Spring 2023

An Autonomous Aerial Drone System for Water Fluorescence Mapping and Targeted Sampling

Kazi Ragib Ishraq Sanim

An Autonomous Aerial Drone System for Water Fluorescence Mapping and Targeted Sampling

by

Kazi Ragib Ishraq Sanim

Bachelor of Science in Mechanical Engineering
Bangladesh University of Engineering and Technology 2021

Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science in

Mechanical Engineering

College of Engineering and Computing

University of South Carolina

2023

Accepted by:

Nikolaos Vitzilaios, Major Professor

Michael Myrick, Reader

Cheryl L. Addy, Interim Vice Provost and Dean of the Graduate School

Dedication

"In the name of Allah, The Most Gracious and The Most Merciful"

Acknowledgments

First, I would like to thank my supervisor, Dr. Nikolaos Vitzilaios and the Principal Investigators of this project: Dr. Michael Myrick, Dr. Tammi Richardson, and Dr. Michael Hodgson for their invaluable guidance and feedback during the project. I am sincerely grateful to Zechariah Kitzhaber and Caitlyn English for their collaboration and immense help during the system development and deployment. I also want to thank my colleagues from USRL, Dr. Michail Kalaitzakis, Bhanuprakash Kosaraju, and Keith Lewandowski, for their help, feedback, and support throughout the project. Additionally, my heartiest gratitude goes to my lovely wife and beloved parents for their innumerous sacrifices and unconditional support during my Masters.

I am also grateful to Mr Dick Foote, homeowner and member of the Lake Wateree Association, for permitting us to use his private dock to deploy our system. This project was partially supported by the ASPIRE grant for Research at the University of South Carolina from the office of the Vice President.

Abstract

At present, aerial drones, also known as Uncrewed Aircraft Systems (UAS) or Unmanned Aerial Vehicles (UAV), are increasingly used in time and effort-heavy scientific exploration applications. One such application is inspecting the physical, chemical, and biological state of water bodies. This thesis presents a novel autonomous system capable of sensing water properties and collecting up to three 250 mL water samples from multiple sampling locations. The system features a customized aerial drone with an in-house built fluorescence sensor and pumping mechanism. It can also map the gradient of fluorescent content across the body of water to determine the best sampling spot and perform targeted sampling based on *in situ* fluorescence measurements. Multiple sensor fusion with an Extended Kalman Filter has been implemented for better altitude estimation within 1.5 m from the water surface, and a failsafe routine has been designed to ensure safe near-water operation. The system's performance has been experimentally validated in three different water environments, namely the A.C. Moore Garden Pond, Columbia, SC; Congaree River, SC and Lake Wateree, SC.

Table of Contents

Dedication iii

Acknowledgments iv

Abstract v

List of Tables viii

List of Figures ix

Chapter 1 Introduction 1

1.1 Contributions 3

1.2 Publications 4

Chapter 2 Related Work 6

2.1 Water Sensing and Sampling Aspect 6

2.2 Autonomy Aspect 9

Chapter 3 System Design & Development 17

3.1 UAS Platform 17

3.2 The Water Sensing & Sampling Apparatus (WSSA) 18

3.3 Altitude Estimation using Sensor Fusion 22

Chapter 4 System Control 28

4.1 Mission State Machine 28

4.2 WSSA State Machine 33

Chapter 5 Experiments and Results 36

5.1 EKF Validation Experiment 36

5.2 Water Sampling Experiment 38

5.3 Fluorescence Mapping Experiment 43

5.4 Failsafe Performance during autonomous Operation 46

Chapter 6 Conclusion and Future Work 48

Bibliography 50

Appendix A conversion from LLA to ENU coordinates 55

List of Tables

Table 2.1 Comparison of different UAV-assisted water sampling systems developed by different research groups. (Abbreviations: D = Depth, Tmp = Temperature, EC = Electrical Conductivity, DO = Dissolved Oxygen, Trb = Turbidity, WSD = Water Sampling Device, IEPS = Integrated Electromechanical Pneumatic System, μUV = Micro underwater vehicle, FWS = Floats while sampling, HWS = Hovers while sampling.) 13

Table 2.2 Comparison of different commercially available UAV-assisted water sampling systems. (Abbreviations: D = Depth, Tmp = Temperature, EC = Electrical Conductivity, DO = Dissolved Oxygen, Trb = Turbidity, H_6N_2 = Ammonia Nitrogen, FWS = Floats while sampling, HWS = Hovers while sampling.) 16

Table 3.1 On-board sensing\sampling system components 21

Table 5.1 Waypoints for Sampling and Mapping missions at Lake Wateree . 41

List of Figures

Figure 1.1 The developed UAS water sensing and sampling system during autonomous deployment in Lake Wateree, SC. 3

Figure 3.1 The developed aerial drone sensing and sampling system with highlighted main components. 18

Figure 3.2 3D-printed bottle cap with knurls and barbed connectors. 19

Figure 3.3 Schematic of the Water Sensing and Sampling Apparatus (WSSA). M is the master pump, and A,B, and C are the sampling pumps connected to the bottle A,B, and C respectively. The main channel is shown in blue, the sampling channels are shown is cyan, and the flushing lines are shown in red. 19

Figure 3.4 Depth sensor and Conductivity sensor attached with tube inlet. . 20

Figure 3.5 Transformation of proximity measurement for different orientation of the UAS. 25

Figure 3.6 Altitude measurement from the sonar sensor when the UAS is in flight. The raw sonar reading is shown in blue dots, the max filtered datapoints are shown in orange crosses, and the OptiTrack altitude is shown in black as the ground truth. 27

Figure 4.1 The communication between Mission State Machine and WSSA State Machine during a sampling mission. SF=Sampling Flag, DF=Depth Flag. The yellow boxes indicate the pump flags, the blue boxes indicate the channel flags, and the green boxes indicate the sampling flags. The red dashed rectangle is the workflow of the Mission State Machine, and the green dashed rectangle is the workflow of the WSSA State Machine. 30

Figure 5.1 (a) shows a comparison of altitude measurement from GNSS, OptiTrack, and EKF. The GNSS altitude is shown in blue, the OptiTrack altitude is shown in black, and the EKF altitude estimate is shown in red. The horizontal green dotted line marks the 1.5 m altitude. (b) shows the error in the EKF altitude estimate with respect to the ground truth (OptiTrack). Error when OptiTrack altitude > 1.5 m is plotted in red, and the error when OptiTrack altitude < 1.5 m is plotted in blue. A ± 15 cm error bound is shown is cyan dotted lines. 37

Figure 5.2 Change in depth, fluorescence level, temperature, and total dissolved solids (TDS) in the five different stages (I. Takeoff, II. Approach, III. Sensing, IV. Sampling, and V. Return) of the water sampling mission for a single sample (Moore Garden pond, Columbia, SC, USA). The fluorescence level is interpreted to $\mu g/L$ on the right vertical axis. The red portion of the graphs marks the readings of the sensors, while the sample was being collected. 39

Figure 5.3 Depth, Fluorescence, temperature, and TDS (Total Dissolved Solids) plots from the sampling experiments at (a) Congaree river and (b) Lake Wateree. Red is when the sampling pump was on, Blue is when the sampling pump was off. The horizontal black dashed line shows the target depth and the fluorescence trigger level in their respective plots. The vertical black dashed lines annotated by a,b,c,d,e shows the time instances for mission start, master pump on, fluorescence measurement start, master pump off, and fluorescence measurement end, respectively. 42

Figure 5.4 Google Earth image of (a) Sampling and (b) Mapping experiments at Lake Wateree. The manual flight path is shown in green, and the autonomous flight path is shown in red. (T = Takeoff, L = Land) 43

Figure 5.5 Change in depth, fluorescence level, temperature, and total dissolved solids (TDS) in the three different stages (I. Takeoff and Approach, II. Sensing and Traverse, and III. Return) of the fluorescence mapping mission (Moore Garden pond, Columbia, SC, USA). The fluorescence level is interpreted to $\mu g/L$ on the right vertical axis. 44

Figure 5.6 Depth, Fluorescence, temperature, and TDS (Total Dissolved Solids) plot from the mapping experiments at (a) Congaree river and (b) Lake Wateree. The horizontal black dashed line shows the target depth and the fluorescence trigger level in their respective plots. The vertical black dashed lines annotated by a,b,c,d,e shows the time instances for mission start, master pump on, fluorescence measurement start, master pump off, and fluorescence measurement end, respectively. 45

Figure 5.7 Failsafe performance during autonomous (a) Sampling and (b) Mapping operations. The EKF altitude estimate is shown in magenta, the GNSS altitude is shown in green, and the altitude setpoints are shown in blue. The 'safe' altitude is shown by horizontal black dashed line. 46

Chapter 1

Introduction

A single specimen of water from a river, lake, reservoir, or estuary can reveal a lot of information about a waterbody and its surroundings. However, the time, effort, and resources spent in traditional water sampling methods are quite exorbitant [1]. Uncrewed Aircraft Systems (UAS) or Unmanned Aerial Vehicles (UAV), widely known as aerial drones, can significantly increase the ease, temporal resolution, and spatial scale of water sampling. Moreover, a UAS assisted water sampling system can eliminate the risks of collecting water samples from hazardous locations like mine pit lakes, volcanic lakes, etc. by autonomously collecting adequate samples without human intervention.

This work introduces an aerial drone equipped with a Water Sensing and Sampling Apparatus (WSSA) that can perform a wide-range of *in situ* measurements and collect three 250 mL sensor-triggered samples using a novel method to preserve intact living cells. A custom-made fluorometer [2, 3] (developed by the Myrick group at the Department of Chemistry and Biochemistry, University of South Carolina) is mounted on the drone that can directly measure the amount of chlorophyll-*a* present in the water. The collection and sampling processes are automatically triggered by the *in situ* measurements. Finally, all measurements and samples are stamped with time and location using the drone's Global Navigation Satellite System (GNSS) sensor and thus help to build a geographic information system (GIS) database.

Initially, the drone was manually piloted, and the system was successfully deployed at the A.C. Moore garden pond, Columbia, South Carolina and then the Congaree

River, South Carolina. However, despite this early development serving as a proof of concept [4], there were limitations to the system that required further improvements. One of the limitations was the open-loop control of the WSSA. Another was the lack of accurate sensor information about the proximity to the water surface. During field experimentation, it was observed that the drone's altitude drifts quite a lot while hovering. Since the water sampling application requires the drone to fly very close to the water surface (< 1.8 m), information about the proximity to the water surface is vital. During manual deployment, the safety of the UAS depended on the operator's visual approximation and judgement. However, the visibility of the drone decreased as the distance between the operator and the UAS increased. Therefore, the chance of crashing into the water became high [5]. Since our main objective is to minimize human intervention to save time and effort, an accurate altitude estimation from the water surface was required for a higher level of autonomy.

Therefore, in the final iteration, a set of improvements were made to address these shortcomings. The WSSA was equipped with feedback control by adding water detection sensors at the flushing lines for feedback. A mission state machine was developed so that the UAS could autonomously navigate to user-specified coordinates for sample collection or fluorescence mapping. Additionally, an Extended Kalman Filter was designed and implemented for fusing multiple sensors for better estimating the proximity from the water surface. Paired with that, a failsafe was designed to ensure the safety of the UAS during near-water autonomous operation. The improved system capabilities were also tested in a field deployment at Lake Wateree, SC (Figure 1.1).

The rest of the thesis is organized as follows: Chapter 2 presents related work in the area of UAS-based water sensing and sampling. Chapter 3 discusses the system improvements and the proposed altitude estimation technique. Chapter 4 discusses the autonomous operation of the UAS and the state machine of the WSSA. Chapter

Figure 1.1 The developed UAS water sensing and sampling system during autonomous deployment in Lake Wateree, SC.

5 presents the experimental results from the three deployments and discusses the system's performance. Finally, in Chapter 6, conclusions are drawn, and future research direction is discussed.

1.1 Contributions

The main contributions of this work are as follows:

- The development of an aerial drone system capable of fluorescence-triggered water sample collection from three sampling locations and generate a map of fluorescence gradient along the trajectory.
- The development of a vacuum-assisted Water Sensing and Sampling Apparatus capable of collecting water samples without destroying the living cells.

- The design and implementation of an Extended Kalman Filter for sensor fusion of GNSS and Ultrasound sensor for accurate altitude estimation within 1.5 m of the water surface.
- The implementation of a failsafe routine utilizing the filtered altitude estimate to ensure the safe hovering of the drone over water.
- The development of two state machines for an autonomous mission. One is for the water sample collection process, and the other is for the autonomous navigation to the given locations.
- The field deployment of the platform in three different water environments: A.C. Moore Garden pond, Congaree river, and Lake Wateree situated in South Carolina, USA.

1.2 Publications

In addition to the research presented in this thesis, a list of publications produced related to this research are as follows:

1. **Kazi Ragib Ishraq Sanim**, Caitlyn M. English, Zechariah B. Kitzhaber, Michail Kalaitzakis, Nikolaos Vitzilaios, Michael L. Myrick, Michael E. Hodgson, and Tammi L. Richardson. "Autonomous UAS-based Water Fluorescence Mapping and Targeted Sampling". In: *Journal of Intelligent and Robotic Systems* (Under Review).
2. **Kazi Ragib Ishraq Sanim**, Michail Kalaitzakis, Bhanuprakash Kosaraju, Zechariah B. Kitzhaber, Caitlyn M. English, Nikolaos Vitzilaios, Michael L. Myrick, Michael E. Hodgson, and Tammi L. Richardson. "Development of an Aerial Drone System for Water Analysis and Sampling". In: *2022 Interna-*

tional Conference on Unmanned Aircraft Systems (ICUAS). Dubrovnik, Croatia: IEEE, (2022).

3. Caitlyn M. English, Zechariah B. Kitzhaber, **Kazi Ragib Ishraq Sanim**, Michail Kalaitzakis, Bhanuprakash Kosaraju, James L. Pinckney, Michael E. Hodgson, Nikolaos Vitzilaios, Tammi L. Richardson, and Michael L. Myrick. "Chlorophyll fluorometer for Intelligent Water Sampling Using a Small Uncrewed Aircraft System (sUAS)". In: *Applied Spectroscopy* (2022).

4. Zechariah B. Kitzhaber, Caitlyn M. English, **Kazi Ragib Ishraq Sanim**, Michail Kalaitzakis, Bhanuprakash Kosaraju, Michael E. Hodgson, Nikolaos Vitzilaios, Tammi L. Richardson, and Michael L. Myrick. "Fluorometer Control and Readout Using an Arduino Nano 33 BLE Sense Board". In: *Applied Spectroscopy* (2022).

5. Michael E. Hodgson, Nikolaos Vitzilaios, Michael L. Myrick, Tammi L. Richardson, Matt Duggan, **Kazi Ragib Ishraq Sanim**, Michail Kalaitzakis, Bhanuprakash Kosaraju, Caitlyn M. English, and Zechariah B. Kitzhaber. "Mission Planning for Low Altitude Aerial Drones during Water Sampling". In: *Drones* 6.8 (2022).

Chapter 2

Related Work

Lately, the applications of UAS are not only limited to visual inspections. With the rapid improvement of UAS payload and flight time, UAS are now deployed for highly complex applications like sensor deployment and retrieval [6], multi-robot collaboration [7], aerial manipulation [8], hydrology research [9] etc. For these applications, mathematical modeling of the nonlinear dynamics of real systems is necessary for optimal state estimation and robust control. The nonlinear model of a system can be approximated by a linear model with time-variant parameters. However, the influence of disturbances, modeling errors, and various uncertainties can differ from system to system. Therefore, the design and modeling of the system rely heavily on the intended application.

One such application is UAS-based water sampling. There have been several studies on UAS-based water sampling applications in the recent years. The literature can be grouped in two main categories, based on (a) the type of water sensing and sampling capabilities, and (b) the level of autonomy for the drone operation.

2.1 Water Sensing and Sampling Aspect

For the water sensing and/or sampling aspect, past works can be grouped into three main categories: i) aerial drones deployed only for sampling, ii) aerial drones deployed only to sense different water characteristics, and iii) systems that can do both sensing and sampling in the same mission.

One of the attempts that fall in the first category is the study by Schwarzbach *et*

al. [10, 11], where a helicopter UAS was deployed to collect a water sample of up to 500ml. In a similar study by Cornell *et al.* [12], a small commercial UAS is used to collect 50ml of water for remote sensing of bacterial flora. Doi et al [13], deployed a UAS with a suspended bottle to collect 1L of water for e-DNA survey. Terada *et al.* [14], collected 250ml sample of water from a volcanic crater lake with a sampling bottle with ball valves suspended from a UAS. Banerjee *et al.* [15] developed an Integrated Electromechanical Pneumatic System (IEPS) water sampler with six syringes of 25ml each. It could collect 150ml of water sample, saving much deployment time. The first category also includes the works of Graham *et al.* [16], Neto *et al.* [17], Horricks *et al.* [18], and Farinha *et al.* [19]. Some commercially available solutions that fall into this category are the Nixie water sampler by Reign maker [20] and the remote water sampling platform by UgCS Integrated Systems [21].

The second category, which focuses only on sensing, includes the works of Koparan *et al.* [22], Chung *et al.* [23], and Demario *et al.* [24]. Koparan *et al.* [22], performed *in situ* measurements of Dissolved Oxygen (DO), Electrical Conductivity (EC), pH, and temperature with the help of an open-source multi-probe meter suspended from a custom-built UAS. Chung *et al.* [23] measured the temperature and the corresponding depth of a water body to make a thermal profile. Whereas Demario *et al.* [24] performed thermal imaging and measured the temperature and pressure of a water plume with a quadcopter UAS.

The third category is the most complex category. Because these drones were not only sent to collect water samples but also to perform some *in situ* measurements. The UAS developed by Ore *et al.* [25] was equipped with a temperature and a conductivity sensor to take real-time measurements of water. It also contained three 20ml vials to collect samples from three different locations. Moreover, the apparatus was designed to protect against cross-contamination of the samples by flushing the pump and system before gathering each sample. In another study, a DJI Matrice 600 was used

to collect up to 2L of water while measuring the EC, temperature, and depth at the sampling point on a mine pit lake [26, 27, 28]. In [29], Koparan *et al.* improved their system to collect three 130ml samples while taking *in situ* measurements of EC, DO, pH, and temperature. Esakki *et al.* [30], measured the pH, DO, EC, temperature, and turbidity of the sampling point with their custom-built hovercraft and collected 1L of water sample using a robotic manipulator having a pump as an end-effector. Song *et al.* [31] and Benson *et al.* [32] also proposed systems with both sensing and sampling capabilities. Some commercial solutions such as Nero [33] and SplashDrone 4 WQMS by SwellPro [34] offer some integrated or add-on sensors for *in situ* water quality measurements in addition to the water sampling capabilities. This thesis also fits in this category, as the system can perform both sensing and sampling.

However, within the third category, some platforms are capable of targeted sampling using the sensors onboard. Because, the collection of water samples from random locations and depths might not always contain the analyte of interest. Sensor-triggered sampling ensures a sample is collected only when the measured physical and chemical properties of the water body match some predefined conditions. In [35], Koparan *et al.* developed a water sampling device that would trigger only if the pH, EC, and temperature were outside the allowable limits specified by the Environmental Protection Agency (EPA). Washburn *et al.* [36] fabricated a depth-activated Niskin bottle that would only trigger at a set depth when deployed from a UAS. The platform developed in this thesis also has a targeted sampling feature. The system may only collect a water sample with a target fluorescence level based on user instructions. If the measured fluorescence is outside the desired range, the system may skip sampling from the location and move on to the following location of interest.

2.2 Autonomy Aspect

From the autonomous navigation aspect, the previous works can be categorized based on either manual or autonomous operation. In the works of Song *et al.* [31], Cornell *et al.* [12], Doi *et al.* [13], Castendyk *et al.* [26, 27, 28], Banerjee *et al.* [15], Pinton *et al.* [37], and Reign Maker [20], the UAS was flown manually to the point of interest. However, in this chapter, our main focus is the studies done with autonomous flights.

One of the first autonomous attempts at sampling was made by Schwarzbach *et al.* [10]. In this work, an onboard computer functioned as a flight controller, and the sensors and actuators were connected over a serial interface with the computer. The autopilot system was programmed using MATLAB/Simulink, while the position information was fetched from a differential GPS system. The UAS was led to the sampling position using standard guidance algorithms. All systems were connected to the PLANET framework, which stored data and sent control commands via a central system including a special user interface for the end users [11]. The UAS was capable of autonomous takeoff and landing, however, it didn't land on water, rather collected a sample while hovering.

In Koparan *et al.* [22, 29], autonomous control of the UAS navigation and sampling was accomplished with a Pixhawk flight controller. The Mission Planner[1] software was used to specify flight boundaries, waypoints, and autonomous navigation details. The water sensing node was activated when the UAS reached a sampling location. Since the UAS had the capability to takeoff and land on water. Therefore, precise altitude estimation near water was not required and only the position information from GPS was adequate for the proper operation of the UAS.

For making non-contact water plume temperature measurements, Demario *et al.* [24] used a Pixhawk flight controller and pre-programmed the waypoints using the

[1] `https://ardupilot.org/planner/`

Mission Planner software. In this case, the UAS did not come in close proximity with the water surface. In a different study by Terada *et al.* [14], optimal waypoints for the sampling locations were programmed into the flight control unit for autonomous navigation to the sampling points. The sampling bottle was suspended with a 30 m long rope from the UAS. Therefore, the operating altitude was at around 30 m from the ground level and the position information from GNSS was sufficient to keep the UAS safe. The UAS used by Washburn *et al.* [36] used the Ardupilot flight control software for autonomous navigation. Similarly, Singh *et al.* used the smart-RTL feature of Ardupilot to navigate back and forth from the starting point to the sampling waypoints [38]. In the work of Benson *et al.* , the Waypoint Intelligent Flight mode in the DJI GO 4 app was used to establish a grid of defined GPS waypoints [32]. Neto *et al.* also used the DJI GO 4 app to set the waypoints [17]. Horricks *et al.* used the open-source software QGround Control[2] for semi-autonomous navigation to the waypoints [18].

The systems discussed till now mostly utilized an off-the-shelf flight planning software (Mission Planner, QGround Control, etc.) to achieve autonomous navigation capabilities. However, there are two problems with this approach when it comes to specialized missions like water quality sensing and targeted sampling.

- First, the limited flexibility of missions in such software.
- Second, the mission programmed by the flight planning software operates using the positioning from GNSS, which does not provide precise enough altitude information during near-water operation. Adding new sensors for precise positioning is possible but difficult to integrate with such software.

One way to solve the first issue is by designing a dedicated state machine for a specialized water sensing and sampling mission. Solving the second issue becomes

[2] `http://qgroundcontrol.com/`

crucial when the UAS must hover close to the water surface during sampling. Most of the works discussed till now have bypassed this problem either by hovering at a high and safe altitude utilizing a long tether for the sampling device or by landing on water with the help of a floatation attachment. However, in our case, hovering close to the water surface is essential to generate a fluorescence map and collect targeted samples. The systems with the most similar goals to ours are the water sampling UAS developed by UgCS Integrated systems [21] and Ore *et al.* [25].

The commercially available remote sampling platform by UgCS Integrated Systems uses an onboard computer UgCS SkyHub and an in-house flight planning software UgCS PRO for autonomous navigation to pre-programmed sampling locations [21]. It has a precise radar altimeter for real-time altitude measurement during the flight and control of descent for sampling. In the system proposed by Ore *et al.* [25], a ground station with Robot Operating System (ROS) handled the low-level communication with the UAS for relaying information like mission control, navigation, and altitude estimation. There was a second embedded controller on the UAS, which received the instructions from the ground station and controlled the water sampling subsystem and sent feedback. Since the UAS came in close proximity to the water to collect a sample, positioning from GNSS was not adequate to ensure the safety of the drone. Therefore, the researchers designed a low-altitude Kalman filter utilizing two ultrasonic sensors and a pressure altimeter to estimate the altitude below 1.85 m from the water surface. They also implemented a scoring heuristic as a prefilter to reject false measurements caused by sensor occlusions by the suspended pipe. At high altitudes, the mission control guided the UAS towards the sampling point using GNSS position feedback, but during near water operation, the low-altitude flight controller took over to ensure safe hovering of the UAS. In the next iteration of their work [39], Ore *et al.* estimated the altitude using a depth sensor at the end of the suspended pipe for near-water hovering. However, the depth of the pipe end was not a

direct measurement of the altitude, therefore, prone to errors. Most importantly, for this technique to work, the suspended pipe had to be either rigid or always pointing straight down, which was not always the case during flight.

In this thesis, two state machines were developed for autonomous navigation and water sampling respectively. To solve the near water altitude estimation problem, a low-altitude Kalman filter was developed for safe operation near water. However, instead of fusing two ultrasonic sensors, the altitude data from GNSS, pressure altimeter, and a single ultrasonic sensor were fused for a better altitude estimate near water. A measurement gating was in place for outlier rejection instead of a scoring heuristic-based prefilter. The mission control, navigation, and altitude estimation happens on the onboard computer of the UAS in a ROS framework, while the Pixhawk flight controller handles the low-level controls. Therefore, the UAS can make decisions faster. Another novel aspect of this work is the autonomous fluorescence mapping mission, where the UAS safely traverses close to the water to make a fluorescence map along a trajectory.

A comprehensive comparison has been made among different UAS-based water sampling platforms developed by different research groups, including the proposed system in this thesis in Table 2.1 and similar commercially available products in Table 2.2. The comparison has been made concerning the base UAS platform used, sampler size, *in situ* measurement capabilities, water collection mechanism, and near water altitude awareness. Some special features of the platforms have also been mentioned in the *Remarks* column, including whether the UAS floats on water or hovers in the air during sample collection.

Table 2.1 Comparison of different UAV-assisted water sampling systems developed by different research groups. (Abbreviations: D = Depth, Tmp = Temperature, EC = Electrical Conductivity, DO = Dissolved Oxygen, Trb = Turbidity, WSD = Water Sampling Device, IEPS = Integrated Electromechanical Pneumatic System, μUV = Micro underwater vehicle, FWS = Floats while sampling, HWS = Hovers while sampling.)

Author Name	*Base Platform*	*Sampler Size (mL)*	*In-situ Measurements*	*Collection mechanism*	*Near Water Altitude Awareness*	*Remarks*
Schwarzbach *et al.* [10, 11]	Helicopter, payload: 4 kg	500x1	None	Submersible pump	None	HWS
Ore *et al.* [25, 39]	Hexacopter (Asctec Firefly), payload: 600 g	20x3	D, Tmp	Submersible pump	Sensor fusion using pressure altimeter & ultrasonic sensor	HWS, detects sensor occlusion
Song *et al.* [31]	Hexacopter (Asctec Firefly)	20x3	D, Tmp, EC	Peristaltic pump	None	HWS
Cornell *et al.* [12]	Quadcopter (DJI Phantom 3)	50x1	None	Falcon tube	None	FWS
Doi *et al.* [13]	Hexacopter (Luce Search)	1000x1	None	Bottle with weighted ring	None	HWS, bottle suspended with 10m tether

Continued on next page

Table 2.1 – continued from previous page

Author Name	*Base Platform*	*Sampler Size (mL)*	*In-situ Measurements*	*Collection mechanism*	*Near Water Altitude Awareness*	*Remarks*
Koparan *et al.* [29, 35]	Hexacopter (Custom), payload: 2.1 kg	130x3	DO, EC, pH, Tmp, D, Trb	Custom WSD	None	FWS, sensor-triggered sampling
Terada *et al.* [14]	Hexacopter (EnRoute lab LAB645)	250x1	None	Sampling bottle with valves	None	HWS, bottle suspended with 30m tether
Esakki *et al.* [30]	Quadcopter (Custom), payload: 7 kg	1000x1	pH, DO, EC, Tmp, Trb	Submersible Pump	None	FWS, Amphibious, robotic arm
Washburn *et al.* [36]	Not Specified	500x1	D	Depth-activated Niskin bottle	None	HWS
Castendyk *et al.* [27, 28]	Hexacopter (DJI Matrix 600), payload: 6 kg	2000x1	EC, Tmp, D	Hydrasleeve	None	HWS, sleeve suspended with 100m tether
Banerjee *et al.* [15]	Octocopter (Walkera QR-X900)	25x6	None	Custom IEPS	None	HWS, 2.5m suspended tube

Continued on next page

Table 2.1 – continued from previous page

Author Name	*Base Platform*	*Sampler Size (mL)*	*In-situ Measurements*	*Collection mechanism*	*Near Water Altitude Awareness*	*Remarks*
Singh *et al.* [38]	Quadcopter (Hudson EDU 450), payload: 1.33 kg	95x1	Tmp, pH, EC	Diaphragm pump	Ultrasonic sensor	HWS, retractable tube, plug-and-play
Benson *et al.* [32]	Quadcopter (DJI Phantom 4)	50x1	Tmp, D	Sampling tube	None	HWS, creates bathymetric map
Neto *et al.* [17]	Quadcopter (DJI Mavic Pro)	50x2	None	Peristaltic pump and solenoid valve	None	HWS
Horricks *et al.* [18]	Quadcopter (Spiri Mu), payload: 1 kg	250x1	None	Sampling bottle	None	HWS
Farinha *et al.* [19]	Hexacopter (Tarot X6 Frame)	200x1	None	Peristaltic pump and filters	None	FWS, deploys a depth-controlled μUV
Proposed system	Hexacopter (Aurelia X6 Standard), payload: 5 kg	250x3	T, D, EC, Fluores-cence	Vacuum-assisted sample collection by diaphragm pump	Sensor fusion using GNSS, pressure altimeter & ultrasonic Sensor	HWS, targeted sampling, creates fluorescence map

Table 2.2 Comparison of different commercially available UAV-assisted water sampling systems. (Abbreviations: D = Depth, Tmp = Temperature, EC = Electrical Conductivity, DO = Dissolved Oxygen, Trb = Turbidity, H_6N_2 = Ammonia Nitrogen, FWS = Floats while sampling, HWS = Hovers while sampling.)

Developer Name	*Base Platform*	*Sampler Size (mL)*	*In-situ Measurements*	*Collection mechanism*	*Near Water Altitude Awareness*	*Remarks*
Reign Maker [20]	Quadcopter (DJI M300 RTK) / Hexacopter (DJI M600)	250x1	None	Nixie	None	HWS
Nero [33]	Quadcopter (DJI M300 RTK)	250x4 (Nero Poli), 2000x1 (Nero Mia)	Tmp	Peristaltic pump	None	HWS
UgCS Integrated Systems [21]	Quadcopter (DJI M300 RTK) /Hexacopter (DJI M600 Pro)	1000x1 or 5000x1	None	Ruttner Water Sampler	Radar Altimeter	HWS
SwellPro [34]	Quadcopter (Splashd-Drone 4 WQMS)	350x1	DO, EC, Trb, pH, H_6N_2	Sampling bottle	None	FWS, waterproof

Chapter 3

System Design & Development

In this chapter, the UAS platform and the design and development of the water sensing and sampling apparatus are first discussed. Then the design and implementation of the Extended Kalman Filter for sensor fusion is explained.

3.1 UAS Platform

The developed system is based on the Aurelia X6 Standard LE aerial drone platform, a U.S. made, development focused platform with the open-source flight controller Pixhawk Cube Blue. It has in-built triple-redundant accelerometers, and gyroscopes. The Aurelia platform offers a high payload capacity (5 kg) and flight time (up to 45 min). The platform was modified to allow for the placement of a custom-built fluorescence sensor [2, 3], a water sampling mechanism, a Raspberry Pi 4 model B (RPi 4B), an off-the-shelf ultrasonic distance sensor, a depth sensor, and a conductivity sensor. Several parts were designed, and 3D printed to mount the peripherals. Figure 3.1 shows the retrofitted drone with its different components.

A Raspberry Pi 4 model B acts as the on-board computer and can communicate with the Pixhawk flight controller and the added sensors. It utilizes the Robot Operating System (ROS)[3] framework for overall system control. It facilitates communication between different peripherals and runs the water sampling state machine, the mission state machine, and the Extended Kalman filter.

[3] `https://www.ros.org/`

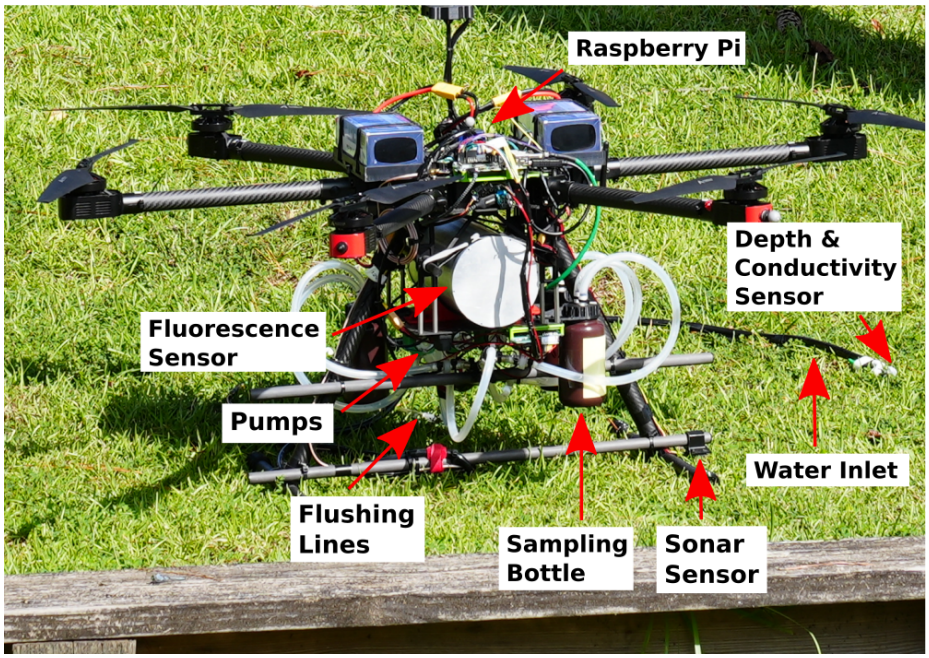


Figure 3.1 The developed aerial drone sensing and sampling system with highlighted main components.

3.2 The Water Sensing & Sampling Apparatus (WSSA)

The main objective of the WSSA, is to collect water samples without damaging the living cells in the water. It comprises three sampling bottles, each with a capacity of 250 mL, and four self-priming, diaphragm, micro water pumps each capable of creating up to -35 KPa of vacuum. Black silicon tubing is used for the input lines. Three 3D-printed bottle caps with two barbed connectors on each are designed and mounted on the UAS. The bottles can be screwed or unscrewed to the cap for rapid replacement and redeployment. In the bottle cap design, several ridges or knurls are added to the side of the caps for reinforcement. Also, the base of the barbed connectors is rounded to reduce the shear force. The design of the 3D-printed bottle caps is shown in Figure 3.2.

Among the four identical pumps, one acts as the master pump and pulls water through the main channel, while the remaining three act as sampling pumps tasked to fill each sampling bottle. As shown in Figure 3.3, the main channel of the tubing

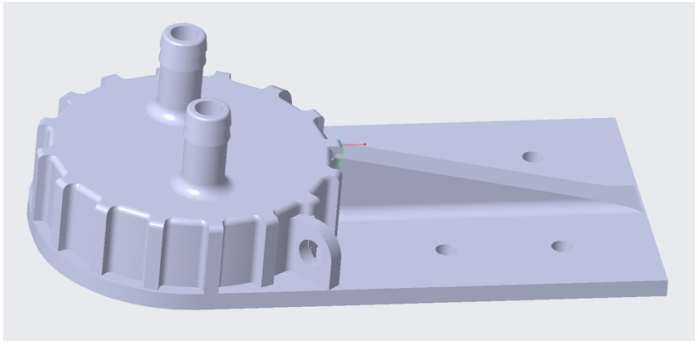


Figure 3.2 3D-printed bottle cap with knurls and barbed connectors.

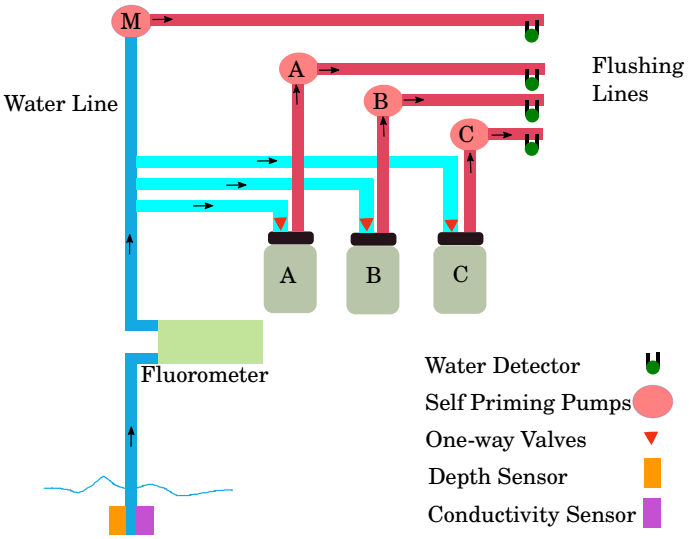


Figure 3.3 Schematic of the Water Sensing and Sampling Apparatus (WSSA). M is the master pump, and A,B, and C are the sampling pumps connected to the bottle A,B, and C respectively. The main channel is shown in blue, the sampling channels are shown is cyan, and the flushing lines are shown in red.

circuit comprises the fluorometer and a 1.8 m long tube suspended from the landing gear of the drone. A mesh grid at the tube inlet prevents anything bigger than 200 μm in diameter from entering the system. The depth sensor and the conductivity sensor probes are fastened to the tube inlet as shown in Figure 3.4. Following the fluorometer, the main channel splits into three separate sampling channels going to the three sampling bottles and one exit channel for continuous flushing. One of the two connectors on each bottle cap is connected to a sampling channel, and the other is connected to a sampling pump. Finally, there are three individual flushing lines,

one from each sampling pump for flushing of excess water when the bottles are full. In the initial design [4], the continuous flushing line and the flushing lines from the bottles were all connected to a common line. In the current design, the four lines are kept separate.

Figure 3.4 Depth sensor and Conductivity sensor attached with tube inlet.

A custom water detection sensor (using BC547 npn transistor) is installed in each flushing line. Separation of the flushing lines ensures that the water detection sensors at each line are not triggered by water from another line. Each sensor has two probes that are pierced into each flushing line for feedback when water is flowing. When water flows through the lines, the probes become electrically connected, and the sensor sends a feedback signal to the Raspberry Pi. These feedback signals indicate when the main channel is primed and bottles A,B,C are full, respectively.

Another modification in the improved setup is the section of the main channel with connections going to each bottle has been moved below the level of the bottles to prevent any cross-contamination of the samples due to gravitational pull. The diameter of the tubing has been changed. Previously 3/8 inch inner diameter (ID) black tubing was used for the WSSA. In the current setup, it was replaced with 1/4 inch ID black tubing for the whole apparatus. Switching to 1/4 inch tubing saved a total payload of 0.25 kg and also improved the flow rate of the master pump. New one-way valves (rated for vacuum up to -300 KPa) are placed in the sampling lines to restrict the flow of air from the bottles to the main pump, therefore allowing adequate

Table 3.1 On-board sensing\sampling system components

Item	Weight(g)	Units	Total Weight(g)
Pump	68	4	272
Bottle (Full)	320	3	960
Pipe & Valves	1241	1	1241
Fluorescence Sensor	1300	1	1300
Raspberry Pi	48	1	48
Conductivity Sensor	56	1	56
Depth\Pressure\Temp Sensor	40	1	40
Custom electronics	121	1	121
3D Printed Parts	458		458
		Total	4496

vacuum to be created. A simplified schematic of the improved WSSA is shown in Figure 3.3, and the weight of different components of the improved WSSA is shown in Table 3.1.

The water sampling apparatus working process is the following:

- First, the master pump creates a vacuum in the main channel when the suspended end of the tube is inside the water. Because of the vacuum, water starts to enter into the tube and rise up to the fluorometer.
- When none of the sampling pumps are running, the water exits from the fluorometer and goes directly to the flushing line after passing through the master pump.
- When one of the sampling pumps is started, it creates a vacuum in the corresponding bottle, therefore, water starts flowing from the main channel towards the bottle.
- When the bottle is full, water flows out of it, passes through the sampling pump, and exits through the flushing line.

Since the water does not contact any moving parts before the fluorescence sensor and the containers, the organic matter in the water sample is not introduced to sudden change in pressure. Therefore, the organic matter present in the water samples is collected unharmed.

3.3 Altitude Estimation using Sensor Fusion

For the water sampling and mapping experiment, a precise sense of altitude is required to ensure the safety of the UAS. The UAS avionics sensors report the altitude using the GNSS, barometric sensor, and the Inertial Measurement Unit (IMU) data. The GNSS receiver on the Aurelia drone in this study used a single-frequency receiver. Although capable of augmentation, such as Satellite based augmentation system (SBAS), the receiver was only operated in single-frequency Civilian Acquisition (C/A) code. Recent estimates for vertical accuracy for a single constellation, such as GPS, using only C/A code is ~ 5 m (at the 95% confidence level) [40]. Additionally, the barometric data drifts over time [41]. The IMU data originally measures the acceleration, and then integrates twice to get the altitude, therefore, it contains a high integration error. Thus, relying completely on the altitude reading from the UAS avionics sensors is not safe. Furthermore, the reported altitude from the Pixhawk flight controller used in this study is relative to the takeoff level. However, the water level at the point of interest may not be at the same level as the launch site.

The use of a proximity sensor can provide the correct distance from the surface beneath. The Aurelia X6 standard drone comes with a LiDAR sensor[4] with a range of 12 m. But optical proximity sensors have poor performance on water because of the low return. To overcome this issue, an Ultrasonic Distance Sensor (Sonar) from Parallax Inc.[5] was added to the drone in this work. This sensor was mounted on the drone's landing gear facing downwards for water proximity detection while hovering above water. The newly mounted sonar sensor was specified to have a range of 3 m but had an excellent return from the water surface. However, when it was mounted on the drone, it was affected by the vibration of the drone. As a result, the useful range reduced to 1.5 m and the outliers in the sensor readings also increased

[4] `https://aurelia-aerospace.com/product/aurelia-x6-standard/`

[5] `https://www.parallax.com/product/ping-ultrasonic-distance-sensor/`

drastically. Additionally, there was a suspended tubing that occasionally occluded the sensor which also contributed to outliers. Therefore a robust outlier rejection technique was required.

In this section, an Extended Kalman Filter (EKF) is proposed which can estimate altitude over water by fusing the GNSS altitude and the sonar sensor reading. The proposed EKF can also estimate the GNSS drift as an extra state modeled to have a random walk noise. Subtracting the estimated GNSS drift from the GNSS altitude and fusing it with the sonar measurements improved the accuracy of altitude estimate drastically within a 1.5 m range. There was also a measurement gating in place which removed the outliers caused by the noisy sensor reading or pipe occlusions. The performance of this filter is discussed in Section 5.1.

3.3.1 Process Model

The proposed Extended Kalman Filter has five states. The altitude z, the vertical velocity $\dot{z}$, the roll angle ϕ, the pitch angle θ, and finally, the GNSS offset γ.

$$x = \begin{bmatrix} z & \dot{z} & \phi & \theta & \gamma \end{bmatrix}^T \tag{3.1}$$

The altitude, the roll angle and the pitch angles are assumed to have a constant velocity model. The vertical acceleration, the roll rate, and the pitch rate readings from the IMU are used to propagate the states. The GNSS offset models the drift in the altitude reported from the GNSS module. It is assumed to have a random walk error. Estimating this offset enables us to subtract it from the GNSS altitude measurement in each iteration to estimate the true altitude. The state representation in the continuous time domain is as follows:

$$
\begin{aligned}
\dot{z} &= \dot{z} \\
\ddot{z} &= R_B^W a_{input}^B - g \\
\dot{\phi} &= \dot{\phi}_{input}^W \\
\dot{\theta} &= \dot{\theta}_{input}^W \\
\dot{\gamma} &= 0
\end{aligned}
\tag{3.2}
$$

Here, a_{input}^B is the 3-axis acceleration vector from the accelerometer of the IMU in all three axes of the body frame, $\dot{\phi}_{input}^W$ is the roll rate, and $\dot{\theta}_{input}^W$ is the pitch rate reported from the gyroscope of the IMU. g is the gravitational acceleration. R_B^W is the third row vector of the three-dimensional rotation matrix. Therefore, $R_B^W a_{input}^B$ is the acceleration along the z-axis of the world frame. The discrete time equations for a time step dt are as follows:

$$
\begin{aligned}
z_{k+1} &= z_k + \dot{z}_k dt + w_z \\
\dot{z}_{k+1} &= \dot{z}_k + (R_B^W a_{input}^B - g)dt + w_{\dot{z}} \\
\phi_{k+1} &= \phi_k + \dot{\phi}_{input}^W dt + w_{\dot{\phi}} \\
\theta_{k+1} &= \theta_k + \dot{\theta}_{input}^W dt + w_{\dot{\theta}} \\
\gamma_{k+1} &= \gamma_k + w_\gamma
\end{aligned}
\tag{3.3}
$$

Here, k and $k+1$ are two subsequent time steps. $w_z, w_{\dot{z}}, w_{\dot{\phi}}, w_{\dot{\theta}}$, and w_γ are the process noise modeled as white Gaussian zero-mean noise for each state variable respectively. To get a close approximation of the process noise covariance, the sensor data was recorded from the drone's onboard avionics for two minutes while the drone was stationary. The recorded altitude data were observed to contain a random walk bias. However, the first difference of the recorded altitude data followed a zero mean Gaussian distribution. The variance of this distribution was taken as an initial estimate for w_γ. Later it was further tuned depending on the filter results. The recorded vertical acceleration, roll rate, and pitch rate also followed zero mean Gaussian distributions. Therefore, the variance of these distributions were taken as the initial

estimates for $w_{\dot{z}}, w_{\dot{\phi}}, w_{\dot{\theta}}$ respectively and then further tuned. Finally, w_z was tuned by trial and error.

3.3.2 Measurement Model

GNSS altitude measurements (fused with barometric data) and the roll and pitch angle measurements could be obtained from the Flight Control Unit (FCU). The sonar sensor mounted underneath the drone reported the proximity from the surface beneath the drone along the sensor frame. The sensor frame and the drone's body frame had an offset along the transverse axis since the sensor was mounted a few centimeters off-center along the pitch/transverse axis. When the drone was hovering steadily, the body and world frames aligned, therefore, the sensor measurement was equal to z. However, when the drone rolled and/or pitched at an angle, appropriate transformation from the drone's body frame was required to obtain the altitude measurement in the world frame.

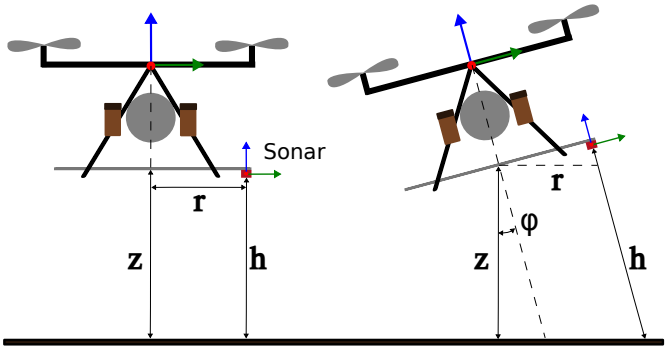


Figure 3.5 Transformation of proximity measurement for different orientation of the UAS.

Figure 3.5 shows the placement of the sonar sensor and the relation between z and the sonar measurement, h is shown in Eq. 3.4.

$$z = hcos(\phi)cos(\theta) - rsin(\phi)cos(\theta) \tag{3.4}$$

The measurement equations are as follows:

$$\begin{aligned} h_1 &= z + \gamma + v_{GNSS} \\ h_2 &= \frac{z + rsin(\phi)cos(\theta)}{cos(\phi)cos(\theta)} + v_{sonar} \\ h_3 &= \phi + v_\phi \\ h_4 &= \theta + v_\theta \end{aligned} \tag{3.5}$$

Here, h_1 is the GNSS altitude measurement from the drone avionics sensors. This GNSS altitude contains the true altitude (z), the random offset (γ) and sensor noise (v_{GNSS}); h_2 is the sonar sensor measurement. r is the mounting offset of the sonar sensor along the transverse axis of the drone. Finally, h_3 and h_4 are the roll and pitch angle measurements reported from the FCU. $v_{GNSS}, v_{sonar}, v_\phi$, and v_θ are the variance of the white Gaussian zero-mean noise of the respective sensors. The values for each of them can be found from the sensor measurements except for v_{GNSS} since the GNSS measurement also contained a random offset. The process noise for the offset, γ contained the GNSS sensor noise as well. However, It was assumed v_{GNSS} to be much smaller than w_γ. It was observed that assuming v_{GNSS} to be 20 times smaller than the w_γ gives decent filter performance. For finding appropriate values for v_{sonar}, v_ϕ, and v_θ, the stationary drone's sonar sensor data, and gyroscope data were recorded for two minutes. The variance of the recorded data were taken as v_{sonar}, v_ϕ, and v_θ for the respective sensors.

3.3.3 Measurement Gating for Outlier Rejection

Even though the sonar sensor was specified to have a range of 3 m, the vibration of the drone in-flight deteriorated the sensor's performance. The number of outliers in the sensor measurement increased with the increase in thrust, thus, during ascend. Beyond 1.5 m, the sensor data lost significance. Additionally, the sensor could be occluded by the suspended 1.8 m long tube under the drone, which might swing

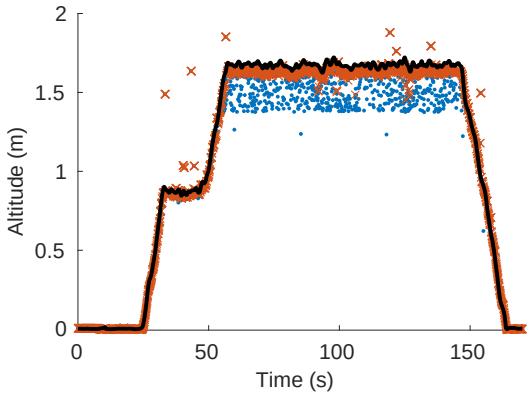


Figure 3.6 Altitude measurement from the sonar sensor when the UAS is in flight. The raw sonar reading is shown in blue dots, the max filtered datapoints are shown in orange crosses, and the OptiTrack altitude is shown in black as the ground truth.

and intercept the sonar sensor's cone of measurement. To reject tube intercepts and outliers, two layers of outlier rejection were employed:

- The first layer was a max filter with a moving window size of five data points. While the UAS was in-flight, the sensor was observed to be more susceptible to low outliers, as shown in Figure 3.6, specially close to 1.5 m. The max filter only accepts the maximum of every five sensor measurements. Therefore, it eliminated all the low outliers as can be seen in Figure 3.6. The remaining high outliers were further eliminated by the next layer.

- The second layer was a χ^2 test to reject any remaining outlying sensor measurement. The χ^2 test used the sensor measurement covariance as follows:

$$\chi^2 = (z_i - \hat{z}_i)S_i^{-1}(z_i - \hat{z}_i) \tag{3.6}$$

Here, z_i is the sensor measurement, $\hat{z}_i$ is the predicted measurement, $S_i = HPH^T + R$ is the innovation covariance matrix for a sensor with observation matrix H, measurement noise covariance matrix R, and *a piori* covariance matrix P of the EKF. If the value of χ^2 was larger than a predefined confidence level (99.5% for this study), then the measurement was rejected and the filter update was skipped.

Chapter 4

System Control

In this chapter, the control scheme of the WSSA and the UAS platform are discussed. The control framework of the UAS had three levels. The FCU handled the lower level controls (mainly attitude dynamics) of the drone. An intermediate level controller was developed in this work for controlling the position of the drone. Additionally, a higher level controller was also developed that generated position setpoints according to the feedback from the WSSA to achieve the mission objective. This chapter will first discuss the state machine for different mission objectives. Then the state machine for the WSSA will be presented.

4.1 Mission State Machine

The developed system has two different mission objectives (mission mode). Therefore, two different mission state machines are designed: One for Sampling and the other for Mapping. The state machine varies depending on the mission objective. The mission mode is specified during startup. The main function of a mission state machine is to check the sampling flags, channel flags, depth flag, and pump flags (discussed in Section 4.2) and update the position setpoints for the intermediate controller accordingly. The takeoff and landing of the UAS is done manually. After takeoff, the autonomous mission is initiated by flicking the master switch on the RC. A plan file stored in JSON file format is provided by the user before the start of the mission. The state machine parses the Longitude, Latitude, and Altitude (LLA) coordinates of the waypoints from the **items** field of the **mission** object from the file. After that,

the workflow of the state machine for two different missions modes are as follows:

1. **Sampling:** The objective of the sampling mission is to collect three 250 mL water samples from different sampling locations. For this mission mode, the waypoints provided by the user are the sampling locations. After the start of the mission, the UAS ascends to the specified altitude and heads towards the first sampling point. If the UAS reaches within 0.5 m radius of the desired position, the first waypoint is achieved. Then the UAS starts to descend at a rate of 20 cm/sec until either the depth sensor hits the target depth or it descends below a "safe" altitude. The "safe" altitude is discussed in detail in Section 4.1.3. When the depth sensor is at the target depth, the WSSA initiates the sampling routine showed in Algorithm 1. During the sample collection, the setpoint of the intermediate controller does not change, therefore it hovers at a set altitude. After the first sample is collected, the UAS ascends to the initial altitude again and heads towards the next sampling point. Finally, after the collection of all three samples, the UAS ascends back up to the initial altitude and switches back to manual mode. The workflow of the mission state machine is depicted in detail in Figure 4.1.

2. **Mapping:** The objective of this mode is to create a fluorescence map along a transect of the waterbody while continuously pumping water through the fluorometer. For this mission mode, the user-specified waypoints form a traversing path for the UAS for fluorescence mapping. When the mission is launched, the UAS ascends to the specified altitude and heads to the first waypoint. As soon as the waypoint is achieved, the UAS gradually descends until the tube inlet is at the target depth. When the inlet is at the target depth, the WSSA continuously pumps water through the fluorometer and the fluorescence data is collected. Simultaneously, the UAS follows the mapping path and traverses

to the consecutive waypoints. During this time, the altitude setpoint remains the same as long as the tube inlet stays at the target depth. In any case, either for GNSS altitude error or drag of water, if the tube inlet exits the water surface, the altitude setpoint is decremented by 20 cm until the target depth is regained. Additionally, during the whole mission, an altitude failsafe condition is checked on a parallel thread which is described in Section 4.1.3. When the final waypoint is achieved, the UAS ascends back up to the initial altitude and switches back to manual mode.

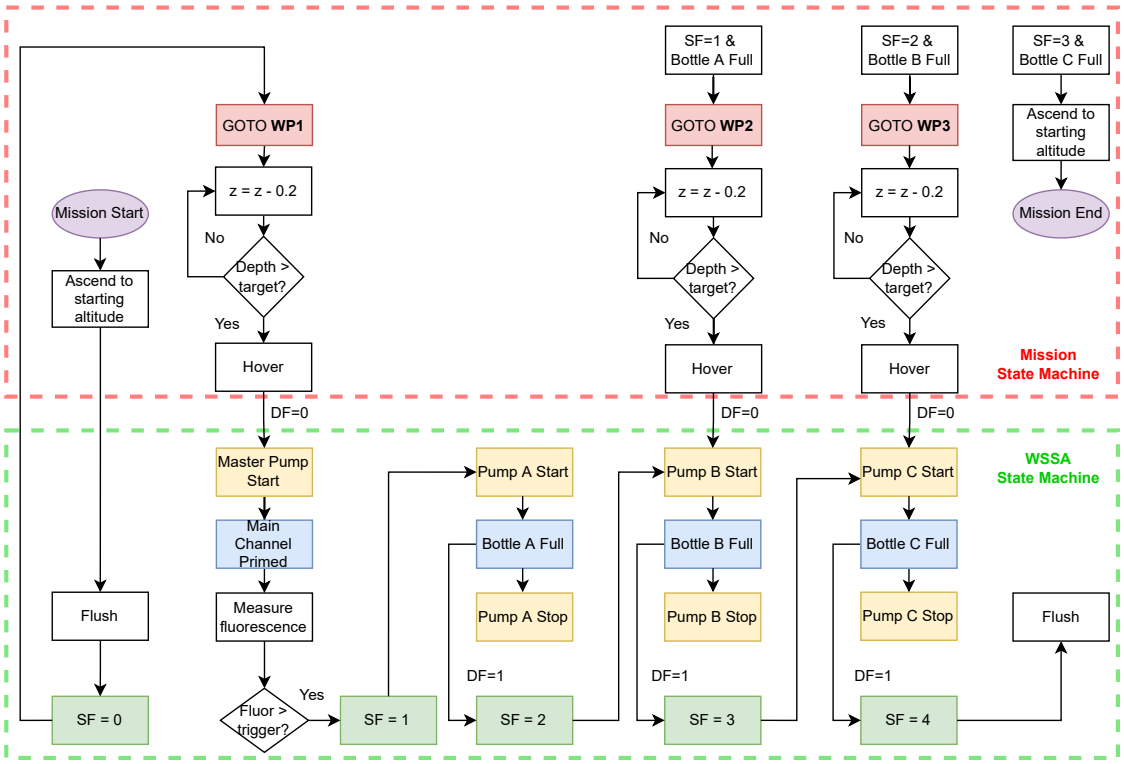


Figure 4.1 The communication between Mission State Machine and WSSA State Machine during a sampling mission. SF=Sampling Flag, DF=Depth Flag. The yellow boxes indicate the pump flags, the blue boxes indicate the channel flags, and the green boxes indicate the sampling flags. The red dashed rectangle is the workflow of the Mission State Machine, and the green dashed rectangle is the workflow of the WSSA State Machine.

4.1.1 Path Generation

For a water sampling or mapping experiment, the longitude and the latitude of the points of interest were provided by the user. However, navigation using LLA coordinates was inconvenient for this application. Therefore, a local cartesian reference frame was used to navigate to those user-defined points. The East, North, Up (ENU) coordinate frame was chosen as the reference frame. The LLA coordinates of the given points were therefore transformed into Cartesian coordinates in the ENU frame [42]. An origin for the ENU frame was defined at the start of every mission. The python class [43] for the conversion from LLA to ENU is attached in Appendix A. From this point, the user-defined coordinates will be referred to as waypoints.

To plan a path from the current position to a waypoint, the mission state machine generated a set of intermediate position setpoints for the intermediate level controller. To generate these points, first, the $2D(x, y)$ distance from the current position to the target position was divided by a set interval, d. This gave the total number of intermediate points, n. Then, n-evenly-spaced intermediate points connecting the current position and the target position with a straight line were calculated. During our experiment at Lake Wateree, the interval, d was set to 1.5 m for the sampling mission, and 0.5 m for the mapping mission. Decreasing the interval decreased the speed of follow. A slower speed during the mapping mission was desirable for better spatial resolution of fluorescence data.

4.1.2 PD Controller

An intermediate level controller controled the position of the UAS during both mission modes. A Proportional-Derivative (PD) controller was implemented as the intermediate level controller. The objective of the PD controller was to minimize the error between the current UAS position and the setpoint position. The position and orientation of the UAS as well as other avionics data from the Pixhawk FCU were

accessed utilizing the MavROS[6] package. Then the position and yaw angle error were calculated and fed to the controller.

In general, the Pixhawk FCU accepts either position, velocity, or attitude commands. In this work, the velocity commands were used as control inputs to ensure a safe velocity during the missions. The accepted command parameters are the following:

- Linear velocity along the three axes of the UAS body frame
- Yaw rate

The PD controller calculated the required velocities along the three axes and the yaw rate to minimize the position and orientation error. However, the calculated velocities were in the world frame (local ENU frame), whereas the FCU accepted control inputs in the UAS's body frame. Therefore, the velocity vector was rotated from the world frame to the body frame. These rotated inputs were the executable velocity commands to achieve the corresponding position setpoint. The Proportional and Derivative gains were carefully tuned using MATLAB's system identification and PID tuner toolbox. MAVROS integration with the PX4 autopilot software[7] was utilized to send the velocity commands to the FCU at a rate of 30 Hz. However, since a large error could result in a large velocity command, a function was in place to limit the velocity to maximum 1 m/s in all directions. An important note here is that, the PD controller achieving a zero error did not guarantee that the UAS would hover without drift. It would still move around because of the random noise in the position information from the GNSS.

[6] `http://wiki.ros.org/mavros`

[7] `https://px4.io/`

4.1.3 Altitude Failsafe

During a sampling or mapping mission, the UAS had to maintain a fixed altitude while pumping water through the WSSA. Though the PD controller did a fine job taking the drone to a position setpoint, it did not consider the error in the position data from the UAS avionics sensors. The error in altitude from a GNSS module could be as high as 5 m [40]. Therefore, completely relying on the GNSS might cause the drone to crash into the water. The fused altitude estimation from Section 3.3 came into play here. A failsafe condition was programmed to run in parallel to the mission state machine which checked the drone's proximity to the water surface. The EKF estimated the proximity to the water surface with very high accuracy within a range of 1.5 m from the water surface. The failsafe condition defined a "safe" altitude and checked whether the proximity to water was less than the "safe" altitude. If it was, then the UAS immediately ascended 2 m from the current altitude. This failsafe condition was checked five times every second. In this study, the "safe" altitude was set to 0.5 m. The performance of the failsafe is discussed in Chapter 5.

4.2 WSSA State Machine

During the early stage of development [4], a master switch from the RC remotely activated the WSSA and the UAS was manually flown to the point of interest. In this study, the master switch starts the autonomous sampling/mapping mission and enables the WSSA. The UAS autonomously flies to the first waypoint and after reaching those coordinates, the WSSA initiates the sampling/mapping. A sampling flag (SF) is assigned to distinguish among different stages of the mission. The flag values and the corresponding stages are as follows:

- **Sampling Flag 0:** The fluorescence reading is less than the preset trigger value and the tube inlet is out of the water

- **Sampling Flag 1:** The fluorescence reading is greater than the preset trigger value and the the tube inlet is at target depth.
- **Sampling Flag 2:** The first sampling bottle is full and the tube inlet is out of the water.
- **Sampling Flag 3:** The second sampling bottle is full and the tube inlet is out of the water.
- **Sampling Flag 4:** The third sampling bottle is full and the tube inlet is out of the water. End of sample collection and start of flushing routine.

Algorithm 1 WSSA State Machine

```
1: SF ← 0
2: DF ← 0
3: if depth ≥ target_depth then
4:     master_pump.start()
5:     if main_channel.is_primed then
6:         fluorometer.start()
7:     end if
8:     if SF = 0 ∧ fluorescence > trigger then
9:         SF ← 1
10:    end if
11:    for i ← 1 to 3 do
12:        if SF = i ∧ ¬bottle[i].is_full then
13:            pump[i].start()
14:        else if SF = i ∧ bottle[i].is_full then
15:            pump[i].stop()
16:            if DF = 1 then
17:                SF ← i + 1
18:                DF ← 0
19:            end if
20:        end if
21:    end for
22: else
23:    DF ← 1
24:    all_pump.stop()
25: end if
```

The sampling flags change only in the event of depth sensor losing target depth after a bottle is full. The value of a depth flag (DF) is 0 when the tube inlet is at the target depth, and its 1 when the tube inlet exits the water. Two other sets of flags are also assigned to indicate the state of each pump and each flushing line, addressed as pump flags and channel flags, respectively. Four pump flags indicate the status of each pump, with values 1 being "on" and 0 being "off". On the other hand, each of the four channel flags changes according to the feedback from the water detection sensor at each flushing line. Therefore they indicate when the main channel is primed and individual sampling bottles are full and overflowing.

Algorithm 1 shows the logical flow of the WSSA state machine during a sampling mission. For a mapping mission, the "for" loop from line 11 to line 21 is omitted, therefore, the water continuously flows through the main channel and then out of the flushing line. None of the sampling pumps are enabled during a mapping mission. Before and after every mission, a flushing routine is run to flush out any entrapped water inside the tubing circuit to prevent cross-contamination. Figure 4.1 gives an overall depiction of how the WSSA and the mission state machines work together for a sampling mission.

Chapter 5

Experiments and Results

The system was evaluated in three different outdoor locations during different stages of development: the A.C. Moore Garden Pond, Columbia, SC; Congaree River, SC and Lake Wateree, SC. The two main experiments performed at each location were: the sampling and mapping experiments. In the sampling experiment, the drone collected single or multiple samples. In the mapping experiment, the drone traversed a transect to generate a fluorescence map. The drone was deployed manually at the A.C. Moore Garden and the Congaree river. However, the experiment at Lake Wateree was fully autonomous. In addition to those, the range and accuracy of the EKF altitude estimate was tested utilizing an indoor motion capture system before deploying the system in the real world. The experimental setup and results for all the experiments are presented herewith.

5.1 EKF Validation Experiment

For validating the EKF altitude estimate against ground truth, a comparison experiment was performed with the help of an OptiTrack[8] Motion Capture system (MoCap). For this experiment, a test drone built with a DJI F550 Hexrotor frame was used. It was equipped with the same RPi 4B and sonar sensor as the main platform. This drone was used because of its smaller form factor which makes it safer for indoor testing. The same EKF algorithm was implemented on the test drone which fused the altitude measurements from the GNSS and the sonar sensor to get an optimal

[8] `https://www.optitrack.com/applications/movement-sciences//#accuracy`

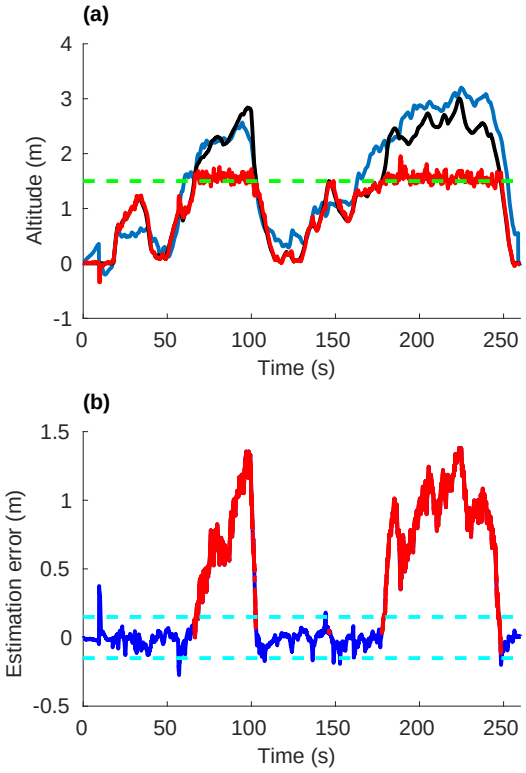


Figure 5.1 (a) shows a comparison of altitude measurement from GNSS, OptiTrack, and EKF. The GNSS altitude is shown in blue, the OptiTrack altitude is shown in black, and the EKF altitude estimate is shown in red. The horizontal green dotted line marks the 1.5 m altitude. (b) shows the error in the EKF altitude estimate with respect to the ground truth (OptiTrack). Error when OptiTrack altitude > 1.5 m is plotted in red, and the error when OptiTrack altitude < 1.5 m is plotted in blue. A ± 15 cm error bound is shown is cyan dotted lines.

altitude estimate. The test drone was flown manually for four minutes inside the MoCap. The comparison among the GNSS altitude, estimated EKF altitude, and the MoCap reported altitude are shown in Figure 5.1. Since the OptiTrack system has an accuracy of 0.2 mm, the MoCap altitude can be considered as the true altitude (ground truth). From Figure 5.1 it can be observed that within 1.5 m from the ground, the EKF altitude estimate closely follows the MoCap altitude. But when the UAS flies higher than 1.5 m, the EKF altitude estimate is not reliable anymore. It can also be seen that the error of the EKF altitude estimate is within ± 15 cm bounds when the UAS is hovering within a 1.5 m range from the surface beneath. Within

this range, the mean error is 0.17 cm and the standard deviation is 5 cm.

5.2 Water Sampling Experiment

The A.C. Moore Garden, located on the University of South Carolina campus, was selected as the initial test location as it features a water pond surrounded by various trees\plants. At this site, the drone was piloted manually to the sampling location and the WSSA was activated\deactivated with an RC.

For the sampling experiment at Moore garden, the UAS took off from an open shoreline of the pond, reached a high enough altitude, and traversed to the sampling point while the WSSA was activated. Once it reached the sampling location, the UAS started descending. When the tube-end reached the desired collection depth, the main pump was activated and started siphoning water through the main channel. When the fluorescence measurements were within the preset range the sampler was activated and a sample was collected. Finally, the WSSA was deactivated and the UAS returned to the starting point.

Figure 5.2, shows the plots of the depth, fluorescence, water temperature, and the total dissolved solids (TDS) readings during the first experiment. The whole procedure can be split into five distinct stages as follows:

I **Takeoff**: The UAS takes off and reaches a high enough altitude to keep the whole pipe suspended in the air.

II **Approach**: The user activates the WSSA as the UAS approaches the sampling point and starts descending.

III **Sensing**: The master pump activates at a set depth and starts pumping water through the main channel. The fluorescence reading starts increasing.

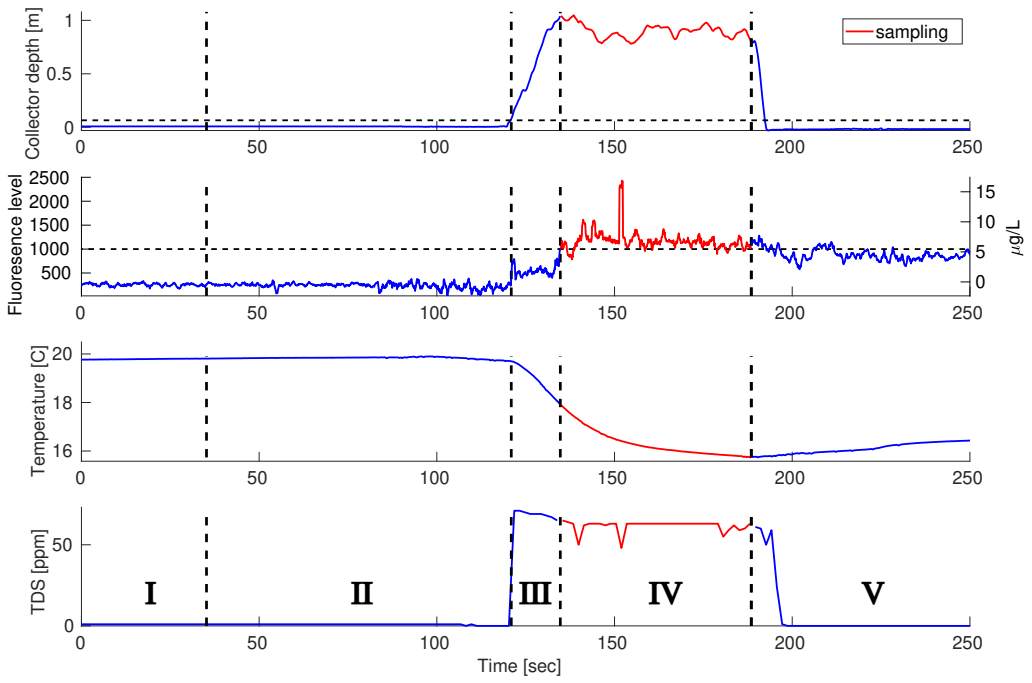


Figure 5.2 Change in depth, fluorescence level, temperature, and total dissolved solids (TDS) in the five different stages (I. Takeoff, II. Approach, III. Sensing, IV. Sampling, and V. Return) of the water sampling mission for a single sample (Moore Garden pond, Columbia, SC, USA). The fluorescence level is interpreted to $\mu g/L$ on the right vertical axis. The red portion of the graphs marks the readings of the sensors, while the sample was being collected.

IV **Sampling**: The UAS maintains altitude and a sampler pump activates when the measured fluorescence reaches a prespecified level. Water starts filling the corresponding sampling bottle.

V **Return**: When sampling is complete, the WSSA is deactivated and the UAS returns to the base.

In Figure 5.2, on the depth sensor plot, the horizontal dashed line marks the trigger depth for the master pump which is 7cm. When the inlet of the pipe reaches the triggering depth, the master pump starts pulling water through the fluorescence sensor. Until the sampling is enabled, the water continuously exits through the

flushing line. Both the temperature and TDS graphs show a change in their readings as soon as the sensors touch the water. While the TDS is rapidly increased, the temperature reading takes some time to settle to the actual temperature of the water. The change in fluorescence level can be observed as soon as the water passes through the sensor. The fluorescence level measured by the sensor does not have a unit, rather it's a value relative to a reference. It can be interpreted to $\mu g/L$ for a known reference. Once the fluorescence trigger level is reached, the sampling is enabled, allowing the water to fill the 250ml container. When the bottle is full, the flow on the flushing line increases, which acted as a visual queue that the sampling was complete. The system is then deactivated, and the UAS starts ascending, pulling the inlet out of the water.

Since the water from a reservoir like a pond is already well-mixed within 1m depth at a certain time of the year [44], the airflow produced by the drone propellers does not affect the water sample when sampling close to the surface. The duration of the flight was 168 seconds, and around 10% of the battery capacity was used. The sampling time for a single bottle was measured to be 53 seconds.

The next deployment site for the system was the bank of Congaree river near Bates Landing, SC. For this experiment, an improved WSSA with water detection sensor at the flushing lines was used. However, similar sequential stages (Takeoff, Approach, Sensing, Sampling, and Return) were followed as the experiment at Moore garden pond. The drone was piloted manually to the sampling locations, and the WSSA was activated via an RC. During this experiment, the drone collected three samples from three different locations of the river.

The third deployment site was a cove at Lake Wateree at Deer Run Road, SC. At this trial, the UAS flew autonomously to the sampling locations and collected three samples. The steps of the autonomous deployment for a sampling mission are the following:

- First, the user uses the QGround Control to select three waypoints on the waterbody corresponding to three sampling locations and generates a plan file in JSON format. Table 5.1 shows the waypoints inputs for the sampling and mapping missions during the Lake Wateree trial.

- Then, the plan file was uploaded to the onboard computer and the sampling mission was launched in the ROS framework.

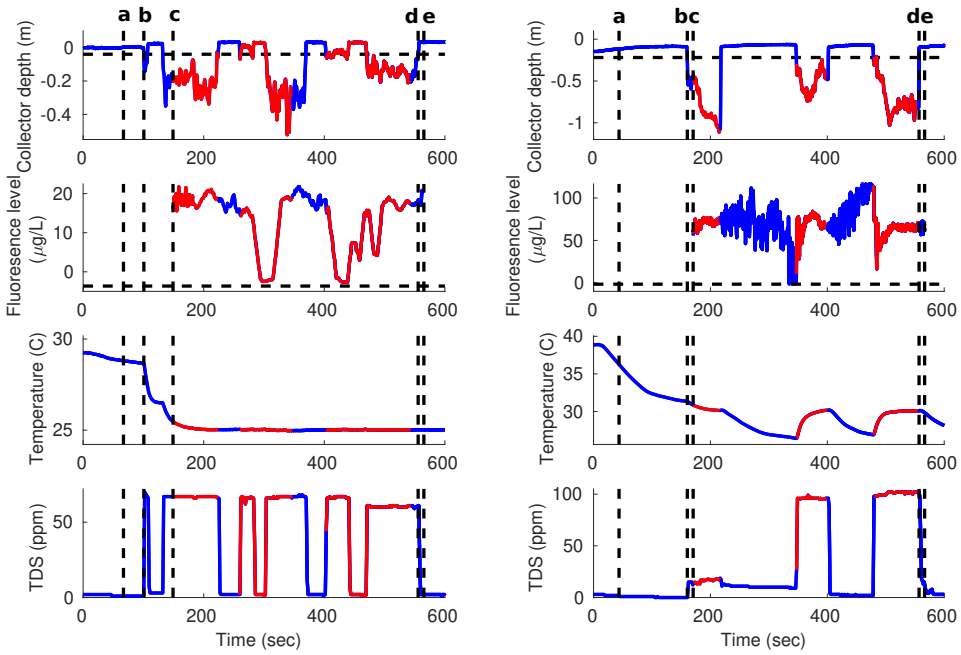
- After that, the pilot armed the UAS and took off manually from the launch site. It was flown to an open space, away from any obstruction.

- Finally, the UAS operator switched to mission mode using the RC, and the UAS autonomously collected three samples from the given waypoints.

In Figure 5.3(a), the depth, fluorescence, conductivity, and the temperature plots for the Congaree trial can be seen, and in Figure 5.3(b) the plots for the same parameters from the Lake Wateree trial are depicted. The fluorescence trigger levels for both experiments were intentionally set to a low arbitrary value. This guaranteed that a sample was collected from each site, which was the goal for this test, but the trigger could be set higher so that samples were only collected from sites with high chlorophyll content. The collection time for three samples from the Congaree trial were 74, 87, and 140 seconds, respectively. On the other hand, the collection times for three samples from the Lake Wateree trial were 46, 53, and 77 seconds, respectively. During sample collection, algae particles accumulate on the mesh grid

Table 5.1 Waypoints for Sampling and Mapping missions at Lake Wateree

Waypoints	Longitude	Latitude	Altitude(m)
Sampling			
1	34.416153	-80.862854	4.0
2	34.415792	-80.863297	4.0
3	34.416076	-80.863171	4.0
Mapping			
1	34.415425	-80.863481	4.0
2	34.415825	-80.863110	4.0
3	34.416012	-80.862935	4.0
4	34.416358	-80.862743	4.0



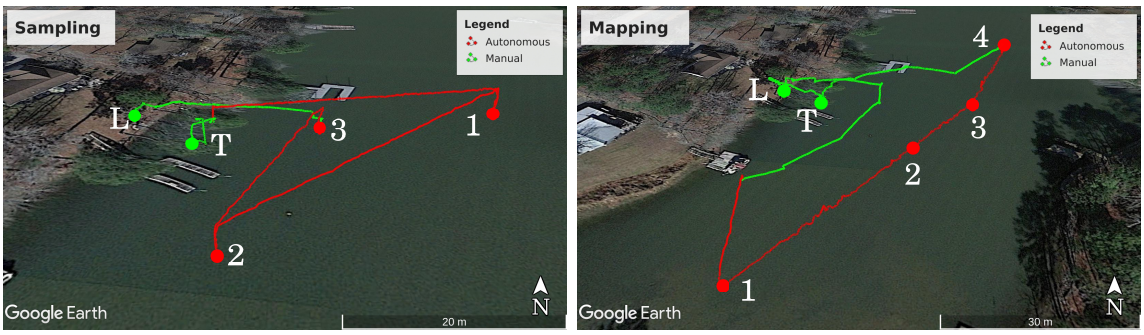
(a) Sampling at Congaree river

(b) Sampling at Lake Wateree

Figure 5.3 Depth, Fluorescence, temperature, and TDS (Total Dissolved Solids) plots from the sampling experiments at (a) Congaree river and (b) Lake Wateree. Red is when the sampling pump was on, Blue is when the sampling pump was off. The horizontal black dashed line shows the target depth and the fluorescence trigger level in their respective plots. The vertical black dashed lines annotated by a,b,c,d,e shows the time instances for mission start, master pump on, fluorescence measurement start, master pump off, and fluorescence measurement end, respectively.

at the tube end and constrict the water flow. Therefore, the collection time for the subsequent bottles in a single flight increases. From the depth plots in Figure 5.3(a) it can be seen that, during sample collection with a manual flight at the Congaree, the tube end exited the water multiple times during the sample collection. Hence it took longer to collect each sample than the Lake Wateree trial. At Lake Wateree trial, the UAS autonomously traversed a distance of 47.5 m from the mission start point to the first waypoint, then 60 m from there to the second waypoint, and finally, 35 m from the second to the final waypoint. The total flight time was 10.5 min. In Figure 5.4(a) shows the trajectory of the UAS during the autonomous sampling at

Lake Wateree taken from Google Earth.



(a) Sampling Experiment (b) Mapping Experiment

Figure 5.4 Google Earth image of (a) Sampling and (b) Mapping experiments at Lake Wateree. The manual flight path is shown in green, and the autonomous flight path is shown in red. (T = Takeoff, L = Land)

5.3 Fluorescence Mapping Experiment

In the mapping experiment at the Moore garden pond, the UAS took off, traversed to the farthest point on the left of the pond, descended to a fixed altitude, maintained that altitude and traversed slowly from left to right while the master pump kept pumping water through the main channel. The GPS coordinates and the sensor measurements along the trajectory were recorded during the traverse. Upon reaching the farthest point on right (a distance of about 6m), it traversed back to the middle, and then returned. This experiment aimed to collect *in situ* measurements along the water body of interest that can be later used in a GIS framework. The whole procedure can be split into three distinct stages as follows:

I **Takeoff and Approach**: The UAS takes off and reaches a high enough altitude to keep the whole pipe suspended in the air. The user activates the WSSA as the UAS approaches the start point of the trajectory.

II **Sensing and Traverse**: The UAS descends until the inlet hits the target depth. The master pump activates at the target depth, pumps water through the main

channel and out the flushing line. The fluorescence reading starts updating. The UAS maintains altitude and traverses along the trajectory.

III **Return**: When the trajectory is complete, the WSSA is deactivated, and the UAS returns to the base.

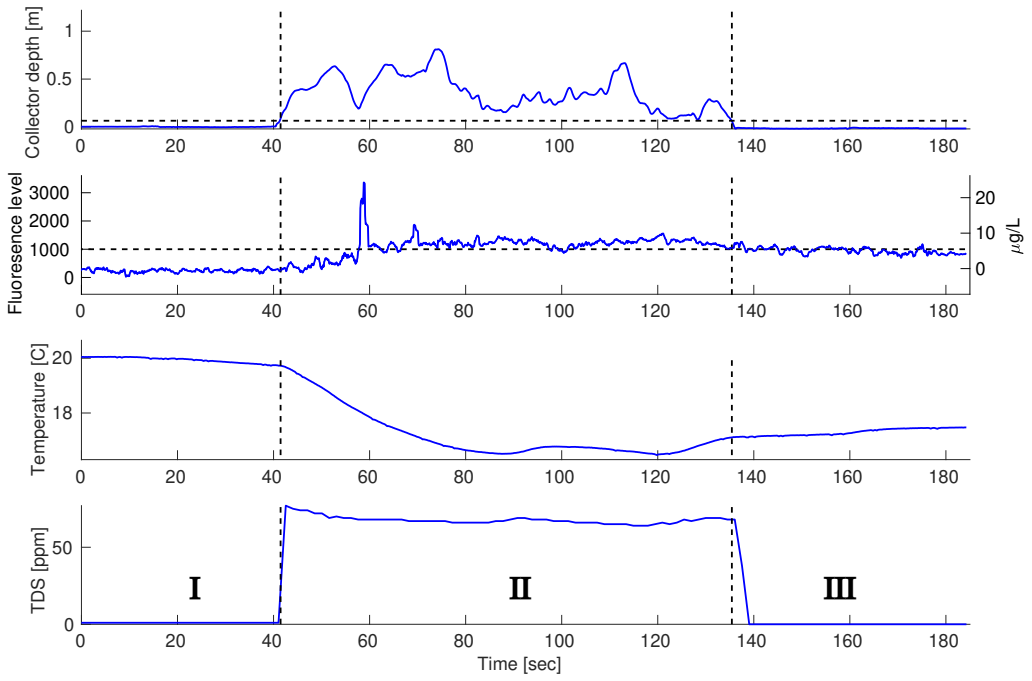
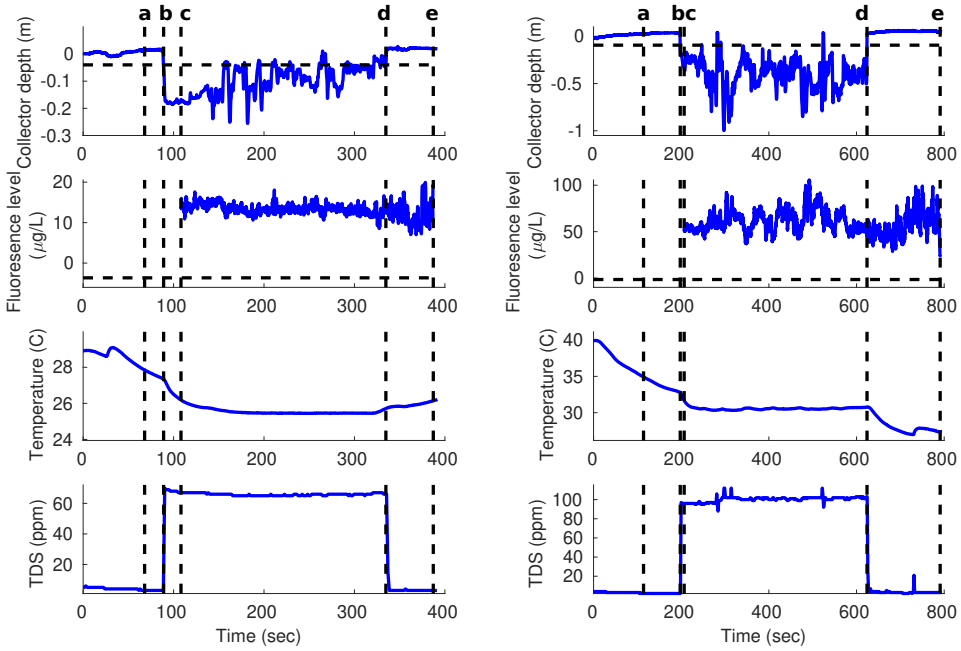


Figure 5.5 Change in depth, fluorescence level, temperature, and total dissolved solids (TDS) in the three different stages (I. Takeoff and Approach, II. Sensing and Traverse, and III. Return) of the fluorescence mapping mission (Moore Garden pond, Columbia, SC, USA). The fluorescence level is interpreted to $\mu g/L$ on the right vertical axis.

In Figure 5.5, the change in depth, fluorescence level, temperature, and TDS are shown while the drone traversed along the trajectory. The target depth was 7 cm for this experiment as well which is marked with a horizontal dashed line. The fluorescence data seems to be quite consistent across the trajectory. The temperature and TDS can be observed to have reached a steady value while traversing.



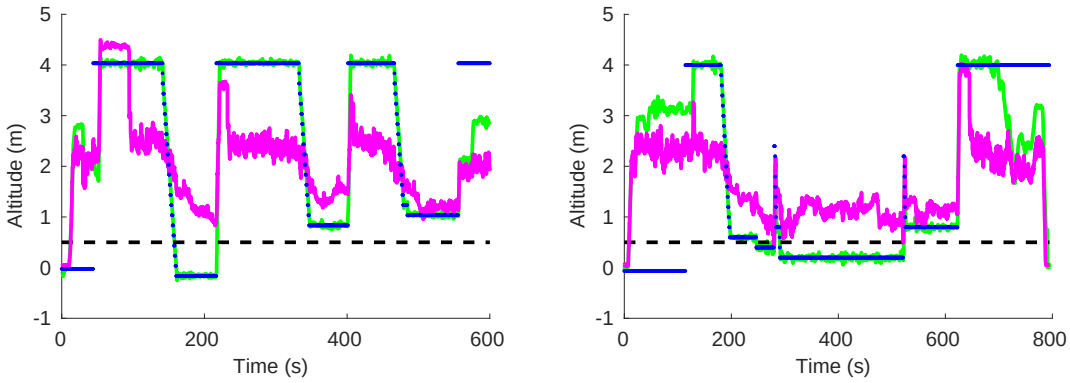
(a) Mapping at Congaree river

(b) Mapping at Lake Wateree

Figure 5.6 Depth, Fluorescence, temperature, and TDS (Total Dissolved Solids) plot from the mapping experiments at (a) Congaree river and (b) Lake Wateree. The horizontal black dashed line shows the target depth and the fluorescence trigger level in their respective plots. The vertical black dashed lines annotated by a,b,c,d,e shows the time instances for mission start, master pump on, fluorescence measurement start, master pump off, and fluorescence measurement end, respectively.

The mapping experiment at the Congaree river followed the same steps as the Moore Garden pond experiment, where the operator manually flew the UAS, ensuring the tube end stayed at the target depth and traversed along a transect of the waterbody. On the other hand, the mapping experiment performed at Lake Wateree was also fully autonomous. The steps of the autonomous trial closely followed the autonomous sampling experiment. The only difference was: the waypoints provided by the user could be more than three. A linear path from one waypoint to the next formed the trajectory to be traversed. After takeoff, as soon as the UAS switched to the mission mode, it started tracking that trajectory while submerging the tube end at the target depth. Figure 5.6(a) and Figure 5.6(b) shows the depth, fluorescence,

conductivity, and temperature for the mapping experiment in Congaree River and Lake Wateree, respectively. The mean EKF altitude estimate during the transect mapping ($time$ = 200 s to 600 s) was 1.1 m and the standard deviation was 0.2 m. The UAS traversed and mapped a 127 m long path on the lake in 10.5 min. In Figure 5.4(b) shows the trajectory of the UAS during the autonomous mapping at Lake Wateree taken from Google Earth.



(a) Sampling Experiment (b) Mapping Experiment

Figure 5.7 Failsafe performance during autonomous (a) Sampling and (b) Mapping operations. The EKF altitude estimate is shown in magenta, the GNSS altitude is shown in green, and the altitude setpoints are shown in blue. The 'safe' altitude is shown by horizontal black dashed line.

5.4 Failsafe Performance during autonomous Operation

During the autonomous operation, a risky situation arises when the UAS descends and hovers close to the water during both experiments. Since the UAS is programmed to keep descending until the depth sensor reaches the target depth, in some instances this could take the UAS dangerously close to the water surface if the tube is not hanging straight down from the UAS. There is also the random GNSS drift for which the UAS may gradually drift closer to the water surface while hovering for a longer period of time. The failsafe condition prevents the UAS from drifting too close to the water and resets its altitude if the proximity from the water surface is less than 0.5

m. From the altitude plots from Figure 5.7(a), it can be seen that the EKF altitude estimate did not cross the failsafe altitude during sampling. But during mapping, as seen in Figure 5.7(b), it went below the failsafe altitude twice at 280 s and 521.5 s. Both times, the altitude setpoint was increased by 2 m to avoid danger. At the first instance, it took about 0.6 s for the drone to respond and ascend above the failsafe altitude, and for the second instance, it took about 0.3 s. Point to be noted that, the PD controller tries to minimize the error in the GNSS altitude compared to the setpoint altitude. For the GNSS altitude and setpoint altitude, the zero line does not necessarily correspond to the water surface level. Therefore, the setpoint altitude and the GNSS altitude may appear to be below the "safe" altitude even though the UAS is actually well above it.

Chapter 6

Conclusion and Future Work

This thesis discusses the development of a novel UAS-based, sensor-triggered, non-destructive sample collection system with autonomous navigation capabilities. The proposed system was proved to be effective in collection of three samples under 10 minutes in a fully autonomous flight. It could also autonomously traverse and map the fluorescence of a 127 m path on water. One of the biggest challenges of this work was keeping the drone safe from water as the UAS must hover close to the water surface for sampling or mapping. This work approached this problem by implementing sensor fusion with an Extended Kalman Filter for reliable altitude estimation near the water surface. The accuracy of the estimated altitude with this technique was 15 cm within 1.5 m from the water surface. Using this altitude estimate, the designed altitude failsafe was successfully triggered each time the UAS descended below the "safe" altitude.

One of the limitations of the proposed system is that the position of the UAS was prone to drift. This is because the PD controller takes the GNSS positioning as input for position control, rather than the EKF altitude estimate. Therefore, the random drift in GNSS consequently affects the position of the UAS. Due to this position drift, the tube inlet's depth was inconsistent during sample collection or mapping. In some water bodies, chlorophyll contents may have high correlation with depth. Therefore, development of a drift-free altitude controller is one of the major scope of improvement for this work.

Another limitation of the system is that the best working range of the EKF alti-

tude estimate being only 1.5 m from the water surface. A robust and accurate altitude estimation technique with a wider working range is required to develop a drift-free controller to maintain a consistent hovering altitude. Designing an adaptive covariance Extended Kalman Filter and fusing proximity sensors with different working ranges could solve this problem. Using vibration tolerant sonar sensors could also improve the accuracy and range of the altitude estimate. Another possible solution would be using real-time kinematic (RTK) positioning supported GNSS onboard the UAS. Typical accuracy in the vertical domain using RTK is about 0.02 m to 0.03 m (i.e. using the South Carolina Real Time Network). This solution would enable us to control the absolute position of the drone and, thus, the tube in the water.

Another noticeable problem was that when submerged in water, the tube tended to exit the water due to drag when the UAS traversed at high speed. Therefore, the max velocity of the UAS was restricted to 1 m/s and the interval for the intermediate setpoints was also set to 0.5 m for a slow-paced follow for the mapping mission at Lake Wateree. The average horizontal velocity of the UAS was 0.3 m/s during the mapping, while the UAS can fly at up to 15 m/s. The slow pace of the UAS cost a lot of unnecessary battery life, which was a downside. The future work will include determining the optimal velocity for the autonomous mission while ensuring the submersion of the tube end.

Bibliography

[1] F. D. Wilde et al. "National field manual for the collection of water-quality data: Field measurements." In: *U.S. Department of the Interior, U.S. Geological Survey* (1998).

[2] Caitlyn M. English et al. "Chlorophyll Fluorometer for Intelligent Water Sampling Using a Small Uncrewed Aircraft System (sUAS)". In: *Applied Spectroscopy* (2022). PMID: 36065908. DOI: 10.1177/00037028221126748. eprint: https://doi.org/10.1177/00037028221126748. URL: https://doi.org/10.1177/00037028221126748.

[3] Zechariah B. Kitzhaber et al. "Fluorometer Control and Readout Using an Arduino Nano 33 BLE Sense Board". In: *Applied Spectroscopy* (2022). PMID: 36197285. DOI: 10.1177/00037028221128800. eprint: https://doi.org/10.1177/00037028221128800. URL: https://doi.org/10.1177/00037028221128800.

[4] Kazi Ragib Ishraq Sanim et al. "Development of an Aerial Drone System for Water Analysis and Sampling". In: *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*. Dubrovnik, Croatia: IEEE, June 2022, pp. 1601–1607. DOI: 10.1109/icuas54217.2022.9836122.

[5] Michael E. Hodgson et al. "Mission Planning for Low Altitude Aerial Drones during Water Sampling". In: *Drones* 6.8 (2022). ISSN: 2504-446X. DOI: 10.3390/drones6080209. URL: https://www.mdpi.com/2504-446X/6/8/209.

[6] Sabrina Carroll, Michail Kalaitzakis, and Nikolaos Vitzilaios. "UAS Sensor Deployment and Retrieval to the Underside of Structures". In: *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*. 2021, pp. 895–900. DOI: 10.1109/ICUAS51884.2021.9476737.

[7] Michail Kalaitzakis et al. "A marsupial robotic system for surveying and inspection of freshwater ecosystems". In: *Journal of Field Robotics* 38.1 (2021), pp. 121–138. DOI: https://doi.org/10.1002/rob.21957. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.21957. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21957.

[8] Alejandro Suarez et al. “Cartesian aerial manipulator with compliant arm”. In: *Applied Sciences (Switzerland)* 11 (3 2021), pp. 1–20. ISSN: 20763417. DOI: 10.3390/app11031001.

[9] Mercedes Vélez-Nicolás et al. “Applications of unmanned aerial systems (UASs) in hydrology: A review”. In: *Remote Sensing* 13.7 (2021). ISSN: 20724292. DOI: 10.3390/rs13071359.

[10] Marc Schwarzbach et al. “Helicopter UAV systems for in situ measurements and sensor placement”. In: *2012 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, July 2012, pp. 4766–4769. ISBN: 978-1-4673-1159-5. DOI: 10.1109/IGARSS.2012.6352548. URL: http://ieeexplore.ieee.org/document/6352548/.

[11] Marc Schwarzbach et al. “Remote water sampling using flying robots”. In: *2014 International Conference on Unmanned Aircraft Systems, ICUAS 2014 - Conference Proceedings*. IEEE, May 2014, pp. 72–76. ISBN: 9781479923762. DOI: 10.1109/ICUAS.2014.6842240. URL: http://ieeexplore.ieee.org/document/6842240/.

[12] Dylan Cornell, Maryann Herman, and Fernando Ontiveros. “Use of a UAV for Water Sampling to Assist Remote Sensing of Bacterial Flora in Freshwater Environments”. In: *Undergraduate External Publications* (2016). URL: https://fisherpub.sjfc.edu/undergraduate_ext_pub/17/.

[13] Hideyuki Doi et al. “Water sampling for environmental DNA surveys by using an unmanned aerial vehicle”. In: *Limnology and Oceanography: Methods* 15.11 (2017), pp. 939–944. ISSN: 15415856. DOI: 10.1002/lom3.10214.

[14] Akihiko Terada et al. “Water sampling using a drone at Yugama crater lake, Kusatsu-Shirane volcano, Japan”. In: *Earth, Planets and Space* 70.1 (Dec. 2018), p. 64. ISSN: 18805981. DOI: 10.1186/s40623-018-0835-3. URL: https://earth-planets-space.springeropen.com/articles/10.1186/s40623-018-0835-3.

[15] Bikram Pratap Banerjee et al. “Development of a UAV-mounted system for remotely collecting mine water samples”. In: *International Journal of Mining, Reclamation and Environment* 34.6 (2020), pp. 385–396. ISSN: 17480949. DOI: 10.1080/17480930.2018.1549526. URL: https://doi.org/10.1080/17480930.2018.1549526.

[16] C.T. Graham et al. “Drones can reliably, accurately and with high levels of precision, collect large volume water samples and physio-chemical data from lakes”. In: *Science of The Total Environment* 824 (June 2022), p. 153875.

ISSN: 00489697. DOI: 10.1016/j.scitotenv.2022.153875. URL: https://linkinghub.elsevier.com/retrieve/pii/S0048969722009676.

[17] José C. D. Neto et al. "An Adapted Unmanned Aerial Vehicle for Environmental Water Sampling". In: *Química Nova* 45.Quím. Nova, 2022 45(6) (June 2022). ISSN: 0100-4042. DOI: 10.21577/0100-4042.20170892. URL: https://doi.org/10.21577/0100-4042.20170892.

[18] Ryan A Horricks et al. "Comparison of drone and vessel-based collection of microbiological water samples in marine environments". In: *Environmental Monitoring and Assessment* 194 (6 2022), p. 439. ISSN: 1573-2959. DOI: 10.1007/s10661-022-10095-8. URL: https://doi.org/10.1007/s10661-022-10095-8.

[19] André Tristany Farinha et al. "Off-shore and underwater sampling of aquatic environments with the aerial-aquatic drone MEDUSA". In: *Frontiers in Environmental Science* 10 (2022). ISSN: 2296-665X. DOI: 10.3389/fenvs.2022.1023269. URL: https://www.frontiersin.org/articles/10.3389/fenvs.2022.1023269.

[20] *How drone water sampling enables safer, inexpensive testing that exceeds federal EPA guidelines*. June 2021. URL: https://enterprise-insights.dji.com/blog/drone-water-sampling-nixie.

[21] *Drone Based Remote Water Sampling |UgCS Integrated Systems*. WEBSITE. URL: https://integrated.ugcs.com/drone-based-remote-water-sampling (visited on 02/10/2023).

[22] Cengiz Koparan et al. "In situ water quality measurements using an unmanned aerial vehicle (UAV) system". In: *Water (Switzerland)* 10.3 (Mar. 2018), p. 264. ISSN: 20734441. DOI: 10.3390/w10030264. URL: https://www.mdpi.com/2073-4441/10/3/264.

[23] Michaella Chung et al. "Obtaining the Thermal Structure of Lakes from the Air". In: *Water* 7.11 (Nov. 2015), pp. 6467–6482. ISSN: 2073-4441. DOI: 10.3390/w7116467. URL: http://www.mdpi.com/2073-4441/7/11/6467.

[24] Anthony Demario et al. "Water plume temperature measurements by an unmanned aerial system (UAS)". In: *Sensors (Switzerland)* 17.2 (2017). ISSN: 14248220. DOI: 10.3390/s17020306.

[25] John Paul Ore et al. "Autonomous aerial water sampling". In: *Journal of Field Robotics* 32.8 (Dec. 2015), pp. 1095–1113. ISSN: 15564967. DOI: 10.1002/rob.21591. URL: http://doi.wiley.com/10.1002/rob.21591.

[26] Devin Castendyk et al. “Aerial drones used to sample pit lake water reduce costs and improve safety”. In: *Mining Engineering* 69.7 (2017), pp. 20–28. ISSN: 00265187.

[27] Devin Castendyk et al. “Using aerial drones to select sample depths in pit lakes”. In: *Proceedings of the 13th International Conference on Mine Closure*. Australian Centre for Geomechanics, Perth, 2019, pp. 1113–1126. DOI: 10.36487/acg_rep/1915_89_castendyk. URL: https://papers.acg.uwa.edu.au/p/1915_89_Castendyk/.

[28] Devin Castendyk, Jimmy Voorhis, and Bradley Kucera. “A Validated Method for Pit Lake Water Sampling Using Aerial Drones and Sampling Devices”. In: *Mine Water and the Environment* 39.3 (Sept. 2020), pp. 440–454. ISSN: 16161068. DOI: 10.1007/s10230-020-00673-y. URL: http://link.springer.com/10.1007/s10230-020-00673-y.

[29] Cengiz Koparan et al. “Autonomous in situ measurements of noncontaminant water quality indicators and sample collection with a UAV”. In: *Water (Switzerland)* 11.3 (2019). ISSN: 20734441. DOI: 10.3390/w11030604.

[30] Balasubramanian Esakki et al. “Design of amphibious vehicle for unmanned mission in water quality monitoring using internet of things”. In: *Sensors (Switzerland)* 18.10 (2018). ISSN: 14248220. DOI: 10.3390/s18103318.

[31] Keunyea Song et al. “Using unmanned aerial vehicles to sample aquatic ecosystems”. In: *Limnology and Oceanography: Methods* 15.12 (Dec. 2017), pp. 1021–1030. ISSN: 15415856. DOI: 10.1002/lom3.10222. URL: http://doi.wiley.com/10.1002/lom3.10222.

[32] James Benson et al. “Microorganisms Collected from the Surface of Freshwater Lakes Using a Drone Water Sampling System (DOWSE)”. In: *Water* 11.1 (2019). ISSN: 2073-4441. DOI: 10.3390/w11010157. URL: https://www.mdpi.com/2073-4441/11/1/157.

[33] *Nero Water Sampler - Aerial drone sampling at its best.* WEBSITE. URL: https://nerosampler.com/ (visited on 02/10/2023).

[34] *Drone Water Sampling and Monitoring.* WEBSITE. URL: https://store.swellpro.com/pages/water-sampling-drone (visited on 02/10/2023).

[35] Cengiz Koparan et al. “Adaptive water sampling device for aerial robots”. In: *Drones* 4.1 (Feb. 2020), pp. 1–16. ISSN: 2504446X. DOI: 10.3390/drones4010005. URL: https://www.mdpi.com/2504-446X/4/1/5.

[36] Libe Washburn et al. “Water sampling from aerial drones for water quality research in coastal and inland waters”. In: *2018 Ocean Sciences Meeting* (2018).

[37] Daniele Pinton, Alberto Canestrelli, and Luca Fantuzzi. “A UAV-based dye-tracking technique to measure surface velocities over tidal channels and salt marshes”. In: *Journal of Marine Science and Engineering* 8.5 (2020). ISSN: 20771312. DOI: 10.3390/JMSE8050364.

[38] Digvijay Singh et al. “DAWSSM: A plug-and-play Drone Assisted Water Sampling and Sensing Module”. In: *IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society*. 2021, pp. 1–6. DOI: 10.1109/IECON48115.2021.9589543.

[39] John Paul Ore and Carrick Detweiler. “Sensing water properties at precise depths from the air”. In: *Journal of Field Robotics* 35.8 (2018), pp. 1205–1221. ISSN: 15564967. DOI: 10.1002/rob.21807.

[40] FAA William J. Hughes Technical Center. *Global Positioning System Standard Positioning Service Performance Analysis Report*. Tech. rep. January. 2022, p. 21. URL: http://www.gps.gov/technical/ps/2008-SPS-performance-standard.pdf.

[41] Bo Jhang Hoy et al. “From pressure to path: Barometer-based vehicle tracking”. In: *BuildSys 2015 - Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built* (Nov. 2015), pp. 65–74. DOI: 10.1145/2821650.2821665.

[42] NAL Research. “Datum Transformations of GPS Positions: Application Note”. In: *U-blox ag* (1999), p. 12. URL: http://www.u-blox.ch.

[43] Michail Kalaitzakis. *A python class to convert GPS coordinates to a local ENU coordinate system and vice versa*. 2019. URL: https://gist.github.com/MikeK4y/1d99b93f806e7d535021b15afd5bb04f.

[44] Zheng Yu et al. “Effects of water stratification and mixing on microbial community structure in a subtropical deep reservoir”. In: *Scientific Reports* 4 (2014), pp. 1–7. ISSN: 20452322. DOI: 10.1038/srep05821.

Appendix A

conversion from LLA to ENU coordinates

```
1  import numpy as np
2
3  class GPS_utils:
4    '''
5      Contains the algorithms to convert a gps signal (longitude,
       latitude, height)
6      to a local cartesian ENU system and vice versa
7
8      Use setENUorigin(lat, lon, height) to set the local ENU
       coordinate system origin
9      Use geo2enu(lat, lon, height) to get the position in the local
       ENU system
10     Use enu2geo(x_enu, y_enu, z_enu) to get the latitude, longitude
       and height
11   '''
12
13   def __init__(self):
14     # Geodetic System WGS 84 axes
15     self.a  = 6378137.0
16     self.b  = 6356752.314245
17     self.a2 = self.a * self.a
18     self.b2 = self.b * self.b
19     self.e2 = 1.0 - (self.b2 / self.a2)      # This is e_squared = (
       a2-b2)/a2
20     self.e  = self.e2 / (1.0 - self.e2)      # This is
```

```
    e_prime_squared = (a2-b2)/b2
21
22      # Local ENU Origin
23      self.latZero = None
24      self.lonZero = None
25      self.hgtZero = None
26      self.xZero = None
27      self.yZero = None
28      self.zZero = None
29      self.R = np.asmatrix(np.eye(3))
30
31  def setENUorigin(self, lat, lon, height):
32      # Save origin lat, lon, height
33      self.latZero = lat
34      self.lonZero = lon
35      self.hgtZero = height
36
37      # Get origin ECEF X,Y,Z
38      origin = self.geo2ecef(self.latZero, self.lonZero, self.hgtZero)
39      self.xZero = origin.item(0)
40      self.yZero = origin.item(1)
41      self.zZero = origin.item(2)
42      self.oZero = np.array([[self.xZero], [self.yZero], [self.zZero
    ]])
43
44      # Build rotation matrix
45      phi = np.deg2rad(self.latZero)
46      lmd = np.deg2rad(self.lonZero)
47
48      cPhi = np.cos(phi)
49      cLmd = np.cos(lmd)
50      sPhi = np.sin(phi)
```

```
51      sLmd = np.sin(lmd)
52
53      self.R[0, 0] = -sLmd
54      self.R[0, 1] =  cLmd
55      self.R[0, 2] =  0.0
56      self.R[1, 0] = -sPhi * cLmd
57      self.R[1, 1] = -sPhi * sLmd
58      self.R[1, 2] =  cPhi
59      self.R[2, 0] =  cPhi * cLmd
60      self.R[2, 1] =  cPhi * sLmd
61      self.R[2, 2] =  sPhi
62
63    def geo2ecef(self, lat, lon, height):
64      phi = np.deg2rad(lat)
65      lmd = np.deg2rad(lon)
66
67      cPhi = np.cos(phi)
68      cLmd = np.cos(lmd)
69      sPhi = np.sin(phi)
70      sLmd = np.sin(lmd)
71
72      N = self.a / np.sqrt(1.0 - self.e2 * sPhi * sPhi)
73
74      x = (N + height) * cPhi * cLmd
75      y = (N + height) * cPhi * sLmd
76      z = ((self.b2 / self.a2) * N + height) * sPhi
77
78      return np.array([[x], [y], [z]])
79
80    def ecef2enu(self, x, y, z):
81      ecef = np.array([[x], [y], [z]])
82
83      return self.R * (ecef - self.oZero)
```

```
84
85  def geo2enu(self, lat, lon, height):
86    ecef = self.geo2ecef(lat, lon, height)
87
88    return self.ecef2enu(ecef.item(0), ecef.item(1), ecef.item(2))
89
90  # By using the closed formula set of ECEF to LLA conversion.
91  # https://www.scribd.com/doc/85548161/Datum-Transformations-of-GPS
     -Positions
92  # https://en.wikipedia.org/wiki/Geographic_coordinate_conversion
93  def ecef2geo(self, x, y, z):
94    p = np.sqrt(x*x + y*y)
95    q = np.arctan2(self.a * z, self.b * p)
96
97    sq = np.sin(q)
98    cq = np.cos(q)
99
100   sq3 = sq * sq * sq
101   cq3 = cq * cq * cq
102
103   phi = np.arctan2(z + self.e * self.b * sq3, p - self.e2 * self.a
      * cq3)
104   lmd = np.arctan2(y, x)
105   v = self.a / np.sqrt(1.0 - self.e2 * np.sin(phi) * np.sin(phi))
106
107   lat = np.rad2deg(phi)
108   lon = np.rad2deg(lmd)
109   h = (p / np.cos(phi)) - v
110
111   return np.array([[lat], [lon], [h]])
112
113 def enu2ecef(self, x, y, z):
114   lmd = np.deg2rad(self.latZero)
```

```
115     phi = np.deg2rad(self.lonZero)
116
117     cPhi = np.cos(phi)
118     cLmd = np.cos(lmd)
119     sPhi = np.sin(phi)
120     sLmd = np.sin(lmd)
121
122     N = self.a / np.sqrt(1.0 - self.e2 * sLmd * sLmd)
123
124     x0 = (self.hgtZero + N) * cLmd * cPhi
125     y0 = (self.hgtZero + N) * cLmd * sPhi
126     z0 = (self.hgtZero + (1.0 - self.e2) * N) * sLmd
127
128     xd = -sPhi * x - cPhi * sLmd * y + cLmd * cPhi * z
129     yd =  cPhi * x - sPhi * sLmd * y + cLmd * sPhi * z
130     zd =  cLmd * y + sLmd * z
131
132     return np.array([[x0+xd], [y0+yd], [z0+zd]])
133
134   def enu2geo(self, x, y, z):
135     ecef = self.enu2ecef(x, y, z)
136
137     return self.ecef2geo(ecef.item(0), ecef.item(1), ecef.item(2))
```