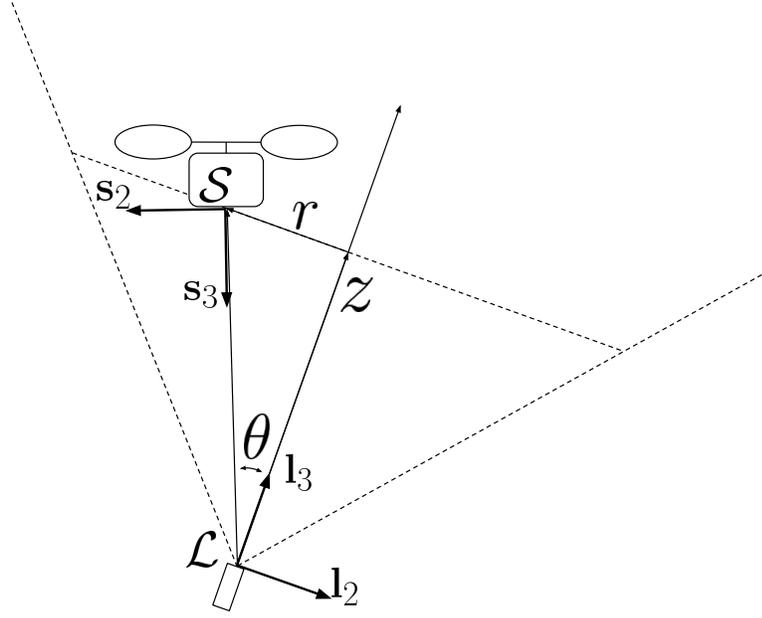


Figure 4.5: The receiver frame ($\mathcal{S} = (S, \mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3)$) and the laser frame ($\mathcal{L} = (L, \mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3)$) with respective unit vectors; note that the laser and receiver point along their third axis ($\mathbf{l}_3, \mathbf{s}_3$)

where θ is the angle between the vectors. Similarly, the radial displacement r is equal to the scalar rejection of $\mathbf{r}_{S/L}$ on \mathbf{l}_1 :

$$r = \|\mathbf{r}_{S/L}\| \sin \theta = \mathbf{r}_{S/L} \cdot (\mathbf{l}_1)^\perp = \mathbf{r}_{S/L} \cdot -\mathbf{l}_3. \quad (4.16)$$

In the case of the upwelling laser, if it is assumed that the UAV and ROV are both level, the calculations for z and r are simplified, where z is the laser depth added to the receiver altitude, and r is the radial distance from the laser to the receiver in the horizontal plane:

$$z = \mathbf{r}_{S/L} \cdot \mathbf{e}_3 = z^\mathcal{L} - z^\mathcal{S} \quad (4.17)$$

$$r = \sqrt{(\mathbf{r}_{S/L} \cdot \mathbf{e}_1)^2 + (\mathbf{r}_{S/L} \cdot \mathbf{e}_2)^2} = \sqrt{(x^\mathcal{L} - x^\mathcal{S})^2 + (y^\mathcal{L} - y^\mathcal{S})^2}. \quad (4.18)$$

4.2 Laser Detectability and Effective Divergence

The intensity model mentioned before is only applicable if the laser is detectable by the receiver. Detectability depends not only on the positions of the drone and ROV, but on their orientations relative to the laser divergence, the receiver FOV, and each other. For example, it is possible that the receiver is within the laser-illuminated area, without detecting any laser intensity, if it is not oriented toward the laser emitter (see Figure 4.6).

First, it is assumed that the receiver is level and facing down, with the laser level and facing up. Also, for the purposes of this project, it can be assumed that the laser divergence is greater than the receiver FOV, making the receiver FOV the limiting factor in laser detectability. If the receiver moves laterally such that the laser is no longer in the FOV, then the signal will not be detected, even if the receiver is in the laser-illuminated area. Within the laser-illuminated area, there is a restricted laser-detectable area (LDA), defined by the receiver FOV.

This limited detectability also affects the fidelity of the Gaussian and flat-top laser models, and must be taken into account. In theory, the effect of a limited receiver FOV would result in immediate loss of signal when the laser is no longer detectable. Instead of a Gaussian or super-Gaussian distribution, the model would have piecewise properties, cutting off the model at the receiver FOV (see Figure 4.7).

There are several disadvantages that arise from treating the measurement model this way. First, the discontinuity of the piecewise function prevents us from using tools that use a derivative of the model. Second, in the case of the flat-top laser, the model is degraded into merely a digital signal — the laser is either detected, or it is not, with no

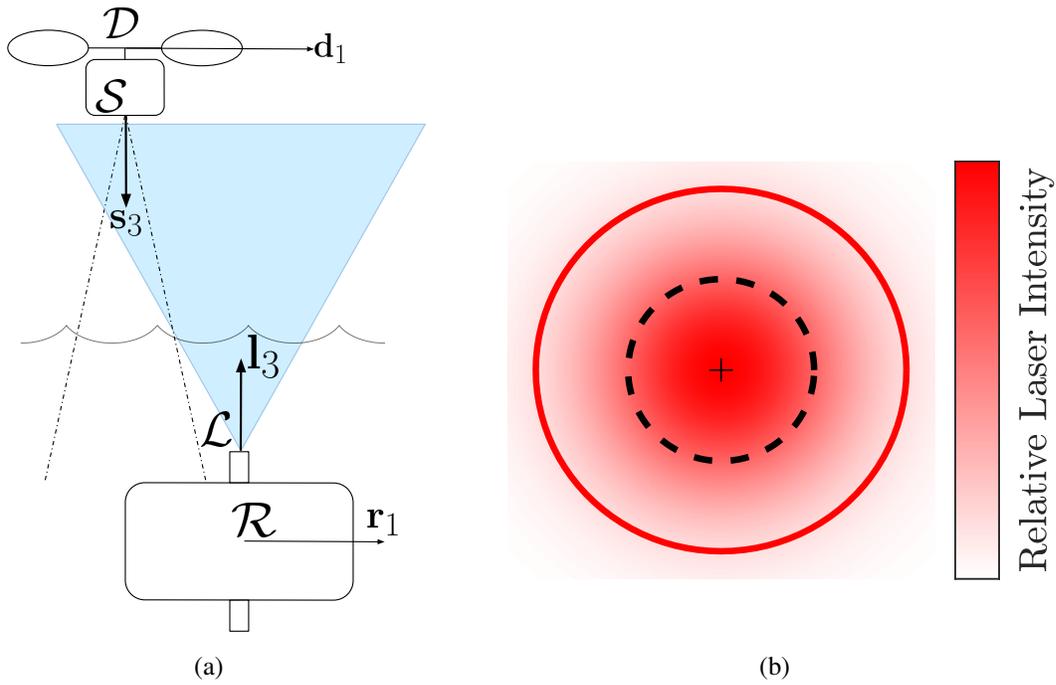


Figure 4.6: It is shown that the receiver can be in the LIA without detecting the laser (a); the relative size of the LDA (dotted) compared to the LIA (solid) (b)

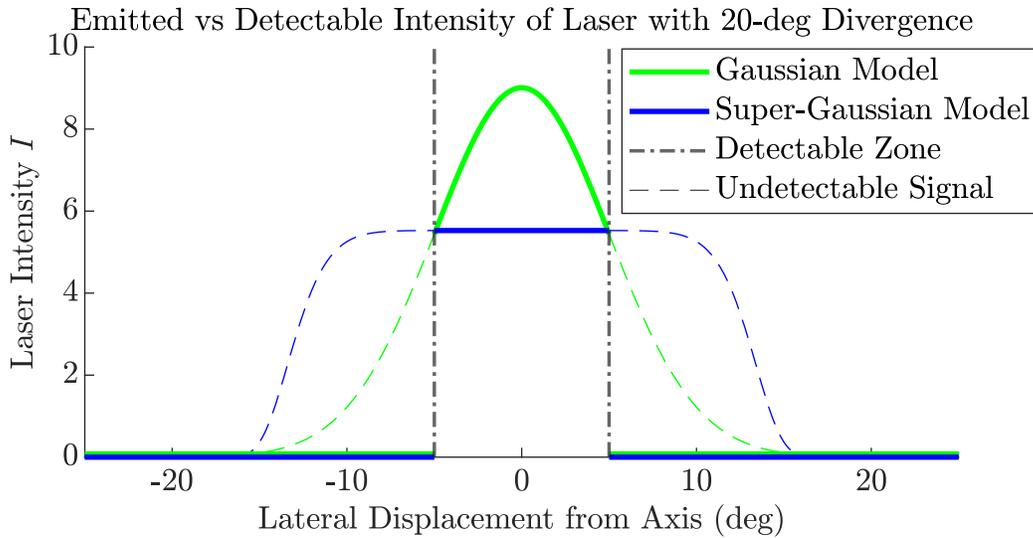


Figure 4.7: The unaltered laser models compared to the expected detectable models.

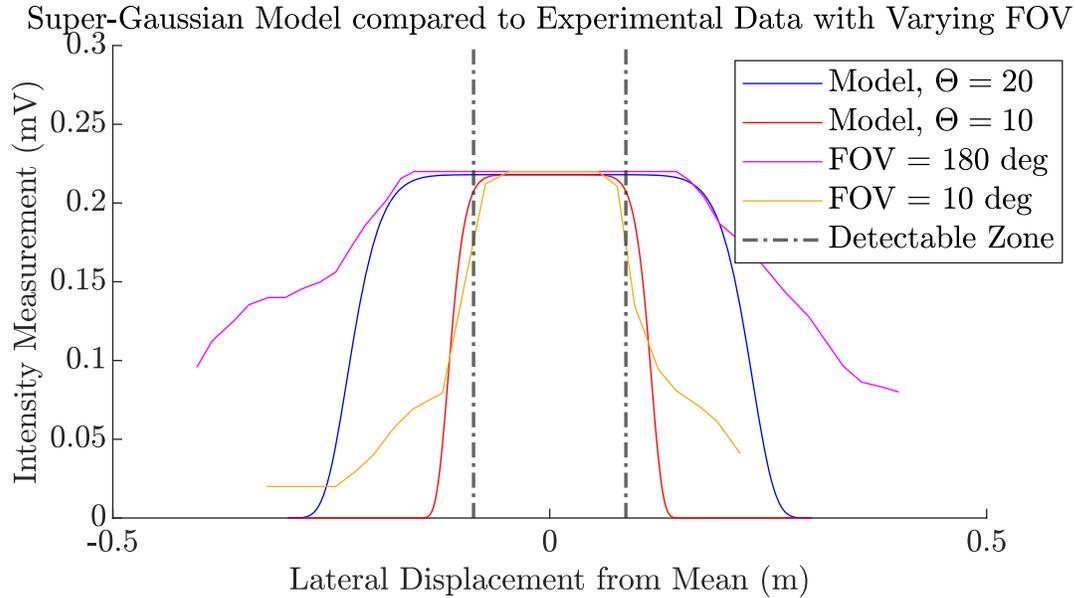


Figure 4.8: A comparison between the super-Gaussian models with experimental data of varying receiver FOV

more information being given. However, the following results demonstrate that there is more nuance to the effect that the limited FOV has on the laser intensity distribution.

Figure 4.8 shows the comparison between two plots of experimental data with two corresponding super-Gaussian model plots. Both sets of experimental data are gathered by passing the receiver across the LIA at constant altitude. The data labeled “FOV = 180 deg” are gathered by pointing the receiver at the laser emitter while the receiver moves laterally, simulating a full (180 deg) FOV. It is compared to the super-Gaussian model with a divergence of 20 deg, which matches the divergence of the flat-top laser used in the test. The data labeled “FOV = 10 deg” are gathered by maintaining level the receiver. It is compared to the super-Gaussian model with a divergence of 10 deg, which is equivalent to the FOV of the receiver.

It is apparent that at the edges of the receiver FOV, the measured intensity does not immediately drop to zero as the model would expect (see Figure 4.7). Instead, it has a

more gradual gradient on either side. Instead of using the piecewise model previously shown, this intensity distribution is better approximated by a super-Gaussian, but using the receiver FOV as the divergence Θ . This choice of divergence in the model is referred to as the system's effective divergence in this project, and is used instead of the piecewise model.

Chapter 5: Estimation and Guidance Algorithms

The objective of the UAV autonomy is to maximize the percent of time that the laser receiver maintains reception of the laser signal. When they are thus connected, the laser imaging data can be taken and worked into 3D images. The following guidance and estimation methods were developed to meet that objective. The measurements and estimation techniques for the vehicles are specified, then laser estimation methods are introduced. Guidance algorithms are developed with these estimation methods in mind.

5.1 Vehicle State Estimation

5.1.1 Unmanned Aerial Vehicle

The UAV uses various sensors to determine its position and orientation. The IMU provides linear acceleration and angular rate measurements. When flying outdoors, a GPS module provides position and velocity measurements. When flying indoors, the tracking camera uses stereo cameras and visual-inertial odometry (VIO) to produce position and velocity measurements. The UAV combines either the GPS measurements or the VIO estimate with the on-board IMU data in an error-state EKF estimator. This estimator compares measurements with the dynamic model of the UAV to provide an improved

state estimate. Having a position and velocity estimate is important not just for the UAV to fly autonomously, but also to provide measurements to the later estimation and guidance algorithms.

5.1.2 Remotely Operated Vehicle

Onboard the ROV is an IMU that is used to measure the ROV orientation. It also has a built-in barometer that gives a depth measurement. These measurements are sufficient to remotely operate the ROV in assisted control modes, such as a stabilized depth-hold mode that frees the operator from having to manually maintain the orientation and depth of the ROV.

To provide a position estimate, ROVs in general often utilize acoustic based measurement systems, such as the short baseline (SBL) system with which this project's ROV is equipped. The systems rely on acoustic communications between a beacon on the ROV and multiple transponders at known locations. The distance between the sensors is calculated, allowing the ROV position to be triangulated with one of various filtering methods. Unfortunately, acoustic positioning systems can struggle in pools and watertanks due to multipath reflections [28], as with the test tank for this project.

If a position estimate is unavailable, other techniques can be used to estimate the vehicle velocity. A doppler velocity log (DVL) also uses acoustics, but with a single unit attached to the ROV. It faces toward the sea/pool floor, using doppler shift to estimate the velocity of the ROV. Another velocity estimation technique that may be available to some ROVs is optical flow. If the lighting and clarity conditions are sufficient, a computer

vision program can track the movement or "flow" of features from a camera, estimating the velocity of the ROV [29]. While this project's ROV is not equipped with a velocity estimator, some of the upcoming algorithms assume that the ROV velocity is known.

5.2 Laser Intensity Estimation

The laser intensity at any point in the LIA is measured by the laser receiver, consisting of a photodiode and source transformation circuit. From the laser intensity measurements, the objective is to estimate the position of the centerpoint of the LIA and LDA. The centerpoint is the ideal location for data collection because (i) the intensity is at a peak, giving the maximum distinction between signal and noise, and (ii) there is an equal buffer in all directions around the point to prevent small disturbances from causing signal loss. Two approaches are described to estimate the position of the peak signal intensity.

5.2.1 Gradient Ascent

Gradient ascent is an optimization algorithm that seeks to find the maximum of a function in a design space. In this case, the function to be maximized is the laser intensity. The spread of the laser intensity over the LIA creates a spatial gradient, in which each position of the receiver corresponds to a gradient in a physical quantity. The spatial gradient of this work's laser intensity models is seen in the sloped areas of the models (see Figure 4.4). The Gaussian beam model has a non-zero gradient anywhere in the LIA except at the peak, while the flat-top intensity model only has a non-zero gradient in the sloped edges of the distribution. Even without having a complete understanding of

the laser model, following the gradient upward would drive the UAV toward the position of peak laser intensity, whether that be at the center of the LIA for the Gaussian beam, or to the edge of the peak region for the flat-top model.

This application of gradient ascent involves two steps: estimation of the local gradient and commanding the UAV to follow the gradient. The gradient at the location of the receiver is calculated by numerically calculating the derivative of the laser intensity I in the x and y directions, i.e.

$$\nabla I = \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix}, \quad (5.1)$$

where the partial derivatives are found numerically as:

$$\frac{\partial I}{\partial z} = \frac{I_k - I_{k-1}}{z_k - z_{k-1}} \quad (5.2)$$

at the k^{th} timestep for a given coordinate z .

There are two challenges this approach has in practice. First, if the change in the position coordinate is extremely small between timesteps, the derivative may be too large or undefined. Second, noise in the intensity signal can cause inconsistency in the derivative value, making the gradient estimate jumpy and inaccurate. To combat these problems, the following measures were implemented. First, the timestep between derivative evaluations is chosen such that the current coordinates differ from the previous by a minimum quantity. Second, the calculated derivatives are smoothed by a low-pass filter. Finally, if the first condition is not met, the estimated gradient slowly decays to 0. This allows the zero-gradient at the Gaussian intensity peak to be estimated, even when the UAV is no longer moving. The complete algorithm is found in Algorithm 2.

This first step estimates the local gradient of the laser. The second step, commanding

Algorithm 2 Gradient Estimation Algorithm

```
1: Inputs:  
   Intensity Measurement  $I_k$ , current position  $\mathbf{x}_k = [x_k \ y_k]^\top$   
2: Initialize:  
    $\mathbf{x}_{k-1} \leftarrow \mathbf{x}_k$   
    $I_{k-1} \leftarrow I_k$   
    $\nabla I_{k-1} \leftarrow \nabla I_k$   
3: if  $\|\mathbf{x}_k - \mathbf{x}_{k-1}\| \geq d_{min}$  then ▷ Only calculates after minimum movement  
4:    $\nabla I_k \leftarrow (I_k - I_{k-1}) / (\mathbf{x}_k - \mathbf{x}_{k-1})$   
5:    $\nabla I_k \leftarrow \beta \nabla I_k + (1 - \beta) \nabla I_{k-1}$  ▷ Low-pass filter to smooth gradient  
6: else  
7:    $\nabla I_k \leftarrow (1 - \gamma) \nabla I_{k-1}$  ▷ Decays  $\nabla I_k$  to 0  
8: end if  
9: return  $\nabla I_k$ 
```

the UAV to follow the gradient, provides a means to estimate the location of the laser axis through guiding the UAV. It is discussed in the Guidance section.

5.2.2 Extended Kalman Filter

An extended Kalman filter (EKF) is a tool in estimation that uses measurements and a dynamic model to provide estimates of unknown states. Unlike the Kalman filter, the EKF can use nonlinear measurement and state transition models, linearizing them at the estimated state. In this case, an EKF is designed to estimate the position of the intersection of the laser axis with the UAV horizontal plane, i.e., the centerpoint of the LIA.

The EKF algorithm has two main steps: state propagation and measurement update. The former uses the given dynamic model to estimate how the states will evolve based on previous states and model inputs, while the latter uses measurements to improve the estimate. While the states are propagated and updated, the covariance for the states is also propagated and updated, and plays an essential part in the update step.

The EKF designed for this application only considers movement in the horizontal plane of the UAV, as it deals with estimating the centerpoint of the LIA at a given altitude. It is assumed that the altitude is fixed, such that the laser measurements vary only with a change in position in the horizontal plane, i.e. the axial distance coordinate z is constant. As such, the radial distance r is the coordinate used by the EKF to perform laser intensity measurement updates.

The relative position of the LIA centerpoint (denoted P) to the receiver ($\mathbf{r}_{P/S}$) is represented with polar coordinates r and θ (see Figure 5.1). Additionally, the relative velocity is considered here, calculated as

$$\mathcal{I}\mathbf{v}_{P/S} = \mathcal{I}\mathbf{v}_{P/I} - \mathcal{I}\mathbf{v}_{S/I}. \quad (5.3)$$

With the assumptions that both the ROV and UAV are operating at level conditions, the velocities of the laser emitter and receiver will be equal to the ROV and UAV velocities, respectively. The velocity of the UAV is known from available sensor measurements. For this approach, the velocity of the ROV must either be estimated, or approximately equal to zero.

The state vector used in the EKF is $\mathbf{x} = [r \ \theta \ \dot{r} \ \dot{\theta}]^\top$, where x and y are the relative position coordinates established previously, and \dot{r} and $\dot{\theta}$ are the polar velocity coordinates. The dynamic models used are linear, kinematic equations, assuming no change in the velocity coordinates. As such, the dynamic models are expressed as

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{r} \\ \dot{\theta} \\ \ddot{r} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{x} = \mathbf{A}\mathbf{x}, \quad (5.4)$$

where \mathbf{A} is the state dynamics matrix.

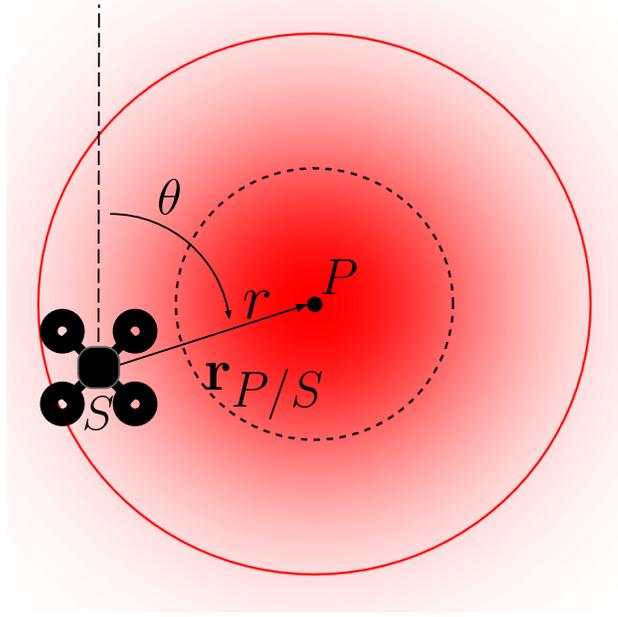


Figure 5.1: Position coordinates r and θ of the LIA centerpoint P relative to the UAV-mounted receiver S

The discrete-time state vector is $\hat{\mathbf{x}}_k = [r_k \ \theta_k \ \dot{r}_k \ \dot{\theta}_k]^\top$ at the k^{th} timestep. The discrete-time state dynamics matrix is then:

$$\hat{\mathbf{x}}_k = \begin{bmatrix} 1 & 0 & \Delta t_k & 0 \\ 0 & 1 & 0 & \Delta t_k \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \hat{\mathbf{x}}_{k-1} = \mathbf{A}_k \hat{\mathbf{x}}_{k-1}, \quad (5.5)$$

where Δt_k is the time interval between the $k - 1^{\text{th}}$ and k^{th} timesteps.

The discrete-time measurement vector is $\mathbf{z}_k = [I_k \ v_{x,k} \ v_{y,k}]^\top$ at the k^{th} timestep, where I_k is the laser intensity measurement and $v_{x,k}$ and $v_{y,k}$ are the relative velocity measurements in the inertial frame \mathbf{e}_1 and \mathbf{e}_2 directions, respectively. The measurement model for laser intensity is from Equation 4.9:

$$h_{I,k}(\mathbf{x}) = I_0 \exp\left(-\frac{1}{2} \left(\frac{r_k}{\sigma}\right)^n\right). \quad (5.6)$$

The models for the velocity measurements are calculated by taking the inertial derivatives of the polar position:

$$h_{v_{x,k}}(\mathbf{x}) = \dot{r} \cos(\theta) - r \dot{\theta} \sin(\theta) \quad (5.7)$$

$$h_{vy,k}(\mathbf{x}) = \dot{r} \sin(\theta) + r\dot{\theta} \cos(\theta). \quad (5.8)$$

Then the complete measurement equation is

$$h_k(\mathbf{x}) = \begin{bmatrix} h_{I,k}(\mathbf{x}) \\ h_{vx,k}(\mathbf{x}) \\ h_{vy,k}(\mathbf{x}) \end{bmatrix}. \quad (5.9)$$

The Jacobian of the measurement equation is taken to produce a linearized output matrix \mathbf{C}_k :

$$\mathbf{C}_k = \begin{bmatrix} c \exp\left(-\frac{1}{2} \left(\frac{r}{\sigma(z)}\right)^n\right) & 0 & 0 & 0 \\ -\dot{\theta} \sin(\theta) & -\dot{r} \sin(\theta) - r\dot{\theta} \cos(\theta) & \cos(\theta) & -r \sin(\theta) \\ \dot{\theta} \cos(\theta) & \dot{r} \cos(\theta) - r\dot{\theta} \sin(\theta) & \sin(\theta) & r \cos(\theta) \end{bmatrix}_{\mathbf{x}_k|k-1} \quad (5.10)$$

with

$$c = \frac{-n^2 r^{n-1} P}{4^{1/n+1} \pi \sigma(z)^{n+2} \Gamma(2/n)} \quad (5.11)$$

Note that the notation $\hat{\mathbf{x}}_{n|m}$ represents the estimate of \mathbf{x} at the timestep n given observations up to and including the timestep m . The steps of the EKF are then as follows:

1. Prediction Phase: The predicted state estimate $\hat{\mathbf{x}}_{k|k-1}$ is calculated with the discrete state dynamics matrix \mathbf{A}_k :

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{A}_k \hat{\mathbf{x}}_{k-1|k-1}. \quad (5.12)$$

The predicted estimate covariance $\mathbf{P}_{k|k-1}$ is also calculated with \mathbf{A}_k , increased by the process covariance matrix \mathbf{Q}_k :

$$\mathbf{P}_{k|k-1} = \mathbf{A}_k \mathbf{P}_{k-1|k-1} \mathbf{A}_k^\top + \mathbf{Q}_k. \quad (5.13)$$

2. Residual Calculations: The pre-fit residual $\tilde{\mathbf{y}}_k$ is calculated with the measurement model $h(\mathbf{x})$, evaluated at the predicted state $\hat{\mathbf{x}}_{k|k-1}$:

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1}). \quad (5.14)$$

The the pre-fit residual covariance \mathbf{S}_k is calculated with the Jacobian of the measurement model, \mathbf{C}_k , increased by the measurement covariance matrix \mathbf{R}_k :

$$\mathbf{S}_k = \mathbf{C}_k \mathbf{P}_{k|k-1} \mathbf{C}_k^\top + \mathbf{R}_k. \quad (5.15)$$

3. Update Phase: The Kalman gain \mathbf{K}_k is calculated, then used to compute the updated state estimate and covariance:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{C}_k^\top \mathbf{S}_k^{-1}. \quad (5.16)$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k. \quad (5.17)$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{C}_k) \mathbf{P}_{k|k-1}. \quad (5.18)$$

This EKF implementation combines the use of a linear dynamical model with a nonlinear measurement model; other than that, it follows Related to the previous mentions of laser detectability is the observability of the system. Although the laser measurement model is non-linear, the linearized output matrix \mathbf{C}_k is used in the discrete observability matrix [30], which is as follows:

$$\mathcal{O}_k = \begin{bmatrix} \mathbf{C}_k \\ \mathbf{C}_k \mathbf{A}_k \\ \mathbf{C}_k \mathbf{A}_k^2 \\ \mathbf{C}_k \mathbf{A}_k^3 \end{bmatrix}. \quad (5.19)$$

If the observability matrix \mathcal{O}_k has a full rank of 4, then the system is observable, such that the state estimation error will be exponentially bounded under proper conditions [30].

It is found that for the system to be observable, then both (i) the partial derivative of either velocity measurement with respect to the relative angle θ (i.e. $\partial v_{x/y,k} / \partial \theta$) and (ii) the partial derivative of the laser intensity with respect to the relative radial position r ($\partial I / \partial r$) must be non-zero. The first condition is met anytime that the LIA centerpoint P is in horizontal motion with respect to the laser receiver, a behavior known as persistent excitation. For the Gaussian beam, the second condition is satisfied whenever the laser

is detected, except at the single point where the receiver is directly on the laser axis. However, the flat-top laser model has zero-slope across the width of the laser divergence. This implies that the system can only be observable when the receiver is in the sloped edges of the laser distribution, an interesting aspect given that the objective of the system is to stay in the flat-top region.

In Section 4.2, detectability and effective divergence are mentioned. It is noteworthy that according to the theoretical piecewise laser model with a receiver with limited FOV (see Figure 4.7), then there would be no detectable region of a flat-top laser. This is because at any given point of the piecewise model, the partial derivative of the laser intensity with respect to the relative radial position ($\partial I/\partial r$) would be equal to zero, and thus the observability condition would not be met. However, because of the experimentally-demonstrated concept of effective divergence, the laser intensity model has regions of observability at the gradients on the perimeter of the flat-top area.

5.3 Autonomous Quadrotor Guidance

The UAV guidance algorithms are divided into two phases: the Search Phase and the Data Collection Phase. The former discusses algorithms to guide the UAV to acquire the laser signal. The latter proposes approaches to remain maintain within the LDA.

5.3.1 Search Phase

5.3.1.1 Concentric Search Pattern

A search algorithm was developed to find a point on the laser-illuminated area (Algorithm 3). The algorithm starts the search at a given position, i.e. the expected location of the laser-illuminated area, then begins searching by flying in a succession of expanding concentric circles. To maintain the flight stability of the UAV, the velocity of the trajectory along the circle is maintained constant by reducing the angular velocity as the radius increases. Unfortunately, the search will tend to stop the UAV on the edge of the laser-illuminated area, meaning that a small movement of the ROV can cause the signal to be lost.

Algorithm 3 Concentric Search Algorithm

```
1: Inputs:  
   Search area centerpoint  $\mathbf{x}_{center} = [x_{center} \ y_{center}]^\top$   
   Spacing between circles  $s$   
   Current time  $t_k$   
2: Initialize:  
    $r \leftarrow s$   
    $t_{start} \leftarrow t_k$   
3: if  $I_k < I_{min}$  then  
4:    $\omega \leftarrow v_{max}/r$   
5:    $t = t_k - t_{start}$   
6:   if  $\omega t < 2\pi$  then  
7:      $x_c \leftarrow r \sin(\omega t) + x_{center}$   
8:      $y_c \leftarrow r \cos(\omega t) + y_{center}$   
9:      $\mathbf{x}_c = [x_c \ y_c]^\top$   
10:  else  
11:     $t_{start} = t_k$   
12:     $r \leftarrow r + s$   
13:  end if  
14: end if
```

5.3.1.2 ROV Position Estimate

Because the ROV is assumed to remain level and the laser is assumed to point directly upward from the ROV, then if the position of the ROV relative to the UAV is known, then the UAV should be able to find the LDA by simply positioning itself directly above the ROV. Even if there is some error in the ROV position estimate, as long as it is less than the radius of the LDA, then this search method will be effective. If the ROV position estimate error is greater than the radius of the LDA, the estimate would still be a good, informed location to begin a concentric circle search pattern. This analysis also holds if the assumption that the laser is pointing directly upward is inaccurate; at least the best guess for a starting search location would be provided.

5.3.2 Tracking Phase

5.3.2.1 Hold Position

The simplest behavior to use once the LDA has been found is to hold the UAVs position. Theoretically, if the UAV and the ROV both stay still, then the receiver will remain in a position to perform data collection. However, if the UAV or ROV drift, then the search pattern would need to be executed again. For example, if the ROV is in constant motion, then the UAV will only be able to hold position for a brief time before having to perform another search.

5.3.2.2 ROV Tracking

As previously mentioned, if the ROV position estimate error is small, then the UAV should be able to track the ROV's position in the horizontal plane and thus remain in the LDA. As the ROV moves, its position can be sent directly to the UAV as a command, and the UAV controller places it in the desired location. This method would maximize exploitation of the laser signal, allowing the bistatic laser imaging procedure to achieve maximum effectiveness. However, if the model assumptions fail or if there is significant error in the ROV position estimate, then this method cannot guarantee any convergence to the LDA.

5.3.2.3 Tracking Laser Axis

The principal issue with the ROV tracking approach is that the UAV guidance ignores the laser intensity measurements. The following guidance methods use the estimates of the centerpoint of the LDA to control the UAV. These approaches are advantageous because they directly guide the UAV to the ideal position for data collection, instead of just following the ROV.

The gradient ascent method works by estimating the centerpoint as the UAV climbs the estimated gradient. The guidance algorithm (Algorithm 4) calculates a velocity command that is proportional to the gradient:

$$\dot{\mathbf{x}}_{c,k} = \alpha \nabla I_k, \quad (5.20)$$

where α is a constant. The velocity command is clipped at a maximum speed, then sent to the UAV controller.

Algorithm 4 Gradient Ascent Guidance Algorithm

1: **Inputs:**

$$\nabla I_k$$

2: $\dot{\mathbf{x}}_{c,k} \leftarrow \alpha \nabla I_k$

3: **if** $\|\dot{\mathbf{x}}_{c,k}\| \geq v_{max}$ **then** ▷ Clip the command if the magnitude exceeds v_{max}

4: $\dot{\mathbf{x}}_{c,k} \leftarrow v_{max} \dot{\mathbf{x}}_{c,k} / \|\dot{\mathbf{x}}_{c,k}\|$

5: **end if**

If the gradient is estimated well, then the UAV is always driven toward the peak of the laser intensity. For a Gaussian-distributed laser, this peak is found at the centerpoint. However, for a flat-top distributed laser, the peak intensity covers the entire divergence of the laser. The only gradient to climb is in the steep edge of the intensity model, so the gradient ascent guidance algorithm's effectiveness is limited in this case.

Tracking the relative centerpoint estimate provided by the EKF overcomes this issue by allowing the UAV to enter the zero-gradient area of the flat-top distribution. If the estimate is accurate, then the receiver will be placed in an ideal position to maintain maximum signal strength and buffer from disturbances. With the ROV's velocity estimate, the UAV is able to track the centerpoint without losing signal quality. Although the estimate covariance increases when the receiver is in the flat-top region of a laser, if the estimate error increases enough, then the measurements from the edge of the laser distribution will correct the estimate again.

Chapter 6: Simulation and Experimental Results

6.1 Simulation Environment

As previously mentioned, the reduced-order simulation only represents the system in two dimensions. The 2D dynamic models and controllers developed in Chapter 3 represent the hardware quadrotor UAV and ROV. Both the simulated UAV and ROV are assumed to have access to their full states. The UAV model control inputs are either position or velocity commands, similar to the hardware control inputs. The simulated ROV is also able to be commanded with a reference position or velocity and orientation; this varies from the hardware, where the ROV is manually piloted because there is not an accurate position estimate.

The upwelling laser is attached to the top of the simulated ROV. The laser can either be represented with the Gaussian intensity model, or the flat-top intensity model. In both cases, the divergence angle would be chosen to be 20 degrees, but because the receiver FOV is limited to 10 degrees, both laser models instead use an effective divergence of 10 degrees. The flat-top intensity model uses an order of $n = 10$.

Zero-mean Gaussian noise is added to the intensity measurements, as well as to the relative position and velocity measurements, in order to better approximate actual measurement data.

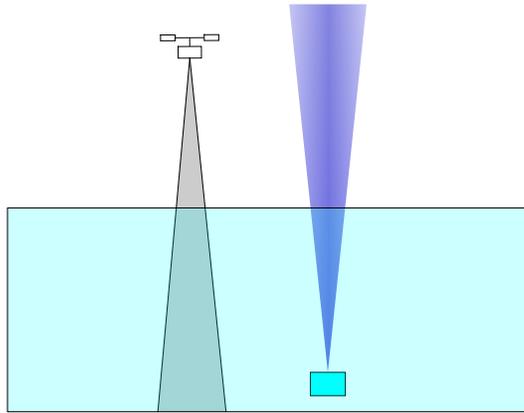


Figure 6.1: Animated UAV, ROV, laser, and receiver FOV from the simulation

6.2 Simulation Results

The simulation results are divided into two categories: the results from individual guidance and estimation algorithms highlighting their important aspects, and the combined results showcasing more realistic situations where the UAV autonomy must explore both the search phase and the tracking phase.

6.2.1 Individual Algorithm Results

6.2.1.1 Concentric Search

Figure 6.2 demonstrates the concentric search algorithm. The UAV initially flies to an estimated position of the LIA. When the laser intensity measurement is insignificant, the search begins. Because horizontal motion of the simulation is restricted to the x axis, the circling behavior appears like an oscillation of increasing amplitude. Also, as the angular velocity of the UAV decreases to keep the lateral velocity constant, the frequency

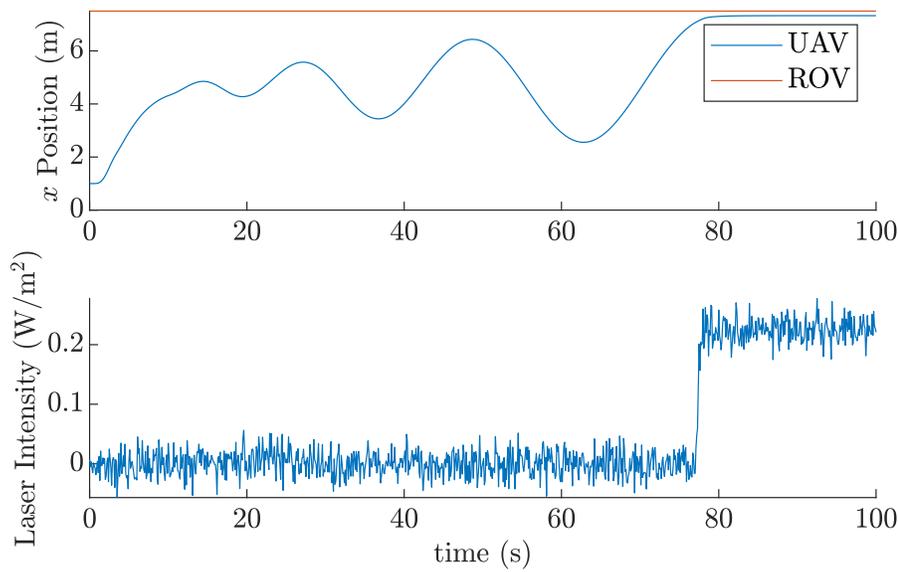


Figure 6.2: Simulation results of the concentric search algorithm

of the oscillation decreases as the amplitude increases.

6.2.1.2 Gradient Ascent

Figure 6.3 shows results of the gradient ascent algorithm for both a Gaussian intensity model and a flat-top intensity model. In both cases, the UAV begins near the edge of the LDA where the laser intensity is relatively low. As the gradient is estimated, the UAV is given a velocity command to climb the gradient, driving it toward the laser axis. Note that the results for the Gaussian intensity model show the UAV getting closer to the laser axis (directly above the ROV) than the results for the flat-top intensity model. This is because in the latter situation, the UAV is commanded to stop at the edge of the laser distribution's flat top, instead of being guided to the centerpoint.

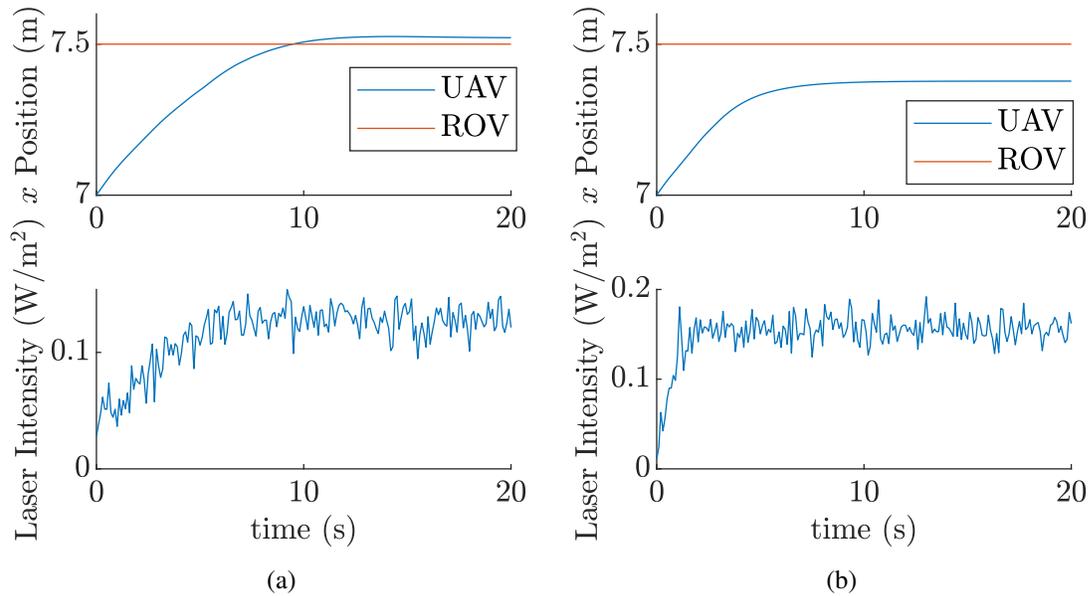


Figure 6.3: Simulation results using the gradient ascent algorithms for (a) a Gaussian laser model and (b) a super-Gaussian laser model

6.2.1.3 Extended Kalman Filter

The following results demonstrate some of the behaviors of the extended kalman filter due to the regions of unobservability in the flat-top intensity model. Figure 6.4 shows how the EKF reacts to the UAV repeatedly passing over the flat-top region. When the laser intensity is at a peak, the UAV is in that unobservable region, so the position covariance increases. However, when the receiver is positioned in the gradient areas of the laser intensity (where the intensity is changing), the covariance is reduced because the system is again observable. The objectives of maintaining peak laser intensity and minimizing error in the estimation end up competing with each other; to minimize the estimation covariance, the EKF seems to need measurements outside the flat-top peak region. However, the following results show that this may not be a consequential issue.

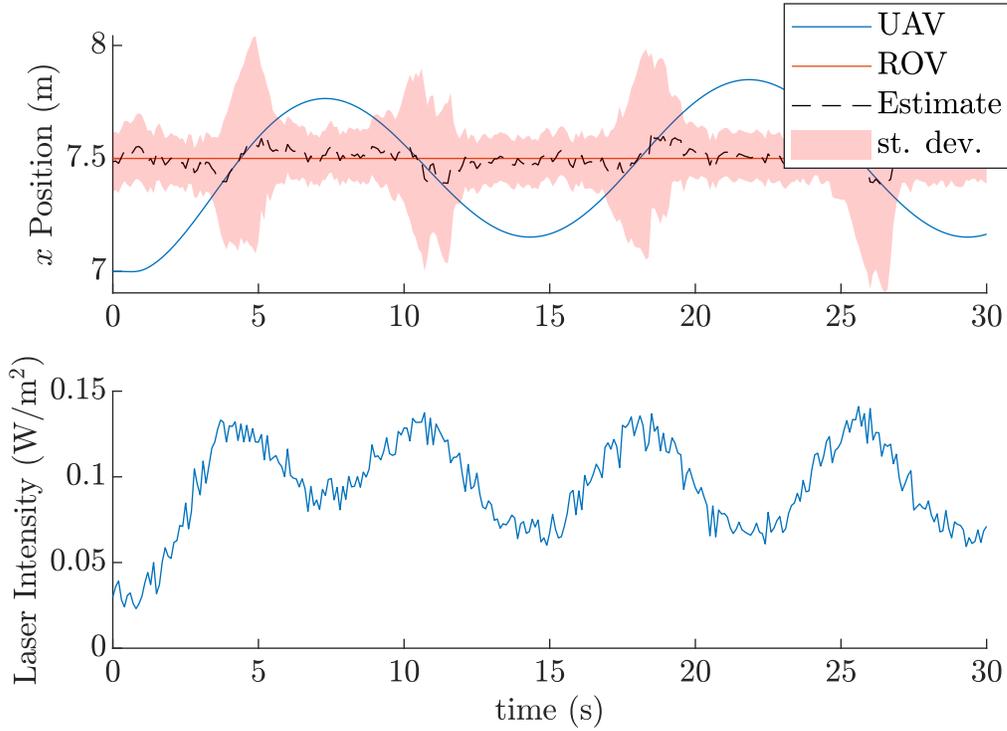


Figure 6.4: Simulation results of the EKF on a flat-top laser

Figure 6.5 shows the situation where the UAV is commanded to track the estimated position of the laser axis. In this case, the receiver remains in the flat-top area of the laser model, so the system remains unobservable. The covariance increases, although it appears to reach a maximum because the position estimate occasionally deviates into the gradient area. The measurement update then corrects the estimated position to return to the flat region, decreasing the covariance as well.

6.2.2 Combined Autonomy Results

In the following results, the simulations include an algorithm from both the search phase and the tracking phase. To draw closer to the experimental setup, the UAV initial position is at ground level several meters away from the location of the ROV. Some results

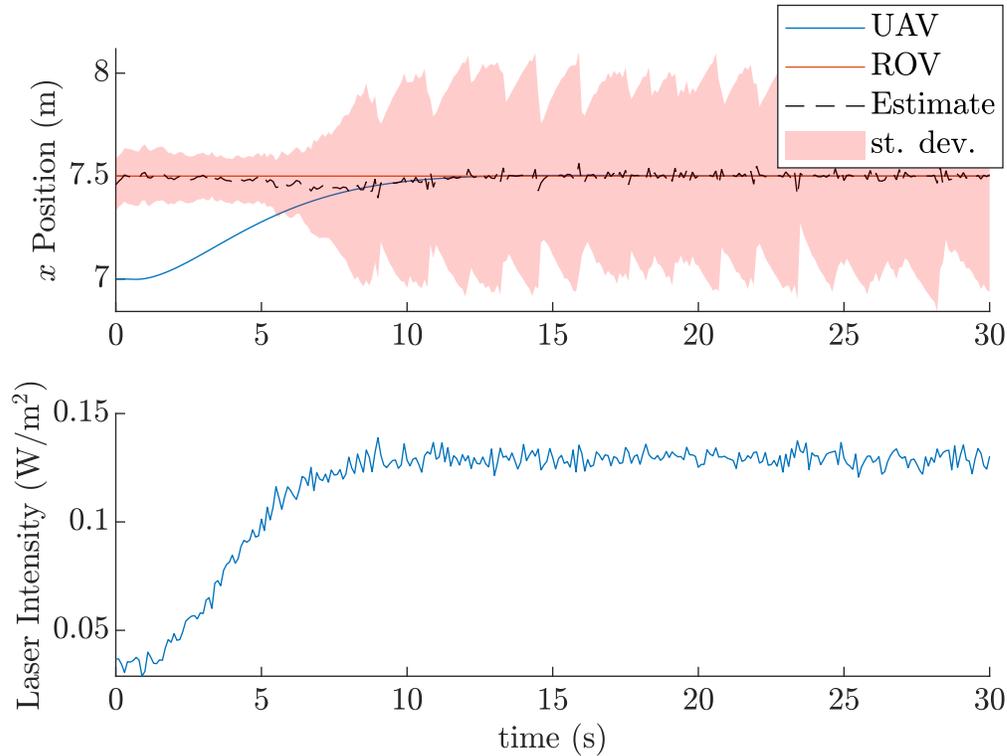


Figure 6.5: Simulated results of the EKF when the receiver remains in the flat-top area show the influence of a moving ROV on the autonomy’s effectiveness. In all cases, the UAV first flies to an initial guess of the location of the LDA centerpoint, which guess is given an error. The UAV then enters the search phase or the tracking phase, depending on if it receives a significant laser intensity measurement.

6.2.2.1 Search Pattern with Hold Position and Moving ROV

Figure 6.6 shows the results of using the concentric search pattern to find the LDA, then just holding position, similar to Figure 6.2. However, in this case, the ROV has a constant non-zero velocity. When the receiver no longer measures a significant laser intensity, the search phase begins again. The direction of motion of the ROV was chosen to be opposite the initial motion of the search pattern to show a worst-case scenario, and

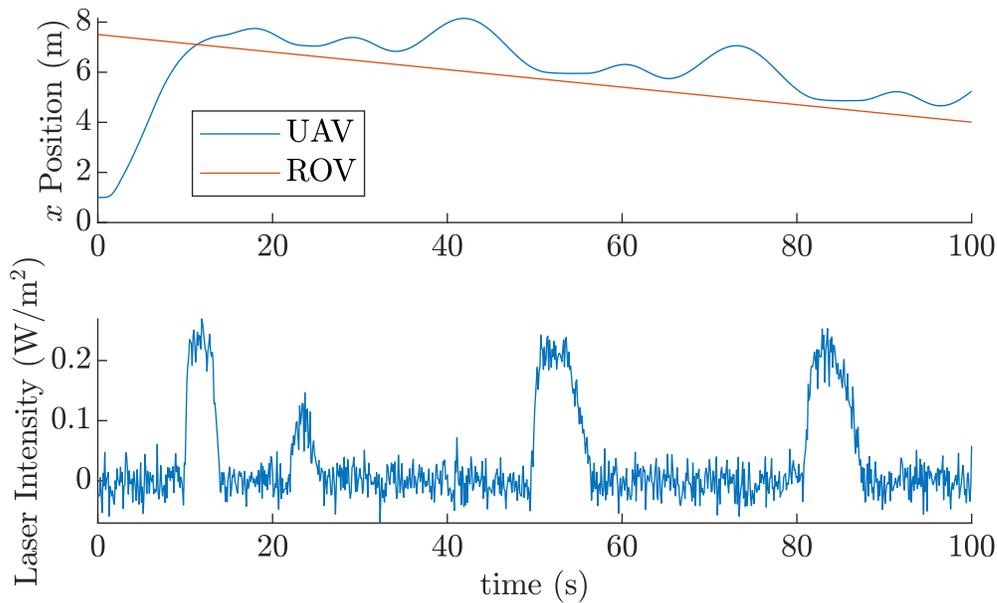


Figure 6.6: Simulated results of the search algorithm with position hold guidance and a moving ROV

as can be seen, the receiver spends very little time in the LDA.

6.2.2.2 Search Pattern with Gradient Ascent

Figure 6.7 shows the results of using the gradient ascent algorithm to track the UAV to the laser axis of a flat-top laser. The results with a Gaussian intensity model are omitted because of their similarity to the shown results. Once the laser is detected, it follows the gradient to be directed toward the LDA centerpoint. Interestingly, the residual motion of the UAV carries it deeper into the flat-top region than the gradient ascent algorithm alone would direct, showing only a small error in the estimate of the laser axis position.

Because the gradient ascent algorithm requires estimation of the spatial gradient, the ROV must either be stationary or have a position estimate in order for the algorithm to work (in the latter case, the relative position can be used to evaluate the spatial gradient).

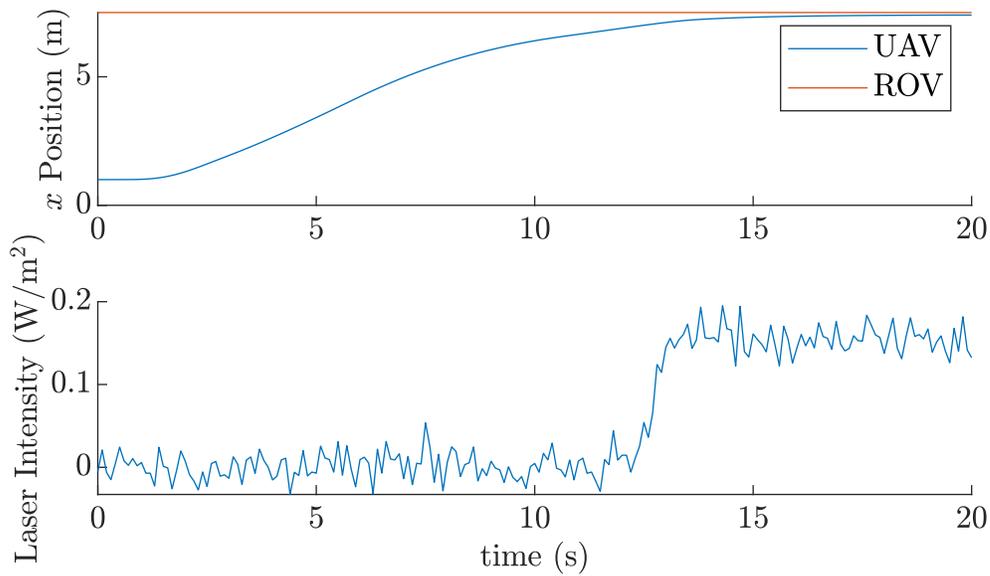


Figure 6.7: Simulated results of the search pattern with gradient ascent on a flat-top laser

For this reason, simulated results with a moving ROV are omitted.

6.2.2.3 Search Pattern with EKF Tracking

For evaluation of tracking the EKF estimate of the LDA centerpoint, only the results from the flat-top laser intensity model are shown, as it is assumed that the estimating the position along the Gaussian intensity model would be easier than with the flat-top model.

Figure 6.8 shows the results with a stationary ROV, and Figure 6.9 shows results with a moving ROV. Note that in both cases, the initial estimate has an offset from the actual LDA centerpoint position, but as it receives laser measurements, that offset is reduced. The performance is shown to be better when the ROV is stationary, but not by much, because the EKF uses the relative velocity measurements to keep track of the LDA movement.

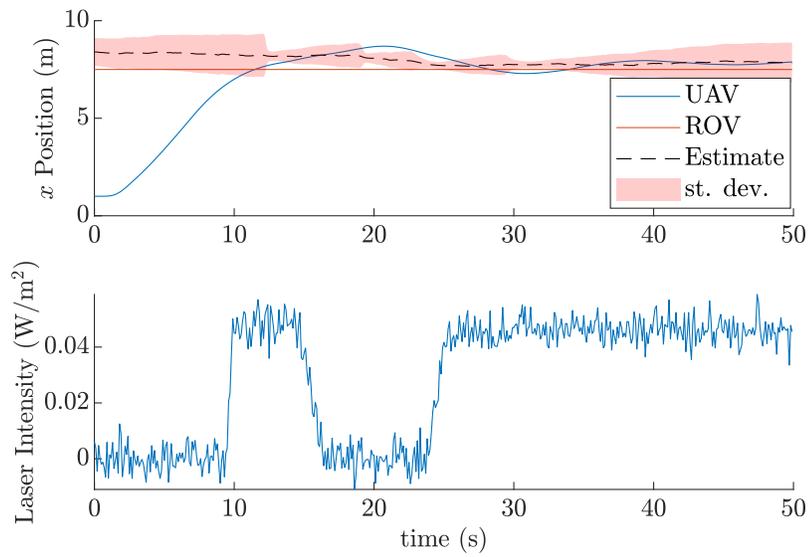


Figure 6.8: Simulated results of the EKF estimation and guidance over a stationary flat-top distributed laser

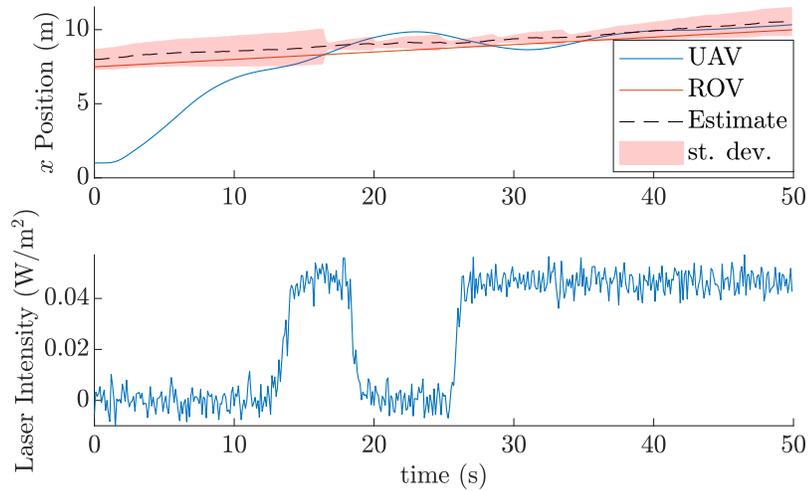


Figure 6.9: Simulated results of the EKF estimation and guidance over a moving flat-top distributed laser

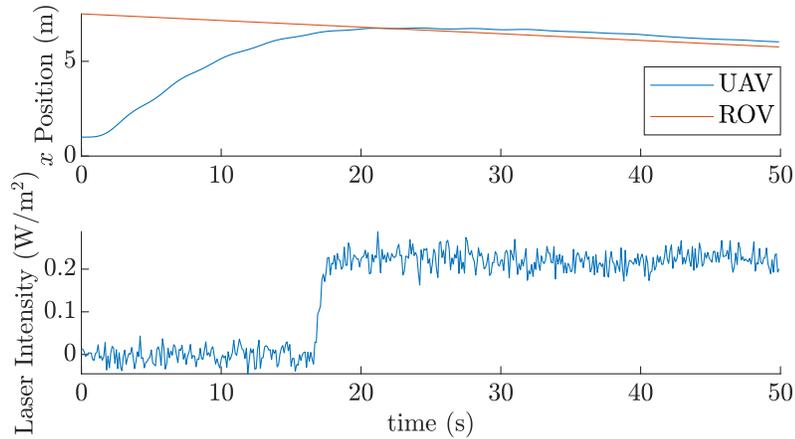


Figure 6.10: Simulated results of using ROV tracking to guide the UAV

6.2.2.4 ROV Position Tracking

Figure 6.10 shows the results from commanding the UAV to track the position of the ROV. Since it is assumed that the laser axis is always directly above the ROV, the only limiting factors for this method are the UAV controller's convergence rate and tracking performance. While the simulated UAV is tuned to move relatively slowly, changes in tuning could allow for quicker and more accurate tracking.

6.3 Experimental Setup

6.3.1 Hardware Laser Models

For the EKF estimation to be effective, the parameters of the laser system must be used in the EKF model. For each decollimating medium (the lens and the diffuser), the effective divergence was verified and the on-axis intensity was found experimentally.



Figure 6.11: Image of the Neutral Buoyancy Research Facility water tank

6.3.2 Test Methods

Testing took place at the Neutral Buoyancy Research Facility (NBRF) at the University of Maryland (see Figure 6.11). The NBRF is one of two currently operating neutral buoyancy tanks in the US, and the tank is 50 feet across, 25 feet deep, and holds 367,000 gallons of water. The ROV with the laser emitter is submerged in the tank, and there is adequate overhead space for the UAV to operate safely. As described in Chapter 2, the ROV has a tether connecting it to the groundstation, while the UAV uses a Wi-Fi connection to communicate with the rest of the system.

As mentioned in Chapter 2, the laser receiver is limited to a FOV of 10 degrees to

limit the noise from the ambient light. Sunlight is a particularly intense ambient light, and was shown to have an immense effect of the noise in the receiver measurement. Although the testing is conducted indoors, the NBRF has some skylights that let in filtered sunlight, which is more than enough to impact the receiver measurements. Unfortunately, the current laser models do not take into account this kind of background noise, so testing was conducted at night as to prevent sunlight from affecting the measurements.

6.4 Experimental Results

6.4.1 Concentric Search

The concentric search algorithm is tested by setting the initial position of the UAV at an offset from the ROV position. The ROV position is fixed by anchoring it to a weight.

Results are seen in Figure 6.12. The UAV begins with a position of about $(0, 0)$ in the x-y plane. Because there is no laser measurement there, it begins the search pattern. Both the commanded position and estimated position are shown. Once it arrives near the x-y coordinate $(0, -3)$, the receiver begins to detect the laser, and the UAV holds position in that area. The laser measurements from the test are shown in Figure 6.13. The z-axis shows the laser intensity. Because the test was conducted at night, there is little ambient noise, up until the receiver enters the LDA of the laser.

While simple, the concentric circle search pattern is shown to be an effective approach to locating the LDA from an initial estimate. It takes into account the UAV's limitations, ensuring that the translational speed is constant even as the radius of the circles increases. This allows the UAV to successfully track the commanded position. As long as the

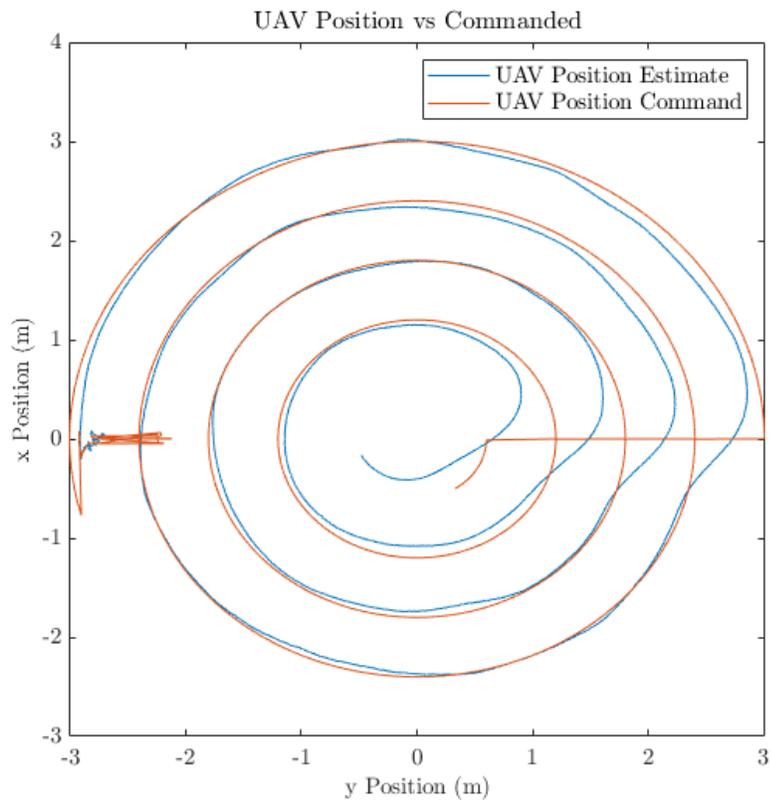


Figure 6.12: Hardware implementation of the concentric search algorithm, starting near the position coordinate (0,0)

Laser Measurements

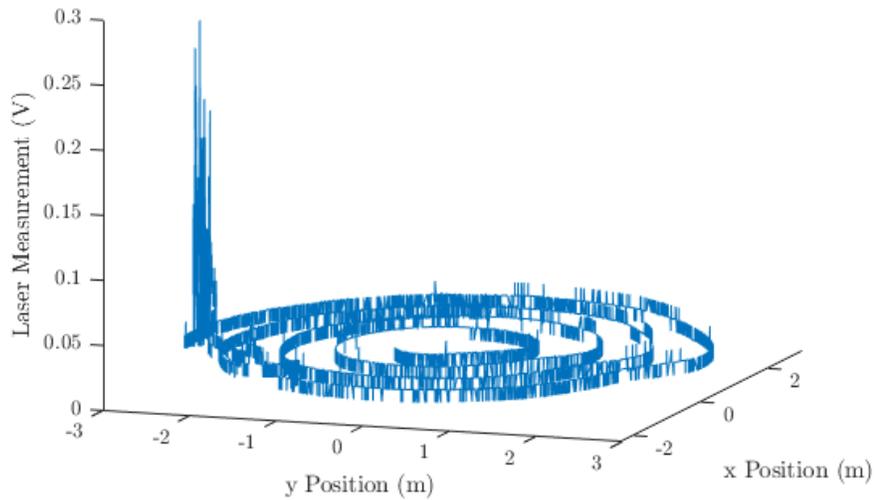


Figure 6.13: Laser intensity measurements made in the LDA in hardware testing

distance between circles is smaller than the diameter of the LDA, then the UAV is able to find the LDA and hold position there.

6.4.2 Gradient Ascent Estimation

Figure 6.14 shows the results from the gradient ascent algorithm in a handheld test, i.e. with the laser emitter above-ground and the UAV being held instead of flown. For the test, the UAV with the mounted receiver is moved through the LDA with nearly-continuous laser measurements. The lens is used as the decollimating apparatus, and thus the laser intensity distribution resembles the Gaussian beam laser model. The receiver is maneuvered on an arbitrary path in the LDA instead of being guided by velocity commands from the algorithm. However, the figure shows the commands that would be relayed to the UAV controller in order to climb the gradient.

It is noted that the majority of the commands, shown by the orange arrows, point

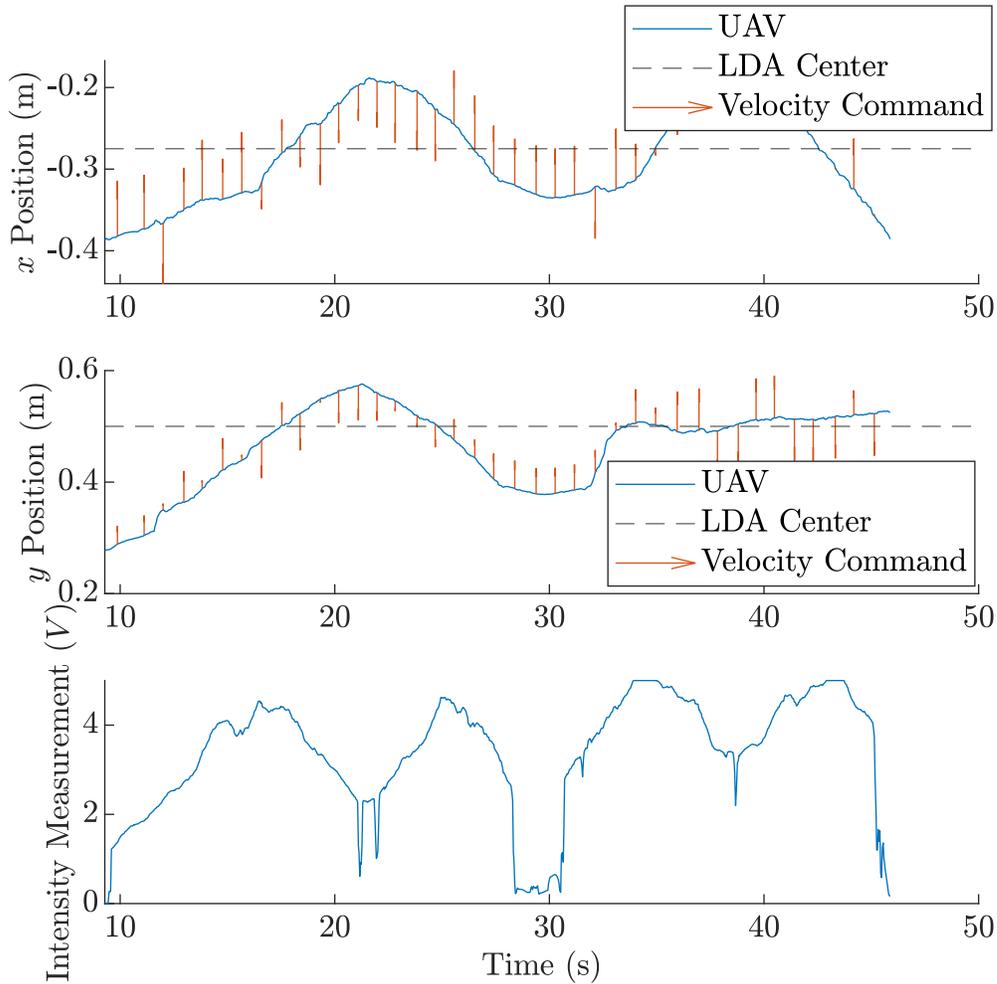


Figure 6.14: Results from the gradient ascent estimation algorithm in handheld testing toward the laser axis, i.e. the centerpoint of the LDA. If the UAV is able to fly and receive continuous measurements as seen in these results, then it is expected that the velocity commands would guide the UAV to the LDA center.

6.4.3 Extended Kalman Filter

Results from testing the extended Kalman filter in a handheld hardware test are shown in Figure 6.15. Similar to the gradient ascent handheld test, the EKF test involves

passing the UAV repeatedly over the LDA. In this case, the x coordinate varies as the receiver moves completely through the LDA, while the y coordinate was maintained nearly constant. Also, this test uses a diffuser as the decollimating apparatus, giving the intensity distribution a flat-top shape.

The EKF estimates the relative distance (r) and angle (θ) from the receiver to the LDA center (see Figure 5.1). The estimate is compared to the actual distance and angle to the LDA center, calculated with the inertial-frame positions of the laser axis and the laser receiver, in Figure 6.15. While the initial estimate has some amount of error in the position estimate, that error is shown to close as laser intensity measurements are made (compare the laser measurements in the second plot to the corrections made to the estimate). Also, note that the errors in the angle θ are relatively insignificant when the distance r is close to zero. Unlike the gradient ascent estimation method, the EKF is able to continue estimating the position of the laser axis when the receiver leaves the LDA.

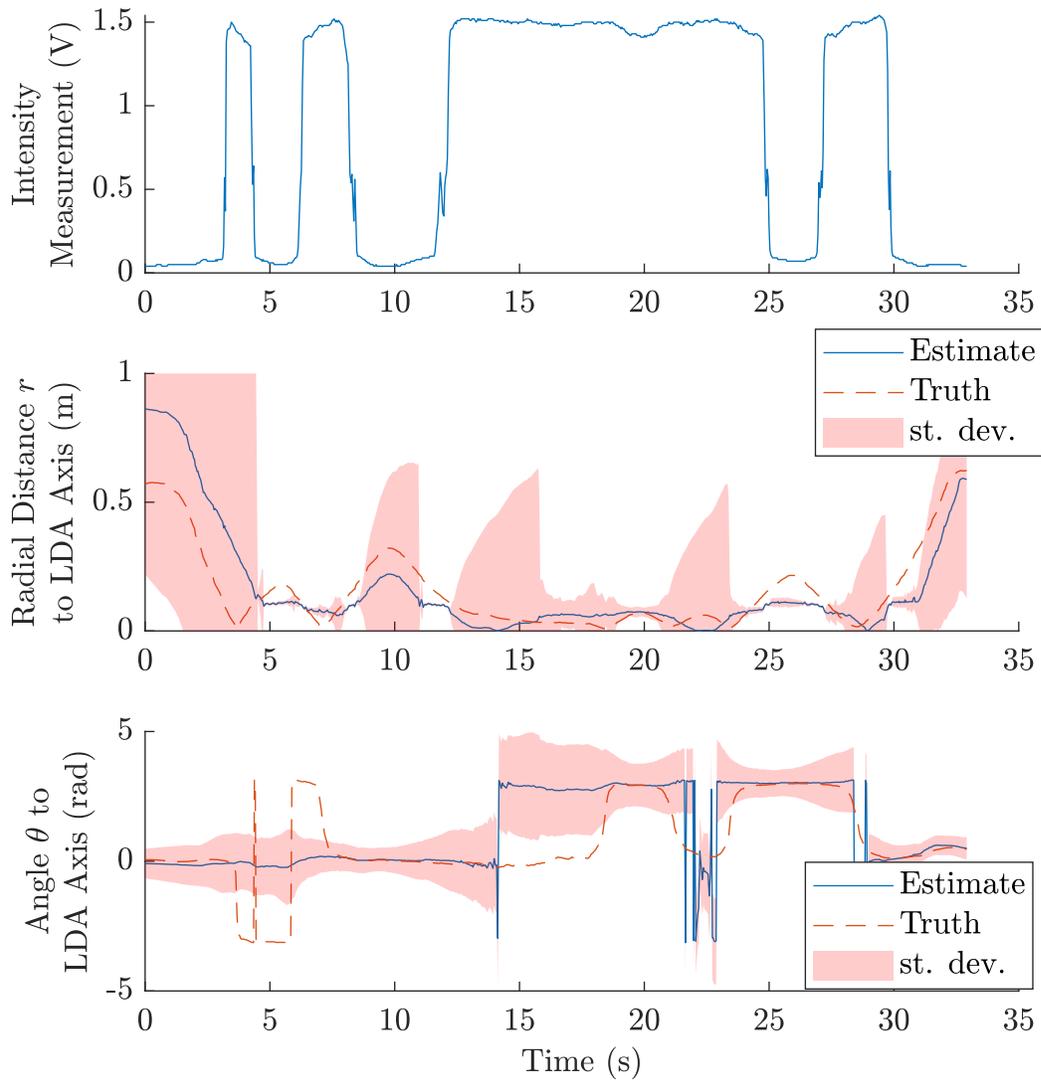


Figure 6.15: Results from the extended Kalman filter estimation algorithm in handheld testing

Chapter 7: Conclusion

7.1 Summary of Contributions

This thesis investigates the use of a quadrotor UAV in novel approach to bistatic underwater laser imaging. This exploration includes considerations in hardware components, system interfaces, modeling and simulation, estimation, and guidance to produce a thorough initial framework for experimentation. An amphibious UAV is developed with unique water-landing capabilities, compatible with a communication framework that allows it to coordinate with the ROV. A super-Gaussian laser intensity model is presented that is useful for both Gaussian and flat-top distributed lasers. This model captures the salient aspects of each laser type, both allowing the lasers to be simulated and providing information for developing laser-based estimation algorithms. An autonomy framework is presented, dividing the UAVs tasks into phases with guidance and estimation algorithms to be used based on measurement availability. Among the developed algorithms are a gradient ascent estimation and guidance method, and an extended Kalman filter based on the laser intensity model. The guidance and estimation algorithms are tested through both simulation and hardware experiments.

7.2 Suggestions for Ongoing and Future Work

Extensions to this work could include development of higher-fidelity laser models. Effects such as refraction and attenuation can be examined. Ambient noise from sunlight or other environmental factors can be included as well, which could remove this work's restriction of testing only at night. Additionally, the relative orientation between the laser emitter and receiver can be included. Because this work's assumption of a level ROV may not represent all test cases, and because the FOV of the receiver is tightly limited, gimbaling the receiver to point at the laser emitter may also be a worthwhile approach.

Further estimation techniques and improvements can be investigated. States such as the relative orientation and ambient noise could be added to this work's EKF. Also, other algorithms can be used for estimation. For example, another bayesian estimator, like a particle filter, may yield favorable results, especially when the measurements don't exactly match the laser model. Other spatial gradient estimation methods can be used, such as using multiple agents to more easily locate the LDA and estimate the spatial laser gradient [31].

Autonomy for the ROV and active coordination between the two agents can also be visited. Autonomous maneuvering for underwater imaging and collision avoidance is an active topic of research [32]. This maneuvering can be made to take into account the sensing limitations of the UAV-mounted receiver, so the UUV can plan its maneuvers and share with the UAV to maximize laser signal strength and connection.

Bibliography

- [1] Derek Alley, Brandon Cochenour, Alan Laux, and Linda Mullen. Multistatic optical imaging system. page 91110L, 5 2014.
- [2] Derek Alley, Brandon Cochenour, and Linda Mullen. Remotely operated compact underwater temporally encoded imager: Cutei. page 982708, 5 2016.
- [3] Derek Alley, Linda Mullen, and Alan Laux. Compact optical system for imaging underwater and through the air/sea interface. page 837203, 6 2012.
- [4] Fraser R. Dagleish, Frank M. Caimi, Walter B. Britton, and Carl F. Andren. Improved lls imaging performance in scattering-dominant waters. page 73170E, 5 2009.
- [5] Jules S. Jaffe. Extended range optical imaging using one and two-dimensional structured illumination. pages 1–4. IEEE, 10 2009.
- [6] Mark E. Kushina, Geoff Heberle, Michael Hope, David Hall, Michael Bethel, and Lonnie K. Calmes. Almds laser system. page 163, 6 2003.
- [7] Paul A. Elmore, Will E. Avera, and Michael M. Harris. Use of the an/aqs-20a tactical mine-hunting system for on-scene bathymetry data. *Journal of Marine Systems*, 78:S425–S432, 11 2009.
- [8] Derek Alley, Linda Mullen, and Alan Laux. Compact, dual-wavelength, non-line-of-sight (nlos) underwater imager. pages 1–5. IEEE, 9 2011.
- [9] Jaime Lara Martinez, Anthony Brescia, Linda J. Mullen, Anthony C. Mulligan, Derek M. Alley, Robert Lautrup, and Drey Platt. Nix usv platform for precision track and trail of uuv platforms. page 29. SPIE, 4 2020.
- [10] Folker Hoffmann, Matthew Ritchie, Francesco Fioranelli, Alexander Charlish, and Hugh Griffiths. Micro-doppler based detection and tracking of uavs with multistatic radar. pages 1–6. IEEE, 5 2016.

- [11] Weijia Wang, Peng Bai, Xiaolong Liang, Jiaqiang Zhang, and Liping Hu. Optimal deployment analysis of airborne radars detecting stealth target using uav swarms. pages 1–8. IEEE, 8 2018.
- [12] A. Lavrenko, Z. Barry, R. Norman, C. Frazer, Y. Ma, G. Woodward, and S. Pawson. Autonomous swarm of uavs for tracking of flying insects with harmonic radar. pages 1–5. IEEE, 4 2021.
- [13] F. Ehlers. Multi-agent teams for underwater surveillance. pages 136–141. IEEE, 9 2010.
- [14] Gabriele Ferri, Roberto Petroccia, Giovanni De Magistris, Luca Morlando, Michele Micheli, Alessandra Tesei, and Kevin LePage. Cooperative autonomy in the cmre asw multistatic robotic network: Results from lcas18 trial. pages 1–10. IEEE, 6 2019.
- [15] Yi Sun, Jizhong Xiao, Xiaohai Li, and Flavio Cabrera-Mora. Adaptive source localization by a mobile robot using signal power gradient in sensor networks. pages 1–5. IEEE, 2008.
- [16] Diana F. Spears, David R. Thayer, and Dimitri V. Zarzhitsky. Foundations of swarm robotic chemical plume tracing from a fluid dynamics perspective. *International Journal of Intelligent Computing and Cybernetics*, 2:745–785, 11 2009.
- [17] Bardienus P. Duisterhof, Shushuai Li, Javier Burgues, Vijay Janapa Reddi, and Guido C. H. E. de Croon. Sniffy bug: A fully autonomous swarm of gas-seeking nano quadcopters in cluttered environments. pages 9099–9106. IEEE, 9 2021.
- [18] Marcelo Jacinto, Rita Cunha, and António Pascoal. Chemical spill encircling using a quadrotor and autonomous surface vehicles: A distributed cooperative approach. *Sensors*, 22:2178, 3 2022.
- [19] Shuo Li and Milutin Stanacevic. Gradient flow source localization in noisy and reverberant environments. pages 257–260. IEEE, 11 2012.
- [20] Dongfang Zheng, Kaiyun Cui, Bo Bai, Gang Chen, and Jay A. Farrell. Indoor localization based on leds. pages 573–578. IEEE, 9 2011.
- [21] Pratap Bhanu Solanki, Mohammed Al-Rubai'ai, and Xiaobo Tan. Extended kalman filter-based active alignment control for led optical communication. *IEEE/ASME Transactions on Mechatronics*, 23:1501–1511, 8 2018.
- [22] Charles J Carver, Zhao Tian, Computer Science, Dartmouth College, Hongyong Zhang, Kofi M Odame, Implementation Nsdi, Charles J Carver, Zhao Tian, Hongyong Zhang, Ko M Odame, Alberto Quattrini Li, and Xia Zhou. Amphilight : Direct air-water communication with laser light this paper is included in the proceedings of the amphilight : Direct air-water communication with laser light. *Nsdi*, 2020.

- [23] James Jackson, Daniel Koch, Trey Henrichsen, and Tim McLain. Rosflight: A lean open-source research autopilot. pages 1173–1179. IEEE, 10 2020.
- [24] Randal W Beard. *Vision-based Estimation and Control of Multirotor Systems*. Brigham Young University, 2021.
- [25] Bahaa E. A. Saleh and Malvin Carl Teich. *Fundamentals of Photonics*. John Wiley & Sons, Inc., 8 1991.
- [26] O. Svelto. *Principles of Lasers*. Springer Science & Business Media, 2010.
- [27] A. Parent, M. Morin, and P. Lavigne. Propagation of super-gaussian field distributions. *Optical and Quantum Electronics*, 24:S1071–S1079, 9 1992.
- [28] Jacek Misiurewicz, Konrad Bruliński, Wiesław Klembowski, Krzysztof Stefan Kulpa, and Jan Pietrusiewicz. Multipath propagation of acoustic signal in a swimming pool—source localization problem. *Sensors*, 22:1162, 2 2022.
- [29] Ming Fang, Hidenori Takauji, Shun’ichi Kaneko, and Hidemi Watanabe. Robust optical flow estimation for underwater image. pages 185–190. IEEE, 9 2009.
- [30] K. Reif, S. Gunther, E. Yaz, and R. Unbehauen. Stochastic stability of the discrete-time extended kalman filter. *IEEE Transactions on Automatic Control*, 44:714–728, 4 1999.
- [31] Mochammad Sahal. Comparison of gradient estimation in cooperative multi-agent source seeking. *JAREE (Journal on Advanced Research in Electrical Engineering)*, 1, 10 2017.
- [32] Leonardo Zacchini, Matteo Franchi, and Alessandro Ridolfi. Sensor-driven autonomous underwater inspections: A receding-horizon rrt-based view planning solution for auvs. *Journal of Field Robotics*, 1 2022.