Spring 4-24-2014

Autonomous Aerial Water Sampling

John-Paul W. Ore
University of Nebraska-Lincoln, jore@cse.unl.edu

AUTONOMOUS AERIAL WATER SAMPLING

by

John-Paul W. C. Ore

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfilment of Requirements

For the Degree of Master of Science

Major: Computer Science

Under the Supervision of Carrick Detweiler and Matthew B. Dwyer

Lincoln, Nebraska

May, 2014

AUTONOMOUS AERIAL WATER SAMPLING

John-Paul W. C. Ore, M. S.

University of Nebraska, 2014

Advisers: Carrick Detweiler and Matthew B. Dwyer

Obtaining spatially separated, high frequency water samples from rivers and lakes is critical to enhance our understanding and effective management of fresh water resources. In this thesis we present an aerial water sampler and verify the system in field experiments. The aerial water sampler has the potential to vastly increase the speed and range at which scientists obtain water samples while reducing cost and effort. The water sampling system includes: 1) a mechanism to capture three 20 *ml* samples per mission; 2) sensors and algorithms for safe navigation and altitude approximation over water; and 3) software components that integrate and analyze sensor data, control the vehicle, and drive the sampling mechanism. In this thesis we validate the system in the lab, characterize key sensors, and present results of outdoor experiments. We compare water samples from local lakes obtained by our system to samples obtained by traditional sampling techniques. We find that nearly all water properties are consistent between the two techniques. These experiments show that despite the challenges associated with flying precisely over water, it is possible to quickly obtain water samples with an Unmanned Aerial Vehicle (UAV).

COPYRIGHT

DEDICATION

To Charles William and Constance Louise, my parents. To my ancestors Ray and Hilda, Hans and Agnes. To my sisters and their families: Heidi, Jon, Janna, Todd, Zoie, Kira, Fiona, and Ursula.

ACKNOWLEDGMENTS

I would like to thank our limnologist and environmental engineering partners Dr. Michael Hamilton, Dr. Amy Burgin, and Dr. Sally Thompson, for their continuous support of these efforts. Deep thanks and gratitude to Dr. Sebastian Elbaum for his insightful suggestions and encouragement and Dr. Witawas Srisa-an for his helpful feedback. I would also like to acknowledge the valued assistance of Baoliang Zhao, Hengle Jiang, Jacob Greenwood, Adam Taylor, Dave Anthony, Jared Ostdiek, Christa Webber, Emily Waring, Dr. Seth McNeil, and the NIMBUS Lab.

GRANT INFORMATION

This work was partially supported by USDA-NIAF #2013-67021-20947, AFOSR #FA9550-10-1-0406, NSF IIS-1116221. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of these agencies.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Water quality varies due to the spatial distribution of water transport pathways and contaminant source areas. Characterizing this large-scale variability remains a critical bottleneck that inhibits understanding of transport processes and the development of effective management plans to address water quality issues. In the US, it is estimated that human-induced degradation of fresh water sources annually costs over $2.2 billion, but the full extent of the cost is poorly known due to insufficient data [2]. In addition to the economic costs there is a huge human toll. As reported by the United Nations Environment Program in 2010: "Over half of the world's hospitals beds are occupied with people suffering from illnesses linked with contaminated water and more people die as a result of polluted water than are killed by all forms of violence including wars." [3] World-wide, water borne diseases cause the death of 1.5 million under-five children every year [4][3].

Increased water sampling, both more frequently and at more locations, could help identify the source of unhealthy water before it causes widespread illness and also it could help identify the causes of water degradation. Since one of the main barriers to increased sampling is the cost of human labor to collect and analyze samples, automating

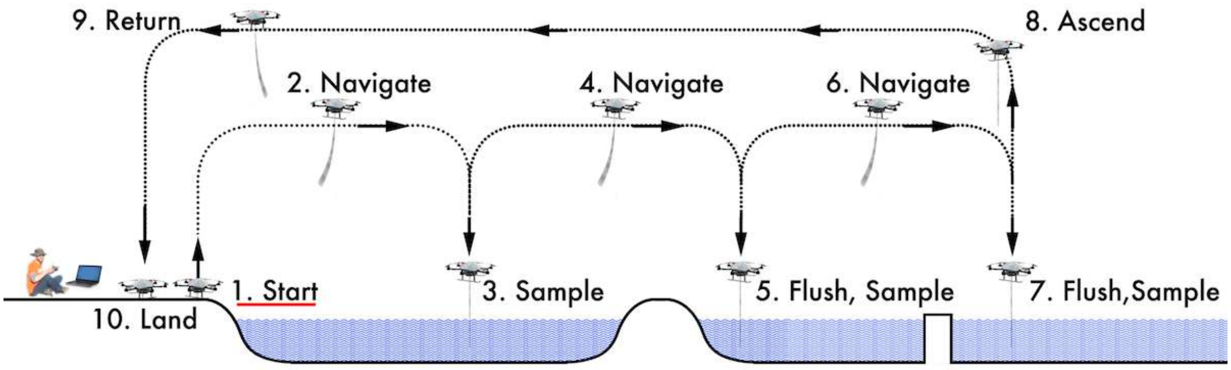


Figure 1.1: Simple Overview of Proposed Method.

water sample collection could increase the total productivity of that human labor, and thereby reduce sickness and ecological damage. In addition to minimizing environmental and human costs, water sampling can benefit basic water science by increasing the spatiotemporal resolution of datasets.

Current water sampling techniques are often based on grab sampling (e.g. dipping a bottle off the side of a kayak) [5], statically deployed collection systems [6], or using mobile sensors affixed to Autonomous Surface Vehicles (ASVs) [7] or Autonomous Underwater Vehicles (AUVs) [8]. Most autonomous systems are used on large, open water features such as seas, large lakes and rivers, and sample for long duration, in deep or distant places, with high quality. All of these methods are relatively slow, spatially restricted, costly, or difficult to deploy; none sample quickly at multiple locations while overcoming barriers, such as dams or land.

Another approach is to come at the problem from the air. Recent advances in sensors, materials, and control systems have yielded a new class of flying devices, the micro Unmanned Aerial Vehicles (micro-UAVs). These flying robots are computer-controlled, lightweight, commercially available, and can carry small payloads ($< 750\ g$) for up to 20 minutes. Fortunately, a UAV's limited payload is not a critical shortcoming because water samples do not have to be very large (20 $ml = 20\ g$) to be scientifically useful. Also, the UAV's battery-limited flights allow it to travel nearly a kilometer and back, which is close enough for many water sampling applications.

Obtaining water samples from a UAV, however, poses challenges that must be addressed before these systems can be deployed in the wild. These challenges include operating a UAV in the field at some distance under computer control, where it can be difficult to assess the state of the system. All the challenges of operating in an unstructured environment are amplified by flying over water at low altitude. Small wind gusts, which at higher altitudes are not problematic, can cause the vehicle to land in the water and sink. The main challenge is that we must get close enough to the water to touch it only as much as we want, without getting so close that we endanger the UAV. Addressing these challenges is a major contribution of this work.

1.1 Requirements

Through discussions with our hydrologist partners we derived a set of requirements for the aerial water sampler:

1. It must capture at least three 20 ml water samples at predefined locations within 1 km.
2. It must be light and small enough to be carried by a single scientist.
3. It samples autonomously once target locations are identified.
4. It must be reliable and safe to reduce cost and risk, since these are the primary barriers for adoption.
5. It must be cost effective.
6. It must not influence water properties.

Requirement 1: Three 20 *ml* water samples

Water scientists collect as little at 15 *ml* or as much at 10 *L* at a single location, depending on the purpose and the property under investigation. However, most chemical and biological properties (discussed in detail in Sec. 7.5) can be measured with as little as 20 *ml*. Water scientists seek samples which are representative and cover large areas of interest, and therefore they choose multiple locations separated by many meters. Our limnologist collaborators usually consult a map or inspect the general location before determining where to sample. They sample in a pattern which depends on the application. For example, they might choose a sample location close to shore and another far away to measure the spatial variability. The scientists know these locations before water sampling commences, and the locations and sample timings are chosen as part of a larger scientific campaign. The locations at which the scientist takes samples are usually recorded with current hand-held GPS technologies, accurate to several meters. Therefore any aerial water sampling system would have to collect at least 20 *ml* from several locations to be useful.

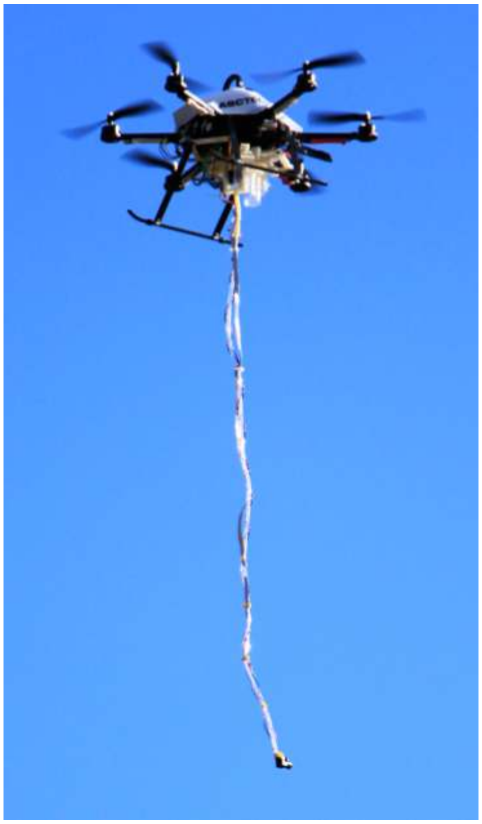


Figure 1.2: UAV-Based Water Sampling.

Requirement 2: Carried by a single scientist

Limnologists study networks of connected water systems. To study these large systems, they occasionally travel hundreds of kilometers per day to collect samples. Since a single scientist can accomplish almost every water sampling protocol by themselves, any new tool that requires multiple people is less feasible. Further, not all water sampling locations are accessible by car, and the water scientist must hike or wade a distance before deciding where to sample. Therefore the system must be portable and deployable by a single scientist.

Requirement 3: Non-Expert Operation

It is not realistic to expect every field limnologist to become an expert pilot of flying robots. Therefore an aerial water sampling system must be able to fly itself to the sample location, find its own way to the right sampling height without human intervention, return with the samples and land with an expert pilot. Separate from the flight of the vehicle, the system must interact with the scientist without requiring the modification of low-level programming files. The system must eventually be deployable without expert knowledge in robotics, software, or flying a UAV.

Requirement 4: Reliable and Safe

Since water scientists work in the field with resource and time constraints. Every tool they use must be dependable, in part because they might work alone in remote areas where help and spare parts might be far away. Therefore an aerial sampling system must be very unlikely to cause serious bodily harm or property damage.

Requirement 5: Cost Effective

The total system must be available for a reasonable price. In order for aerial water sampling to become a viable option, the total system must be available for a reasonable cost. Scientific campaigns are limited by budgetary constraints, and the portion of the budget devoted to equipment for water sampling must be small enough that the remaining budget accomplishes the scientific goals. Therefore the system, both the commercial-off-the-shelf aerial platform, the water sampling mechanism, and the labor required to produce is must be cost effective.

Requirement 6: Not influence water properties

Critically, an aerial water sampler, when compared to existing methods, must provide equivalent scientific results. If the system introduces biases because of the method of sampling, then it will be nearly impossible to adopt because it will be incompatible with existing datasets. Therefore the system should be able to replace existing techniques seamlessly.

1.2 Contributions

This work builds on our previous efforts on water sampling [9][10]. The contributions of this work include:

1. Developing a UAV-based system that autonomously obtains three 20 *ml* water samples per flight.
2. Integrating and characterizing sensors on the UAV to enable reliable, low-altitude hover (1.0 *m*) over water.
3. Testing the system both indoors in a motion-capture room as well as in the field at lakes and waterways.
4. Validating that key water chemical properties are not biased by using a UAV-based mechanism.

We also identify a number of outstanding challenges to be addressed in future work, such as determining the impact of waves, winds, and flowing water on altitude control. Our contributions are commensurate to the requirements, and these requirements together identify the territory we address in this work. The technical challenges, hurdles, tests, and lessons learned, all seek to create a tool for water scientists to collect better data sets.

Much of this work builds on recent advancements in mobile aerial robotics. Small, reliable, GPS-enabled devices are now commercially available with sufficient payloads to carry more robust sensors, which are also getting smaller and lighter. These UAVs, combined with rapid-prototyping equipment such as 3-D printing and component-based open source software platforms, allow system engineers to try out designs quickly, and learn from their flaws and successes. By using an iterative design process, we seek to engineer a system which expands the possible.

1.3 Thesis Overview

This document describes the overall method and introduces some of the challenges (Chap. 1), related work including small robotic submarines, robotic boats, and sensor networks (Chap. 2), potential UAV aerial water sampling applications (Chap. 3), details of the electromechanical design (Chap. 4) including the sampling mechanism, sensors and custom hardware to control it, software architecture for both the onboard and offboard control systems (Chap. 5), altitude experimental results, sampler effectiveness experimental results (Chap. 6), results from field experiments (Chap. 7), in which we demonstrate the viability of the system in the field, and some final thoughts on future work (Chap. 8).

Chapter 2

Related Work

Increasingly, environmental scientists use robots to collect data [11]. Whether on land, in water, or in the air, affordable robots are 'revolutionizing' spatial ecology [12]. Scientists use robots for mapping habitat [13], detecting chemical plumes [14], tracking marine fauna [15], data muling, adaptive sampling, measuring crop height, and taking water samples [10]. Although there are many examples of robots being used to gather data for scientists, there are very few examples of using autonomous aerial platforms to collect physical specimens. Therefore, as we consider related work we examine previous efforts in one of two ways: either an autonomous vehicle is used to take samples in aquatic environments or a UAV is flown at low-altitude. We treat first the former and then the latter.

2.1 Water Sampling Vehicles

Autonomous vehicles used in water sampling are either Autonomous Surface Vehicles (ASVs) or Autonomous Underwater Vehicles (AUVs), both deployed in water features such as oceans or large lakes.

In open freshwater, there are numerous examples of sensing water properties from the surface [16] [17] [18]. One recent example is Dunbabin *et al.*'s [7] Lake Wivenhoe ASV navigates throughout complex inland waterways while measuring a range of water quality properties and greenhouse gas emissions. The purpose of this vehicle is to operate autonomously for long periods of time while continuously monitoring water properties. It carries with it instruments to detect qualities of the environment at that point and does not return or capture water samples. It operates autonomously, with a laser scanner and camera to detect and avoid obstacles in the environment. The initial deployment was for a large inland lake (Lake Wivenhoe) from which the city of Brisbane, Australia draws drinking water, and therefore this water body is under continual observation. The system, deployed on a 16 ft catamaran, can be moved to another body of water by trailering the ASV and towing it to a new location. It cannot be easily deployed by a single scientist, unlike our system. Our missions are intended for flexible, ad hoc, water sampling which require flexible deployment, unlike Dr. Dunbabin's system designed for continual observation.

Moving underwater, there are many examples of AUVs for mapping [19] [20] [21], environmental monitoring [22] [23] [24] [25] [26], and hull inspection using algorithms like SLAM [27]. Cruz *et al.*'s [8] [28] MARES AUV dives up to 100 m deep to monitor pollution, collect data, capture video, or follow the seabed. These systems are designed for autonomous operation with little operator intervention, because the nature of water prevents high-bandwidth communication. Our system, since it travels in the air, can provide constant feedback of its status and a synopsis of its sensor readings. Underwater systems, including Dr. Cruz's AUV can be transported from one water body to another using a wheeled trailer. However, they are difficult for one person to deploy and require improved docks or landings to get them into or out of the water. Our system does not require there to be any existing improvements to the entry and exit infrastructure of the water body. Underwater vehicles and systems are good for long-duration sampling in deep or distant places. In contrast, our system can be carried in a backpack and quickly deployed to sample multiple disconnected water features from a single launch site. Further, *in situ* sampling cannot yet measure all desired water properties such as the presence of suspended solids, petroleum hydrocarbons (benzene, pesticides, chlorinated solvents), pathogens, organic carbon, heavy metals, as identified by Erickson *et al.* [6].

A different approach to water sampling is to instrument the environment [29] [16]. If you know you are going to be sampling in the same way at the same location for a long time, then it could be more efficient to invest in long-term sensors. Instead of putting sensors onto robots that move freely in the environment, Rahimi *et al.*'s [30] networked infomechanical systems (NIMS) system adds semi-mobile sensors to the environment. Scientists deploy these cable-based systems by fastening a metal cable between two trees (or other fixed objects) and the mechanical shuttles move along the cable to gather data at different locations. This provides adaptive sampling by actuating the sensors to move toward interesting information within the confined area. Once installed, these networks provide a fast, flexible way to collect datasets over that particular area of interest for long durations. Similarly, the Jefferson Project at Lake George [31], New York, aims to conduct an extensive aerial Light Detection and Ranging (LIDAR) survey and bathymetric survey before instrumenting Lake George and making it the world's "smartest lake." Lake George is large, surface connected, and the surrounding region reaps billions of tourist dollars. Likewise, Dr. Michael Hamilton oversees the Very Large Ecological Array (VeLEA) [32] installed over thousands of hectares at the Blue Oak Ranch Reserve, part of the University of California's reserve system, and only one of several sensor networks in the UC reverse system [33].

In contrast, our method assumes no prior instrumentation of the environment, and can quickly be deployed and redeployed to address the need to collect data.

2.2 Control Systems and System States

Our control system implements a hybrid control system, in which the vehicle is controlled using a hybrid of continuous controllers responding to continuous input, and discrete controllers reacting to discrete events. Continuous events include changes in position and attitude (or rotation), and discrete events include abstractions from sensors such as 'the pump is in water'. Our control architecture is based on methods proposed by Koo *et al.* [34]. Our discrete controller is implemented as a state machine, and we use timers and guards on certain transitions (pumping and flushing) in the manner of an Extended State Machine (ESM) proposed by Merz *et al.* in [35], which are rooted in Harel's Statecharts formalisms [36]. Unlike Gillula and Tomlin's work using reachability to analyze transitions between discrete states [37], we simplify our design by assuming the UAV's built-in controller can transition from any control input to any control input. Other UAV control systems related to our efforts include Merz *et al.* [38], who show low-altitude flight techniques in rural areas for remote sensing. In a river exploration context, Scherer *et al.* limit the maximum velocity based on effective laser ranges [39]. We likewise limit descent velocity as the vehicle moves towards the water since the sensors are effective only when the water is in range. In contrast, our focus is low-altitude flight over water but does not include obstacle avoidance.

2.3 Information-based Exploration

Amigoni *et al.* use robots to map an environment [40] using information acquired during the mapping to update the path taken by the robot. Likewise, the AquaNodes sensor network [41] discovers the locations of nearby sensor nodes and moves to maximize the total information gained by the network. Stachniss and Burgard identify a method for exploring the unknown regions of maps [42]. These approaches assume that the robot senses information during exploration relevant to the scientific campaign. We do not implement information-based exploration since the information contained in the water samples is unknown until after the samples have been analyzed in a lab. If our system had additional sensors, like salinity sensors, we could use this information to guide where we sample water. We intend to explore this approach in future work.

2.4 Altitude Estimation

Other recent efforts for UAV height estimation include miniature radar altimeters and optical flow altitude estimation as summarized by Kendoul [43]. Radar is an attractive method, since water is an excellent dielectric and therefore has a clear radar reflection. However, the lightest commercially available radar altimeters weigh 300 g, [44], within the payload capacity but heavy for a micro UAV. More importantly for our application, radar altimeters are accurate to only $\pm$ 0.5 m, which is good for higher altitude flight, but below the requirements of our system's near-water requirements. Nuske *et al.* and Scherer *et al.* navigate over water with a UAV using stereo video with inertial sensors (IMUs) and off-axis spinning LIDAR [45] [39]. They find altitude by searching for specular highlights in the LIDAR returns, and find it to be accurate to within centimeters over calm water, which is similar to our system. This approach builds on work by Achar *et al.* who presents methods for segmenting water from land in images of river scenes [46]. Their system is built for 3-D awareness and mapping, and a large portion of the payload is consumed by sensors, whereas our system is not aware of the 3-D environment around it, except for a small portion directly below the vehicle. They assume a clear boundary at the riverbank, whereas we assume a somewhat uncluttered environment within sight of the co-ecologist. Further, many optical methods are perturbed by changing light conditions, so instead we chose ultrasonic sensors.

2.5 Cable Suspended Loads

Our system flies with a small dangling pump. Since a flexible tube connects this mass, it can be thought of as a slung load with single-point suspension. Single point slung load dynamics were explored in the 1970s by Poll and Cromack [47] and more recently for small rotorcraft by Bernard [48]. Also in the 1970s, Dahl demonstrated how cable-suspended loads behave like non-linear damped oscillators [49]. We help ensure that the swinging tube reaches a nearly stable state by arresting translation above the target sample location, and descending to the water, which gives time for the tube to stop swinging. More recently, Sreenath *et al.* [50] explores the flight dynamics of cable-suspended loads on quadcopters, for cables both taut and slack, but does not explore the effects of a load which changes mass over time, which can happen when you are taking on the mass of water. Faust *et al.* demonstrated a machine learning technique to find swing free trajectories for rotorcraft with cable-suspended loads [51], a problem that Palunko *et al.* approached through dynamic programming [52]. We avoid these considerations and dynamics entirely by hanging a sufficiently small mass, such that the forces incurred are negligible relative to those generated by our UAV and thus we can ignore the load with regard to the flight dynamics.

2.6 Water Landing

Landing in the water was an attractive option since waterproofing the UAV would save us the worry of flying near water. Large-scale floatplanes and seaplanes fly and land in the water regularly, and in 2002 Pisanich demonstrated a two-motor fixed-wing UAV capable of water landing and takeoff [53]. More recently, Meadows *et al.* demonstrated a solarpowered micro-UAV seaplane “Flying Fish” capable of persistent ocean monitoring [54]. These systems do not have landing gear, and hence must land in the water, which would be a barrier for field scientists looking to quickly deploy, sample, and recover the system. Further, they assume open water free of debris (like unstructured clutter in post-flooding scenarios). Among rotorcraft, the Aquacopter [55] and QuadH2O [56] UAVs land in and takes off from calm water. In spite of the amphibious options, we do not adopt these platforms or land in the water because: 1) fast-moving water or waves might make it impossible to take off (some users report that waves invert the vehicle, which is then helpless) [57]; 2) the sampling mechanism and battery enclosure would have to be complete sealed, thereby decreasing the efficiency of swapping vials or batteries; and 3) radio strength attenuates near the water’s surface and we want the UAV and base station in constant contact.

2.7 Kalman Filter

Forms of the Kalman Filter [58] are used extensively in robotic applications, especially the Extended Kalman Filter (EKF) [59]. In aviation and aerial robotics, the EKF is often used to estimate the attitude (pose) of the vehicle, since this is a non-linear state estimation problem [60] [61] [62]. We utilize the simpler, linear form called a plain Kalman Filter since we’re estimating one variable (altitude) for which linear assumptions hold. Further discussion of our implementation of the plain Kalman Filter is in Sec. 6.1.3.

2.8 Low-Altitude with Human Experts

Our work most resembles the low-altitude UAV presented by Göktoğan *et al.* [63], depicted in Fig. 2.1 wherein the authors surveil and spray aquatic weeds at low altitude using a RUAV ("rotary UAV", a model helicopter). Like our work, Göktoğan's work seeks to enable environmental scientists and land managers who monitor water systems. Unlike our system, this RUAV measures altitude with a laser altimeter. Their system extends the perception of human experts, who can decide remotely how to address the relayed information. Ours, on the other hand, extends the actuation of humans who decide where to collect information without having to go there themselves to get it. Both systems require a human backup pilot. Our work similarly does not address global planning and requires a human expert to decide where to perform tasks (weed experts in Göktoğan's case and limnologists in ours). Our work differs from this in that we use ultrasonic with pressure for altitude, and we retrieve a liquid rather than depositing it, although both systems envision a repeated cycle of sorties with tightly integrated human collaboration. Another key difference is that their aerial system does not interact with external objects during flight other than by 'throwing' a liquid overboard, whereas we touch and interact with the environment while flying.

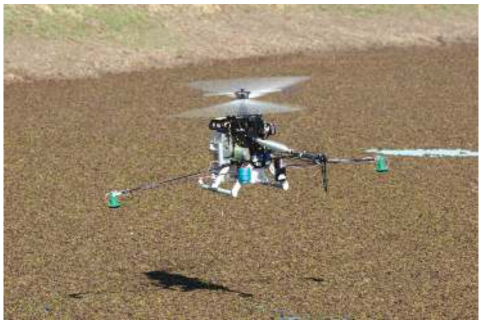


Figure 2.1: Weed Spraying UAV [63]

2.9 Summary

Increasingly, scientists use UAVs to collect data, but to date most UAV applications have been limited in scope to taking pictures from the sky. However, aerial mobile manipulation, which involves approaching a problem from the air and getting close enough to touch it, is emerging as a promising way to extend the usefulness of aerial platforms. The current work aims at demonstrating the possibility that simple sensors and actuators, together with a commercially available UAV, can be used to help scientists today. The related work addresses challenges in mapping, situational awareness, flight dynamics, and platform hardening, which we reduce to the simpler problem of practical co-robotics for field scientists, where the human and the robotic system work together in the field.

Chapter 3

UAV Water Sampling Applications

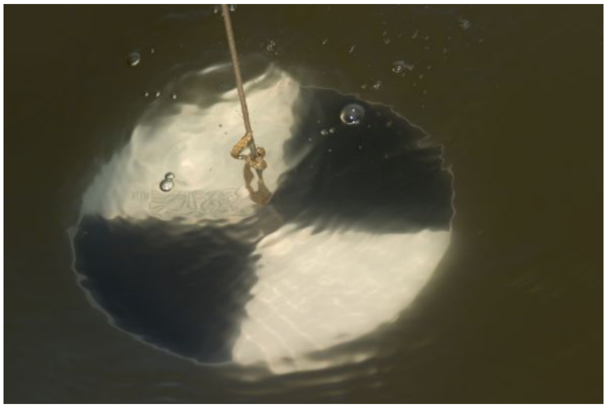


Figure 3.1: Secci Disk for measure water opacity. Image courtesy of White Lake Stewardship.

Water sampling has applications in both basic science and environmental modeling. Since water is critical for health and agriculture, a long history of tools and techniques have been used and some, like the Secchi disk shown in Fig. 3.1, have been used since 1865 up to the present. Despite all the technological advances in the past 150 years, limnologists still rely in part on dipping a jar into the water, a technique called 'grab sampling,' shown in Fig. 3.2. Table 3.1 shows a summary of current applications, their sample size requirements, frequency, and spatial distribution. This chapter provides an overview of the uses for water sampling, and why aerial water sampling will be an invaluable tool for scientists and civil engineers.

3.1 Basic Science: Limnology

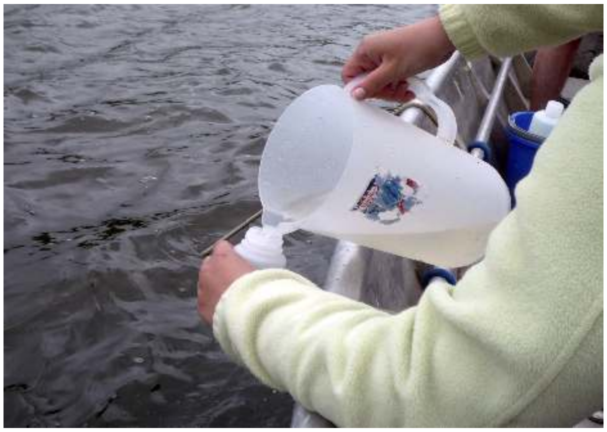


Figure 3.2: Grab Sampling

Useful properties can be measured *in situ*, but require a literal boatload of equipment, used to measure temperature, conductivity, pH, dissolved oxygen, light, turbidity, and Secchi transparency. However, limnologists and hydro-chemists still require water samples for lab analysis, because lab equipment is much more sensitive. They measure chemical properties of surface water, including phosphate, total phosphorus, nitrate, nitrite, nitrogen, and ammonia, as well as biological properties, such as the presence of toxic microcystins. All of these field measurements, along with lab analysis, together present much of the canonical data through which surface water phenomenon are understood [66]. By facilitating data collection, lightweight UAVs, together with our collaborators, will improve, if not revolutionize spatial ecology [12]. We see applications of UAV-based water sampling in two areas: 1) increasing the ease of capturing routine small samples from disconnected water features; and 2) improving the quality of event-based datasets by increasing spatial and temporal resolution.

Table 3.1: Water Sampling Applications Summary

Task	Sample Size	Frequency	Spatial Domain
Limnology	15 *ml*-1 *L*	variable	local, regional, and global
Env. Monitoring	$15\ ml - 5\ L$ [5]	variable	surface and ground water
Oil Spills	$30 - 50\ ml$ [64]	month-years	surface water
Disease Tracking	$10 - 100\ ml$ [65]	once	open wells, rainwater collection systems
eDNA	15 *ml*-10 *L* [1]	once, few	Lagoons, Rivers, Streams, Lakes

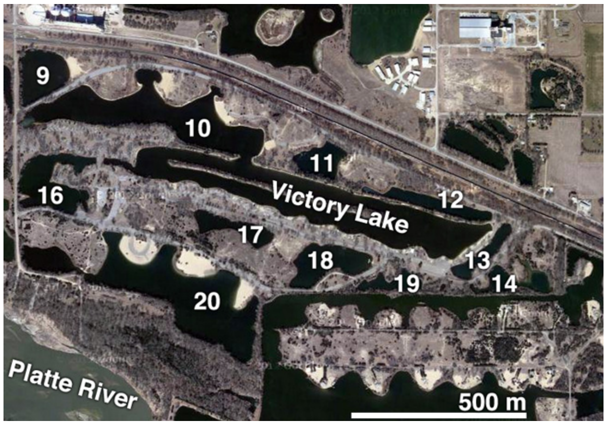


Figure 3.3: Fremont Sandpit lakes

For example, our collaborators study the Fremont Sandpit lakes, shown in Fig. 3.3. Each numbered lake is groundwater connected, surface water disconnected, chemically distinct, and must be sampled separately. Scientists study these lakes to better understand the causes of toxic algal blooms, specifically toxic microcystins. From 2004 to 2006, Fremont Lake 20 suffered an algal bloom and was closed because of the public health hazard [67]. To address this problem, water resource managers applied alum to the water, which bonded with phosphorus and removed the microcystins' source of food. Since that time, water scientists routinely monitor the water chemistry of the Fremont lakes. Currently, a team of three scientists tow a boat to the lake, launch the boat, navigate to the sample location, collect samples and take measurements, return to dock, get the truck, put the boat back on the trailer, and drive to the next lake. Each of 10-15 lakes is sampled in this manner over a long 10-15 hour day. But in just two hours, one scientist with our UAV-system could sample all these lakes, enabling the possibility of capturing data with unprecedented spatiotemporal resolution.

3.2 Environmental Monitoring

Environmental managers collect routine water samples to monitor the quality of ecosystems and human-drinking water sources. In contrast to Limnology, environmental monitoring usually happens at the same locations asking the same question over long periods of time. Since this water monitoring is routine, there is a greater incentive to invest in static, mechanized samplers. We do not expect aerial water sampling to entirely displace static sampling, but rather to augment the set of monitoring tools.

During floods, high waters can make it unsafe for humans to collect water samples. In recent flooding of the Platte River in Nebraska, USA, water scientists removed their automated sampling equipment because of the danger posed by high waters to expensive equipment. During this flooding, public health managers and scientists could not gather data to make informed decisions. Aerial water sampling can help collect data when it is not safe or too expensive to apply traditional methods.

Many interesting scientific questions are event driven. A water event, like a heavy rain after a season of drought, presents a tremendous amount of potential data simultaneously. However, it is difficult and usually not feasible to instrument the entire landscape, or to deploy sensors where you predict the weather will cause interesting data. Therefore, ad hoc water sampling, where the location and scope of the sampling are not known in advance, is an especially compelling case for aerial water sampling, because it can be delivered and deployed within hours and collect many more samples than a human in the same amount of time.

When deploying a sensor network, a critical decision is where to situate the expensive, sensitive instruments to gather the most useful information. This decision is often made by domain experts with experience, or by making a limited initial survey of the unknown area. UAV aerial water sampling would be useful to scientists who want to make a quick, high-level survey to increase the likelihood of deploying long-term sensors in the best locations.

3.3 Disaster Response

Disaster response can be difficult because infrastructure might suddenly be absent. In a newly unstructured area, obtaining water samples can be difficult because transportation pathways are unavailable. The kinds of disasters in which water sampling might be useful fall into two broad categories: floods and hazardous conditions. In the case of floods, water sampling is important to assess the quantity of pollutants or to test for the presence of biological hazards (diseases). After Haiti's 2010 earthquake, a cholera outbreak sickened hundreds of thousands of people and killed thousands [68]. Inadequate water infrastructure and poor distribution of clean drinking water were major factors, as well as an inability to quickly locate disease-carrying water sources. By utilizing fast-capture airborne water-samplers, aide workers with personnel and budgetary constraints might be able to pinpoint disease vectors more quickly. Finding and addressing the source of the disease vector reduces the effective contact rate and thereby reduces the overall transmission risk.

During the response to the Fukushima Daiichi nuclear disaster, responders flew micro UAVs into the reactor buildings first with video cameras and later with radioactive sensors [69] [70]. However, these system could only look or sense and could not retrieve a water sample to help determine what portions of the radiation came from iodine-131, caesium-134, and caesium-137. Subsequent ground robots collected water samples to determine whether the water at the site came from inside the pressure vessel [71]. Since ground robots cannot always enter wrecked buildings, UAV aerial water sampling will likely be an important tool when quickly retrieving a physical specimen informs critical decisions.

In flooding, it is not usually dangerous for a human to obtain water samples, but there is so much ad-hoc sampling required that it is unfeasible to use human power. In the case of atomic, biological, or chemical hazards, using humans might be effective but incurs unacceptable risk. UAVs can bypass ground rubble, sample efficiently over a large area, and minimize human exposure to toxins.

3.4 eDNA

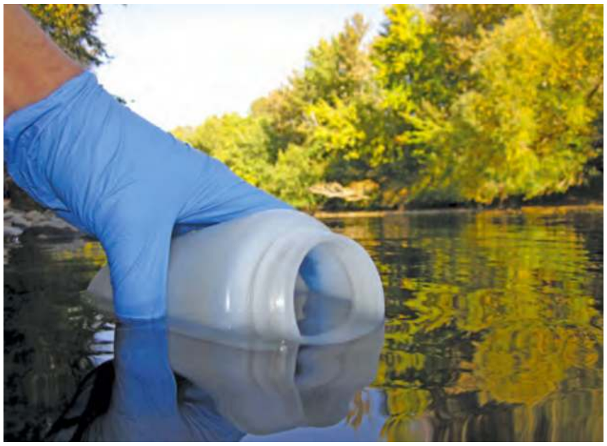


Figure 3.4: Collecting water for eDNA - Photograph by Matthew Laramie [1].

Environmental DNA, also known as 'eDNA', is the set of genetic material found free in the environment. These free DNA strands can be used to identify the presence of a species without directly contacting the organism, as first demonstrated in 2008 by Ficetola *et al.* [72], based on techniques ("shotgun sequencing") developed by Venter *et al.* [73]. Ficetola took 15 *ml* water samples from several wetlands in France and extracted DNA strands from an elusive frog, demonstrating that "the environment retains the molecular imprint of inhabiting species." To find the species, scientists amplify the DNA in the sample, sequence it, and pattern match the eDNA against 'DNA barcode' databases, which contain distinguishing sequences from many species. Environmental DNA monitoring has also been used to delimit the invasion front of two species of Asian carp near Chicago, Illinois, USA [74]. Although different eDNA protocols require up to 10 *L* of water, some require as little as 15 *ml* of water, within the capabilities of our system. Although Thomsen *et al.* identify the need for further study of field methods and lab protocols for eDNA, they suggest that eDNA is a very compelling conservation tool because "there is an urgent need for data-driven prioritization of conservation actions [75]." As shown in Fig. 3.4, current eDNA gathering techniques include hand-sampling. This technique for eDNA could be augmented by aerial sampling, provided the collected samples are then stored temporarily in a cooler before the eDNA is concentrated through filtration or centrifugation [1].

3.5 Summary

Aerial water sampling has a wide range of potential applications, from pure science to humanitarian endeavors. Some applications, like eDNA, are only beginning to reveal their usefulness. Other applications, like environmental monitoring, have been practiced for decades, but rely on manual techniques, like grab sampling. Our system focuses on the applications in which the physical act of sampling is a primary barrier to widespread adoption of routine water sampling.

Chapter 4

Electromechanical Design

4.1 Overview

The water-sampling system is composed of several electrical and mechanical subsystems. A visual overview of these subsystems and their relation to the UAV is show in Fig. 4.1. Each major subsystem discussed in this chapter is identified in this picture. This chapter contains details about the physical subsystems, while a detailed discussion of the software system can be found in Chap. 5.

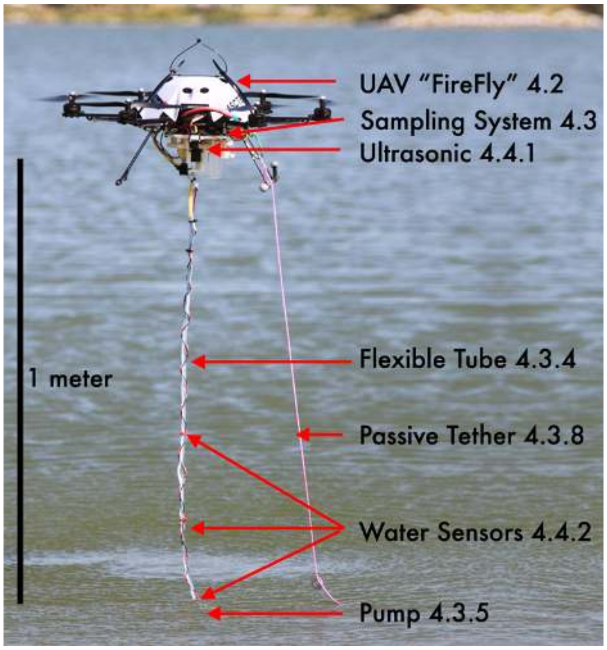


Figure 4.1: Water sampling system with chapter labels.

As discussed in the introduction (Sec. 1.1), our limnologist colleagues helped derive a set of requirements which guided the design of the system. These requirements included: 1) collect three 20 *ml* samples; 2) system must be carried by a single scientist; 3) usable by a non-expert; 4) reliable; 5) cost effective; and 6) not bias water properties. These requirements guided the design of the electro-mechanical subsystem of the aerial water sampling system. The contributions of this work addresses these requirements through:

- Finding an appropriate UAV base vehicle
- Building a software control system, discussed in Chap. 5
- Designing, building, and testing a custom embedded microcontroller with radio
- Building a breakaway mechanism to reduce risk to the UAV
- Identifying, characterizing, and configuring sensors for near water flight
- Choosing a suitable servo and micro-pump
- Selecting cost-effective components
- Designing a process by which non-experts use the system
- Isolating the samples to prevent cross-contamination
- Determining how and where to store water on the UAV
- Adopting an iterative, rapid-prototyping methodology

Additional requirements not addressed in the current work include a simple user interface for scientists to use and endurance and robustness to work in any climate. We chose to address first the core functionality of the system and save secondary requirements for future work after characterizing the system in the field.

This chapter includes a discussion of the UAV base vehicle (Sec. 4.2), the water sampling subsystem, including the frame or 'chassis' (Sec. 4.3.1), the vials and lids which hold the water in transit (Sec. 4.3.2), the servo which directs the flow of water into the vials (Sec. 4.3.3), the micro-pump which pushes the water up the tube (Sec 4.3.5), the design which allows water to be jettisoned overboard to flush the system (Sec. 4.3.6), the mechanism allowing the pump and tube to 'breakaway' from the UAV (Sec. 4.3.7), the use of a tether during outdoor flight (Sec. 4.3.8), the custom-built embedded system to control the water sampler (Sec. 4.3.9), the ultrasonic sensors for near-water altitude estimation (Sec. 4.4.1), and finally the water conductivity sensors adorning the dangling tube which sense water (Sec. 4.4.2).

4.2 Aerial Vehicle

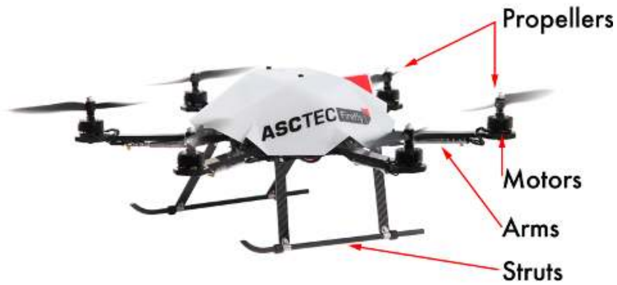


Figure 4.2: AscTec Firefly.

We chose to build our water sampler onto an Ascending Technologies Firefly UAV [76], shown in Fig. 4.2. The Firefly is a hexrotor with a maximum payload of 600 g, of which we use 300 g when loaded with three full water vials. It comes equipped with GPS (Global Positioning System), 3-axis accelerometers and gyroscopes, compass and an air pressure sensor. This UAV communicates with a human backup pilot using a radio link, and has two 2.4 GHz 802.15.4 radios for remote autonomous control and sensor feedback. The UAV is powered by an 11.1 V 2400 mAh lithium polymer battery. With a fully charged battery, the vehicle can fly for 15-20 minutes, which bounds the maximum mission distance at approximately 2 km. Total battery power depends on the mission and also the ambient temperature. To comply with local regulations regarding UAVs [1] , we fly outdoors with a passive string tether connected to the frame of the vehicle and wrangled by a human operator. In practice, the tether limits the distance the UAV can travel but does not otherwise impact its mobility.

We chose this UAV because it is portable, certified by European aerial vehicle safety

[1] At the time we experimented outdoors, the United States Federal Aviation Administration regulates UAV operation with 'Certificates of Authority' (COA), without which we cannot fly outdoors untethered. We have recently received our COA and in the future will only tether when we require extra safety.

standards, includes extensively-tested control software, has ample payload for the quantity of water we intend to carry ($\approx 100\ g$ of $600\ g$ payload), and in case of motor failure can still fly with only five of six motors functioning.

The Firefly is built with a modular design, so that the arms, struts, motors, motor controllers, (see Fig. 4.2) and radios can be removed independently. We take advantage of this modular design by using the screw mounts on the frame to connect the water sampling system to the UAV. This allows the vehicle to be maintained more easily as broken components can be swapped out for new ones. Of all components, the rotors (also called propellers or 'props') are the most likely to break. In fact, the rotors are designed to break away because more rigid props are more dangerous to humans. Further, the vehicle has a 12 V power interface, which powers the embedded system discussed in Sec. 4.3.9.

While operating the Firefly, we use a launch checklist and maintain a safety protocol. Our operation protocol specify that humans should stay at least $3-4\ m$ away from the vehicle during operation. Our launch checklist with the Firefly includes activities for every launch. The checklist includes inspecting the vehicle's propellers and arms, checking the battery connections, ensuring that the launch site is suitable (wide open and will not entangle the vehicle), testing all radio communication channels, checking the vehicles sensors are reporting expected values, engaging the motors without launching to ensure that all motors work, verifying that the ultrasonic sensors are working properly, and setting the backup-pilot controller into a default hover position so that the system tends toward a stable state if the computer-control link is severed.

The on-board control of the Firefly is split into low-level and high-level controllers. We do not modify the included software and therefore avoid introducing potential bugs into the control software, especially since our intent is to fly over water. The low-level controller takes input from the accelerometers and gyroscopes to minimize variations in pose (keep the vehicle steady). The low-level controller cannot be re-programmed, and is verified by the manufacturer. The high-level controller receives commands from the radios. It can be reprogrammed by modifying source code provided by the manufacturer. However, we do not modify the high-level controller because troubleshooting embedded code is slower than rapidly and iteratively developing off-board programs. Therefore, we leave the Firefly code unmodified so that we always have factory-tested code available in case our controls fail. This allows the overall system to quickly switch into a reliable, safe mode.

The Firefly navigates using a built-in GPS circuit and an air pressure altimeter. We utilize the GPS for navigation outdoors, which is sufficiently accurate in x and y dimensions because we assume that lakes and rivers are larger than the margin of error of GPS. We use the on-board pressure for altitude estimation at higher altitudes. However, the pressure sensor is not accurate enough to measure the vehicle's altitude over water, and is augmented with addition sensors. These additional sensors are discussed in Sec. 4.4.1.

The Firefly has three modes of operation: 1) manual; 2) pressure-controlled altitude, and 3) GPS with pressure-controlled altitude. We use the third method, in which the innermost control loops for the vehicle remain on-board within untouched, verified code. In manual operation, the speed of the motors corresponds directly to the 'stick commands' received by the controller. Using manual mode would give us more refined control of the thrust and attitude, but would remove a layer of redundancy in case our control input fails. Pressure and GPS modes are similar, and in both, the Firefly uses the on-board pressure sensor and accelerometers to minimize changes in altitude. We use the GPS pressure-controlled mode, in which the UAV will try to stay at the same altitude and GPS location unless the controller issues significant commands. We also use a PID controller on GPS, which is part of the UAV control software discussed in Sec. 5.1.1. By using the pressure-control mode, we trade some measure of precision control in exchange for a safer mode of operation during development. In pressure-GPS mode, the on-board controller ignores very small changes to the control input. This portion of the ignored control input space is called the 'deadband', and we implement a special controller to overcome this. This pressure-controlled altitude figures importantly in the overall behavior of the system, and further discussions of this mode can be found in the sections on software (Sec. 5.1.1) and altitude estimate (Sec. 6.1.4).

4.3 Design of the UAV Water Sampling Mechanism

The purpose of the sampling mechanism is to capture separate water samples from the environment while the UAV flies above the body of water. First, we explored using a pharmacist's ampoule (a special vial, shown in Fig. 4.3) affixed to the UAV by string. Then, the UAV would fly low enough to submerge the dangling ampoule. It was very difficult to fill the vial without dragging it through the water. In addition, the water-filled ampoule acts like weight on a pendulum, and the pendulum motion of the vial induces a pathological precession in the UAV's flight dynamics. While it is possible to overcome oscillations [51], we chose the simpler approach of avoiding them. Therefore, we considered landing the UAV in the water, but rejected this approach as discussed in the related work, Sec. 2.6. Finally, we settled on an approach in which we fly close to the water, and bring the water up to reservoirs close to the UAV's center of mass. This approach keeps the system safe, dry, and flyable, but requires getting close to a surface without hitting it.

Figure 4.3: Pharmacist's Ampule - an early failed prototype.

A rigid frame, or 'chassis', anchors our mechanical design to the UAV. The chassis holds three 20 *ml* screw-top glass vials, spring-hinged lids, a servo to direct the water, and mount points for the dangling tube, embedded controller, and ultrasonic range sensors. The chassis holds all the pieces together.

4.3.1 Frame to hold components: the 'Chassis'

The chassis of the water sampling mechanism is designed in collaboration with Mechanical Engineer Baoliang Zhao and Dr. Carrick Detweiler. A picture of the 3-D model of the chassis is shown in Fig. 4.4. Baoliang created the design for the mechanism by modeling components in the SolidWorks [77] computer assisted design program. We 3-D printed each component of the chassis using acrylonitrile butadiene styrene (ABS) plastic and a MakerBot Replicator [78]. Initially, we printed designs and evaluated them based on how the parts interfaced with sensors, vials, and other items with fixed shapes. Further we evaluated the print quality to determine if changes to the 3-D design will improve the final piece.

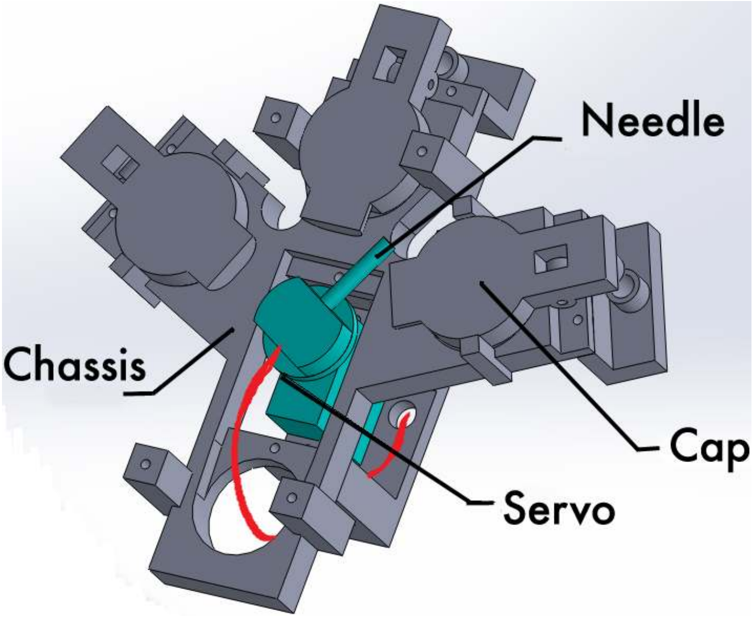


Figure 4.4: Sampling Mechanism

For example, the MakerBot can print at various speeds and densities. Faster prints have less fidelity and printing 'artifacts', which can make the piece difficult to use. Less dense prints are lighter but might also break. After we printed all components, we drilled out and tapped screw holes, and mated the components together with steel 2-56 screws. Next, we tested the chassis for fit with the other components, including the glass vials.

4.3.2 Glass Vials and Lids

The water sampling mechanism consists of three spring-lidded chambers. We constructed the chambers so that a servo-rotated 'needle' lifts the lid and directs the water flow into one of three 20 *ml* glass vials (Fig. 4.9). Once the needle rotates away from the vial, the spring holds the lid closed.

The vials are chemically inert clear glass instead of plastic, which is lighter, because plastics might bias water properties. We also wanted perfectly clear vials to enable quick visual inspection of their contents. The glass 20 *ml* vials are available in several formats, and the ones used in this project are the shortest and stoutest of those available from McMaster-Carr [79]. Having shorter vials is useful because the total height clearance is constrained by the UAV's landing struts. Further, the stouter the vial, the closer the center-of-mass of the water is to the center-of-mass of the UAV.

The vials have screw-top threading, which we use by printing the 'chassis' with mated thread coupling. We measured the threading on the glass vials, both the number of threads per centimeter as well as the maximal width and depth of the thread. We use these measurements to design 3-D receivers with a variety of tolerances, as shown in Fig. 4.5. We labeled each printed receiver with a unique number, and evaluated for a solid fit that is not too tight. We

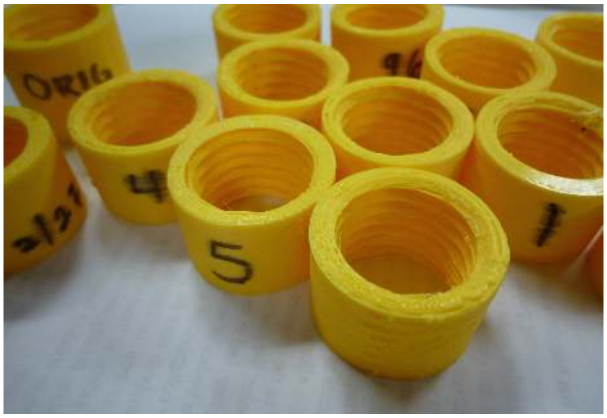


Figure 4.5: Several test vial thread jackets printed to find the best fit.

wanted users of the system to be able to unscrew the vials with one hand, while also having no concern that the vials would shake loose.

4.3.3 Servo to direct flow of water: the 'Needle'

As water flows up the tube, a servo controls where the water goes: either captured in a glass vial, or jettisoned overboard. The servo, an HS-65MG 'Mighty Feather' [80], is affixed to a small plastic housing which confines a 0.5 *cm* diameter rigid plastic tube 4 *cm* in length (the 'needle', shown in Fig. 4.6). The servo confines the rotation of the needle in a plane and can rotate 160° total, 80° from center in either direction. Because the servo's range of motion is limited (it can't spin all the way around), the design of the chassis holding the vials is limited by the servo. In practice, the three vials are sufficient for the initial requirements, but if we want to add more vials, a new 360° servo would have to be added. Also, increasing the range-of-motion of the servo might cause the flexible tube to 'kink' at extreme rotations. The servo rotates the needle into one of five pre-defined positions: three vial-filling positions and two water-jettisoning positions. Each servo position is defined in the software as a number of microseconds governing the length of a pulse in a pulse width modulation (PWM) signal. The number of microseconds corresponding to each position is determined by trial and error after 3-D printing the chassis holding the vials. The microsecond control pulses and 5 *V* power for the servo are generated by the embedded microcontroller.

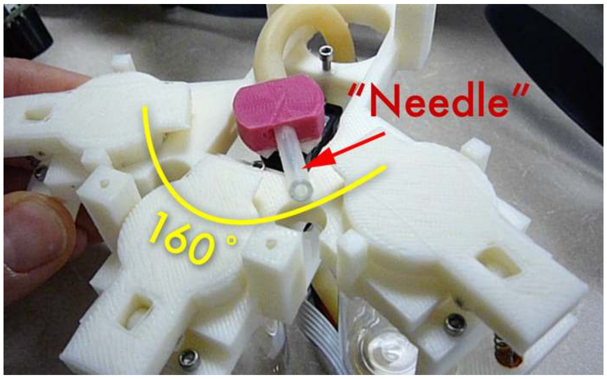


Figure 4.6: Rigid plastic tube to direct water flow: the 'Needle'.

4.3.4 Flexible Tube

We connected the needle to a section of flexible silicone tubing in series with an additional 1.05 m plastic tube hanging below the UAV. The flexible silicone tubing allows the servo to rotate and is not load bearing. We choose the length of the 1.05 m tube because of the characteristics of the pump discussed in Sec. 4.3.5. The tube attaches below the center of mass of the unloaded vehicle, to minimize changes in flight dynamics while pumping. We tested several types of flexible tubing, varying by inner diameter and rigidity and selected the tube's inner diameter by measuring the size of the coupling with the pump. After buying several types of various rigidities, we found a flexible tube that could bend enough to curl under the UAV during landing, while rigid enough to dampen motion-induced oscillations.

4.3.5 Submersible Pump

The 1.05 m plastic tube terminates at a micro submersible water pump [81] attached at the end of the tube. We chose this pump, shown in Fig. 4.7, because it was the lightest we could find that could pump to at least a height of 1 m. The pump's mass is 10 g, and has a 2.64 mm opening. The pump can be powered by a voltage from 3 V to 4.6 V, where additional voltage results in faster pumping. However, the manufacturer recommends against running the pump for extended periods at the highest voltage.

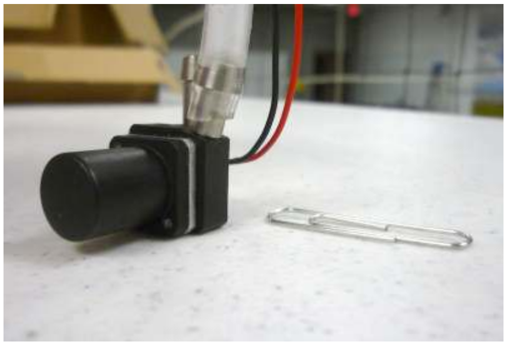


Figure 4.7: Micropump with reference paperclip.

Therefore, we characterized the pumping height and speed over a range of voltages. We put the pump in a large bucket of water and powered it with a variable power supply. We connected a flexible tube that empties into a graduated vessel. We placed the receiving vessel at a measured height using two adjoined meter sticks, and then timed how long it took for the pump to discharge 250 *ml* of water. We explored a range of heights and voltages using 77 test points, seven voltages by eleven heights. The results are shown in Fig. 4.8.

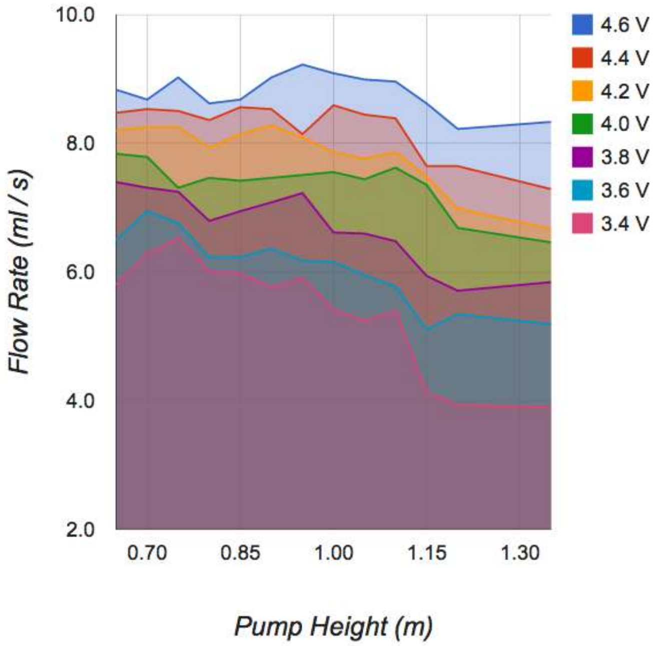


Figure 4.8: Pump Flow Rate by Height and Voltage

At all tested voltages, the pump lifts water to 1.10 *m* over pumping rates increasing proportional to increasing voltage. Above 1.10 *m*, the pumping rate decreases somewhat, and level off at the higher voltages while decreasing at the lower voltages. We chose 4.2 *V* for the voltage used in the water sampling system, as a balance between fast pumping and less wear on the pump.

Since the pump is intended for outdoor use, a course mesh filter was sewn together to form a bag and wrapped outside the pump. We tried different grains of mesh, and found that finer-grained meshes ($<$ 500 *microns*) reduce the flow of water. Larger-grained meshes allow particles that can block and perhaps damage the pump. We chose a 2 *mm*-grained mesh to protect the pump without inhibiting flow

4.3.6 Flushing

The servo can also select an intermediate position to enable flushing of the needle and tubing between samples (Fig. 4.9). The stream of water shoots out of a gap between the sample chambers. We have two gaps: between vials one and two, and between vials two and three. By having two gaps, we reduce the risk of cross-contamination by never moving the needle past a lid that seals a filled vial. The duration of the flushing phase is configurable, defaulting to 20 s, three times the duration required to fill a 20 ml vial [2].

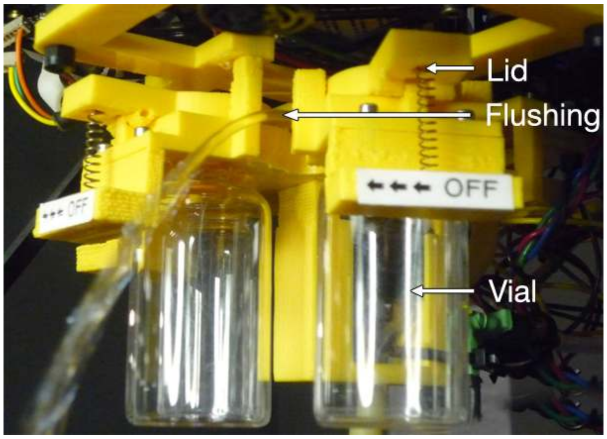


Figure 4.9: Flushing the sample system.

4.3.7 Breakaway Tube

A breakaway mechanism show in Fig. 4.10 allows the pump and tube mechanism to release if subjected to 15.1 N of force. This might happen if the pump becomes entangled in the environment, and the UAV thrusts away from it. We conducted a test of 15 'pull-tests' using a pulley and a hand scale, which averaged 1.54 kg with a standard deviation of 0.19. We measured the maximum thrust of the unloaded UAV to be $\approx$ 17.7 N. So far, all entanglements have worked themselves free before the breakaway released.

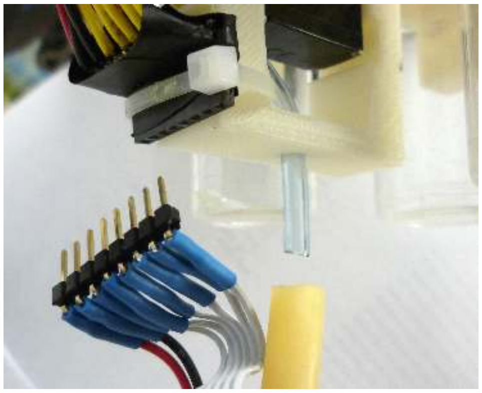


Figure 4.10: Breakaway mechanism.

[2] Initial experiments show that 20 s flushing avoids cross-contamination. We plan to more rigorously characterize this in future work.

4.3.8 Tether for Passive Safety

In order to comply with local and national regulations regarding UAVs, and as another layer of safety in the field, we flew attached to a passive tether. As shown in Fig. 4.1, we attached heavy pink kite cord to the metal frame of the UAV by a metal carabiner. To help the string stay out of the way of the props, a small weight (a washer) is tied approximately one meter from the UAV, shown near the water's surface in in Fig. 4.1. A human operator with a spool holds the other end of the string. The job of the human operator is to help ensure that the string does not interfere with flight. In practice, the string does not interfere with the operation of the UAV so long as the human operator releases sufficient slack.

These components, the chassis, vials, servo, tubing, pump, and breakaway tube together make up the mechanical parts of the water sampling system. In addition to these mechanical parts, the water sampler includes sensors and an embedded control system.

4.3.9 Embedded System

We designed an embedded system for the water sampler. The embedded system must be light, run on the 12 V power available from the UAV with minimal impact to battery life, fit onto the UAV, and be rugged enough for field deployment. It rides along with the UAV, is at the center of water sampling and altitude sensing, and has multiple purposes:

- Receive commands from the ground station.
- Send sensor summary and status to the ground station.
- Control the power to the micro pump.
- Read water sensor values.
- Convert the 12 V power from the UAV to 5 V (CPU, servo, and pump) and 3.3 V (XBee radio)

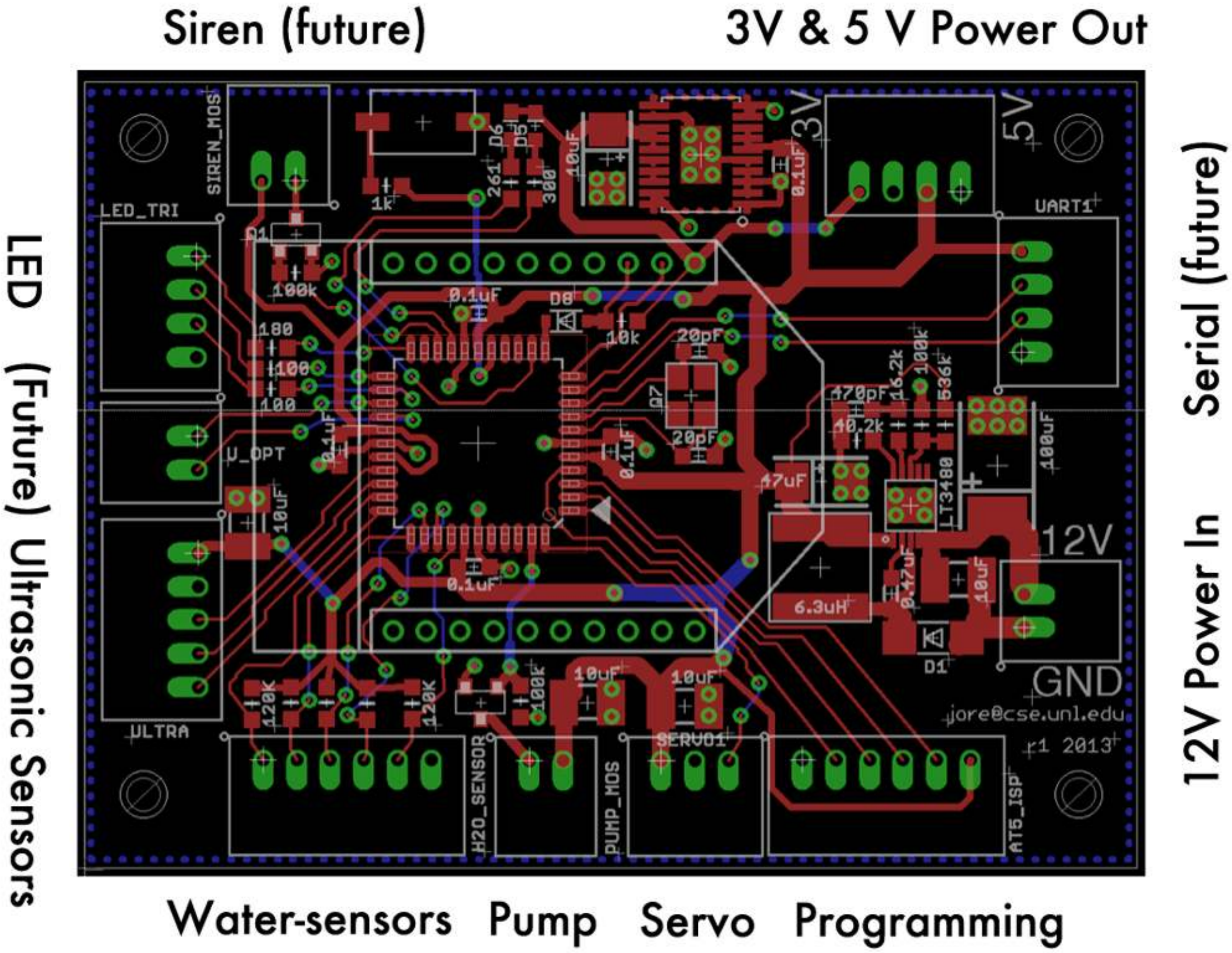


Figure 4.11: Embedded System Schematic

- Read ultrasonic rangefinder sensor values.
- Control the position of the servo.
- Control a water status LED.
- Adapt to multiple project purposes and future needs.

To meet these requirements, we chose an Atmel Atmega 1284p microcontroller. The Atmega 1284p can run at up to 20 *MHz* (we use it at 8 *MHz*), is powered by 5.0 *V*, and includes 128 *Kbytes* of RAM for both instructions and memory.

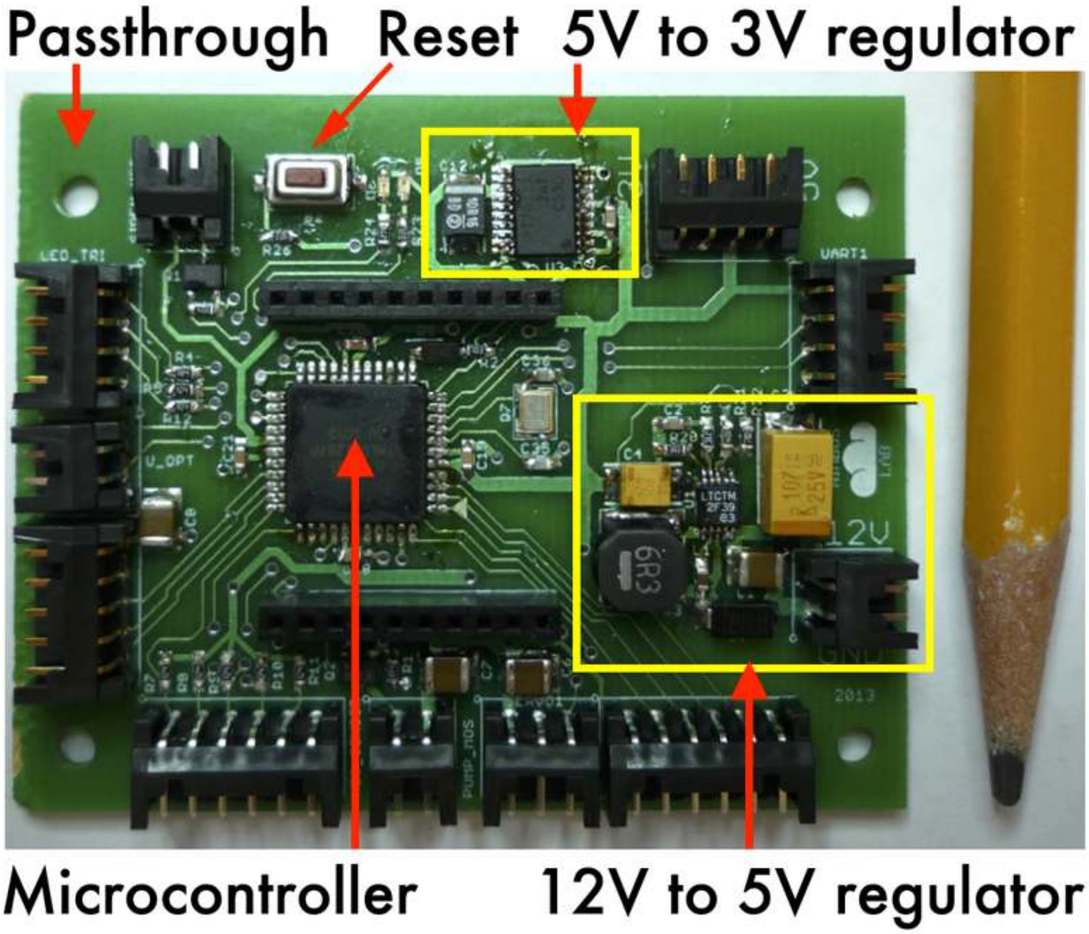


Figure 4.12: Populated Board

The populated board, shown in Fig. 4.12 has power regulators, two power controllers (MOSFETs), two LEDs, 11 plug mounts with 32 total external pins. The purpose of the embedded system is to govern the pump and 'needle' servo, read sensor values, and communicate with the base station. We designed the board using the 'Eagle' layout tool and included two voltage regulators: 12 V to 5 V and 5 V to 3.3 V. As shown in Fig. 4.11, the traces carrying power to the pump, servo, and CPU are all wider to allow more current to flow. The board is a 2-Layer design, with the whole back plane dedicated to ground current. We gave special consideration to the path of ground current from the pump and servo to help make the power cleaner, because in the Arduino version the servo exhibited ‘jitter’ from noisy power. Fig. 4.12 shows the ‘pass-through’ holes (with a pencil for size reference), used to affixing the board to the 3-D printed chassis.

We soldered the components of the boards by hand using a binocular microscope. First, we completed the power regulators, and after testing the power we added the remaining components. The embedded system features a reset button near the top left in Fig. 4.12. After soldering all the components, we programmed the board over the programming port using an AVRISP-mkII. Details of the software used on the embedded system can be found in Sec. 5.2.

4.4 Sensors For Near Water Flight

Our mission requires that the UAV fly near the water to collect samples, but absolutely not land in the water. Flying near the water is difficult because the sensors included with the UAV do not sense proximity. We utilize all the built-in sensors during flight (discussed in Sec. 4.2). However, for flying close to the water, these sensors do not provide sufficient resolution in measuring altitude. Therefore we augmented the system with near-range ultrasonic rangefinder sensors. In addition, we added water conductivity sensors to the flexible tube to detect the presence of water. These sensors are critical for finding and maintaining the correct distance to the water, with the pump submerged, during the pumping and flushing stages of the mission. This section provides a detailed description of the ultrasonic rangefinders and conductivity sensors, their characteristics and limitations, and how they are deployed in the system.

4.4.1 Ultrasonic Rangefinders

To improve height estimation (discussed in detail in Sec. 6.1), we augmented the UAV with ultrasonic rangefinders. Ultrasonic sensors emit high-frequency sound waves that propagate at $\approx 340ms^{-1}$, the speed of sound. The sound wave reflects off of surfaces or objects and the reflected signal vibrates the sensor's membrane, inducing an electrical signal measured by the sensor. By measuring the time from transmission to reception, the sensor estimates the distance to the surface. We use the Maxbotix MB1240-EZ4 ultrasonic rangefinders [82], which the manufacturer specifies to have 1 *cm* resolution. In addition, Maxbotix recommends the EZ-4 for UAV applications because it is more resilient to noise and especially reliable within 3 *m*. The pulse created by the EZ-4, an ultrasonic wave, propagates in the shape of a cone, and this particular model has the smallest cone available. The shape of the cone is important because we want the sensor to detect only what is immediately below it, and not the tube, which is below it at a small angle.

A problem with 'narrow cone' ultrasonic sensors is that they stop sensing reliably when not pointing straight down. At large pitch angles $> 20^\circ$ (the UAV tilts up or down), at an altitude of 1.0 *m*, the ultrasonic wave reflects away from the vehicle and the sensor reports 'MAX_RANGE'. We address the large angle problem by capping the maximum value for the ultrasonic sensors. In practice, such extreme angles are rare while hovering, and we plan missions to approach the water from above rather than flying close to the water at a steep attack angle.

The placement of the ultrasonic sensor is guided by two opposing considerations, the dangling tube and the prop blast. The dangling tube can interfere with the ultrasonic when it swings, so placing the ultrasonic sensors farther away from the tube is good. The farthest point from the tube is directly under the props, but the prop blast causes turbulence which interferes with the sensors, so this is too far. To find middle ground between the tube and the prop blast, we extended the 3-D chassis to include mount points for the ultrasonic sensors, as shown in Fig. 4.13. These mount points flank the dangling tube 10 *cm* from the center, between the props, and directed straight down. However, since the two sensors are offset from the center of the vehicle, at small pitch angles the measured distance at the front and back sensors is different, one higher than the other.

To compensate for the small angle problem, we utilize readings the UAV's gyroscopes to correct the measurement. This configuration increases the likelihood of an unobstructed path to the water's surface, which the swinging tube might otherwise block, while also minimizing the turbulence from the prop blast.

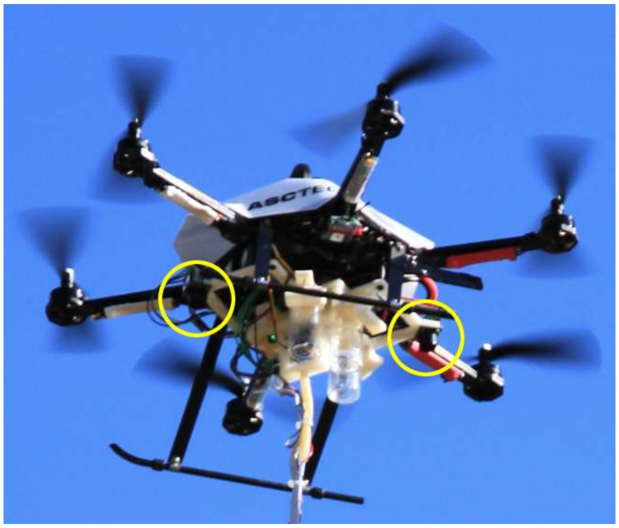


Figure 4.13: Ultrasonic sensor placement.

Having two sensors that both measure altitude helps avoid occlusion from the tube, but we need to be careful because the sensors can interfere with one another. One sensor might detect an ultrasonic wave generated by the other sensor. Therefore, we carefully arrange the timing by running them in 'chaining' mode, offsetting their sample time by 50 *ms*. This ensures that the signal from one sensors can propagate, reflect, and travel back the sensor before the other sensor turns on. Since each sensor runs at 10 *Hz* and their trigger is offset by 50 *ms*, the effective rate of new sensor information is 20 *Hz*. Chaining requires wiring the two sensors together and holding the voltage high to a command pin on one of them to initiate the alternation. Unfortunately, 'chaining' mode does not always start correctly as the electrical system approaches a steady-state, and at present the embedded system must be restarted to fix it. Further, it is difficult to detect this failure while the UAV is on the ground. Therefore, we work around this in our pre-launch protocol by verifying the ultrasonic readings as we lift the vehicle off the ground.

Another problem is that the ultrasonic sensors are noisy in chaining mode above a certain distance. The ultrasonic sensor's specified range is from 17 *cm* to 7 *m* in single mode when mounted to a rigid surface. However, we found their reliability in our UAV application ceases above $\approx 2.5\ m$. Fortunately, we are interested in readings $< 2\ m$, and therefore we cap the ultrasonic readings to $1.85\ m$. Even if the UAV descends toward the water and does not detect it until $1.85\ m$, still the UAV has ample time to decelerate before reaching the target altitude of $\approx 1\ m$. A more complete discussion of how the ultrasonic readings are used in altitude estimation is in Sec. 6.1.4

4.4.2 Water Conductivity Sensors

One reason why it is important to have accurate altitude is that the pump must be submerged and primed prior to operation. To know that the system is actually touching water and not just approaching dry ground, and as an additional safety system, we augmented the system with water conductivity sensors, as shown is Fig. 4.14. These sensors consist of small wires running along the dangling tube with an exposed positive and negative terminal (shown in Fig. 4.14 with yellow circles) every 10 cm from the bottom up to 50 cm. The pairs of exposed wires each form a voltage divider. When submerged, current flows across the pair and the measured voltage on the circuit drops. A 10-bit analog to digital converter (ADC) measures the circuit voltage and 'wet' is determined to be any value below 850 on a $0 - 1023$ scale. In the wild, the conductivity of water varies depending on the number of free ions, so the ADC reading which means 'wet' can be changed at runtime.

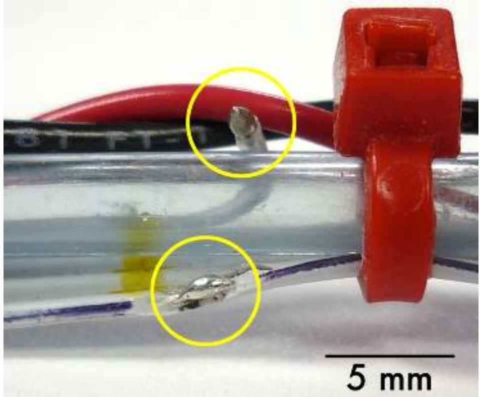


Figure 4.14: Water conductivity sensors.

The conductivity sensors also govern the pump. When the lowest conductivity sensor (at the micropump) reads 'WET', the embedded microcontroller turns on the pump, but only after being wet for more than 400 ms continuously. During this 400 ms, water floods the pump, priming it. If the conductivity sensors read 'DRY' while the pump is supposed to be on, it turns the pump off and must wait another 400 *ms* to turn the pump back on. Experimentally, we determined that 400 *ms* works at least nine times out of ten, but that occasionally it would not prime no matter how long we waited. We believe this might be caused by air bubbles stuck in the pump. We have several pumps of this same model and some prime more easily than others.

The 'WET' and 'DRY' states of the conductivity sensors are displayed by a tri-color LED controlled by the embedded system. This LED is used to communicate the embedded system's perception of the state of the water conductivity sensors (water sensors discussed in Sec. 4.4.2). When all sensors report 'dry', the LED is green, except in the first 8 seconds after turning on the system, which indicates a reboot. When the lowest conductivity sensor only is wet, the light turns blue. When the bottom two conductivity sensors are wet, the light turn purple. If three or more conductivity sensors are wet including the highest sensor, the LED remains purple but the UAV is instructed to gain altitude. The LED is positioned inside the UAVs protective plastic shell, which is semi-translucent and allows the LED to be seen from all sides.

4.5 Electro-Mechanical Summary

The goal of all the electro-mechanical design, from the UAV, to the sensors for near-water flight, to the mechanisms to pump and store water, is to have a total system in which the pieces work together to fulfill the requirements of our water sampling collaborators. This chapter explains all the separate components, including how they were chosen and evaluated. In the next chapter we will consider how all these separate pieces can be brought together to perform a whole task, which requires the pieces to fit together and or operate in a specific order. These pieces are all directed by a software control system, which we will consider in the next chapter.

Chapter 5

Software

The high-level software architecture is shown in Fig. 5.1. After the electrical and mechanical systems have been designed and assembled, we must carefully coordinate the control commands streaming out of the system and the sensor data streaming into the system. This coordination is the job of the control software. The software system can be thought of as having two parts: 1) code on a ground station using the Robot Operating System [83] (ROS) which handles low-level communication with the UAV, mission control, navigation, state machines, and altitude estimate; 2) code on an embedded controller attached to the UAV that receives instructions from the ground station, controls the water-sampling subsystem, reads ultrasonic and water sensor data, and broadcasts the water-sampling sub-system's state. These two parts communicate with one another via XBee radio links. Both sub-systems incorporate predicates to detect unsafe water sampling or navigating conditions based on the sensor readings, and restart a mission. In total, the system includes about 7K lines of C, C++, and Python code.

The system is built on ROS, a 'meta-operating system.' ROS is a way to launch processes across a set of devices and exchange information through 'topics'. We use ROS because it simplifies the system by providing a standard way to interconnect producers of

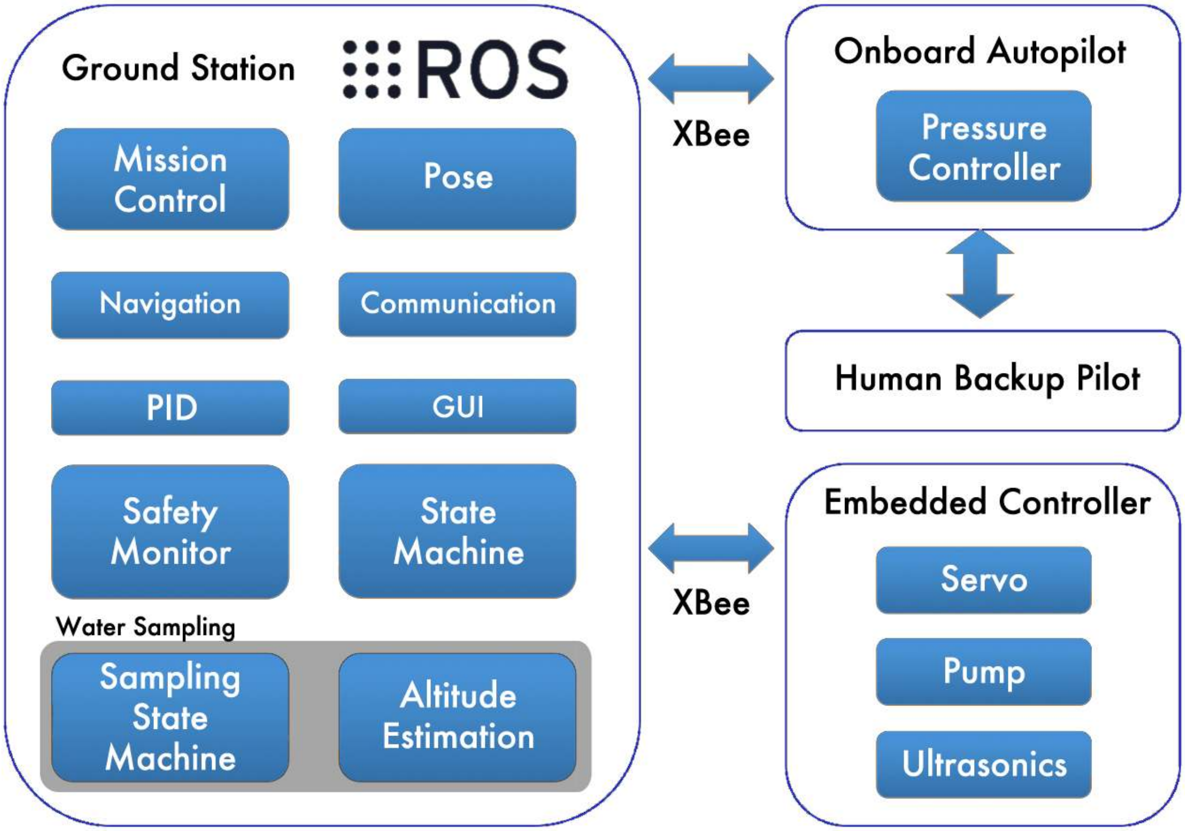


Figure 5.1: High-level organization of software architecture.

information (like sensors) and consumers of information (like filters). Some processes, or 'ROS nodes' both consume and produce information, like a state machine that consumes sensor data and produces a signal called a 'ROS topic message.' A 'ROS topic message' is a strongly typed data structure 'published' over TCP/IP to 'subscribers'. To publish or subscribe, nodes initially talk to the 'ROS Core', a single central process to manage processes. The ROS core lists all available topics and connects publishers with subscribers. We run the ROS Core and all ROS processes on the ground station because it is easier to prototype and frees us from performance constraints of on-board hardware. Eventually, all control software could run on-board the UAV, allowing greater autonomy, but the design of the software architecture is independent of this implementation strategy.

In this chapter, we describe the design of our Finite State Automata (FSA) that implements the water sampling behavior, how the water sampling system utilizes our generic UAV control package (MIT-AscTec), and the layout of the embedded software running on the custom microcontroller built onto the UAV.

5.1 Ground Station Software: Mission Control, State Machine, Safety Monitor

The ground station software is organized into two parts: code for controlling a UAV in general, and code specific to the domain of water sampling. First, we introduce the general UAV code, and then describe the water sampling code in detail.

5.1.1 General UAV Code

We utilize existing UAV control code, called 'MIT-AscTec', initially developed by Dr. Brian Julian and heavily modified by the NimbusLab. The purpose of this code is to provide a framework for organizing commands to the UAV at a higher level than 'stick commands'. 'Stick commands' are command primitives, which mimic moving the remote control 'sticks' with your hands. The 'MIT-AscTec' code handles low-level communication with the UAV including the UAV's sensors. We use data from the UAV's GPS, gyroscopes, pressure sensor, motor status, and battery. 'MIT-AscTec' also provides higher-level code like a Proportional-Integral-Derivative (PID) controller with the ability to navigate to GPS waypoints. Also, Dr. Julian's code includes a high-level state machine to turn on the motors and launch the vehicle. This code has been further modified by members of UNL's NIMBUS lab, to allow the input of a sequence of high level tasks, which we call a script or mission.

Modification to General UAV Code

The only portion of MIT-AscTec modified for water sampling is code contained in the critical PID controller, specifically code governing thrust. We use the Firefly in 'pressure mode' which means the UAV's onboard computer attempts to hover while minimizing changes in pressure. This means for the Firefly to operate in pressure mode (see Sec. 4.2 and Sec. 6.1.4 for more on 'pressure mode'), the 'stick commands' must be given outside the 'deadband' on thrust enforced by the UAV. Therefore, when the UAV is near the desired hover point above water, the PID operates in a special mode which adds significant quantities of thrust to positive values near zero, and subtracts significant quantities of thrust for negative quantities near zero. The overall effect is meaningful altitude commands to the UAV in pressure mode using the ultrasonic sensors as an input. Other than this modification to the general code, all additional ground station code is an extension of the 'MIT-AscTec' package.

5.1.2 Water Sampling Specific Code

The ground station code specific to water sampling, falls into three groups: 1) code for the water sampling state machine; 2) altitude estimation code; and 3) low-level communication code to encode and decode data packets from the radio. Code in the first group is written entirely in Python, the second group is a mix of Python and C++, and the last group is written entirely in C++. We used python for rapid development and entirely new portions of the code, and C++ when extending the existing code base (entirely C++). We consider each group in turn.

The first group of domain specific code implements the water sampling state machine and spawns separate processes to time the pumping and flushing. The water sampling state machine is everything in Fig. 5.2 surrounded by the dotted line. The software coordinates these activities through: 1) waypoints, which are compared to the measured location of the UAV, so that the UAV descends to the target height; 2) timers, which track how long the pump has actually been pumping and infer that the tube has been sufficiently flushed or that the vial is full; and 3) safety predicates on sensor values, which ensure the sampling altitude is safe. If the safety constraints are violated, the UAV retreats to a safe altitude. Then the mission continues with the next sampling location.

The second group of sampling specific code pertains to altitude. We filter the altitude estimation through a Kalman Filter as discussed in Sec. 6.1.3. Code for the Kalman filter [58] subscribes to ROS topics containing ultrasonic range information ('water_sampler_board_processed') and pressure information ('subject_status'). This code uses C++ matrix and vector structures from the Eigen/Dense class. The purpose of this code is to implement a pre-filter on the sensor stream (see Sec. 6.1.4 for details) and a Kalman filter on the pressure and ultrasonic readings. This altitude estimate is published on a ROS topic ('kalman_height') that is consumed by a Python node which publishes the 'final altitude estimate' (more in Sec. 6.1).

The third group of water sampling code is a serial driver that subscribes to water commands, serializes them and adds a checksum, and sends them to the embedded system over the XBee radio on the serial port. This code also receives incoming packets from the embedded system, checks the guard bytes and checksum, and publishes the received information without interpretation as the ROS topic 'water_sampler_board_raw'. Since this code in modeled on existing C++ code, it is programmed in C++.

5.1.3 Finite State Automata (FSA)

At a high level, the software system implements a Finite State Automata (FSA) as shown in Fig. 5.2. These states together are an abstraction of the behavior of the whole system. Each oval in the figure represents a logical state, which is encoded in the software as a configuration of ground control and embedded system software. The arrows in the figure

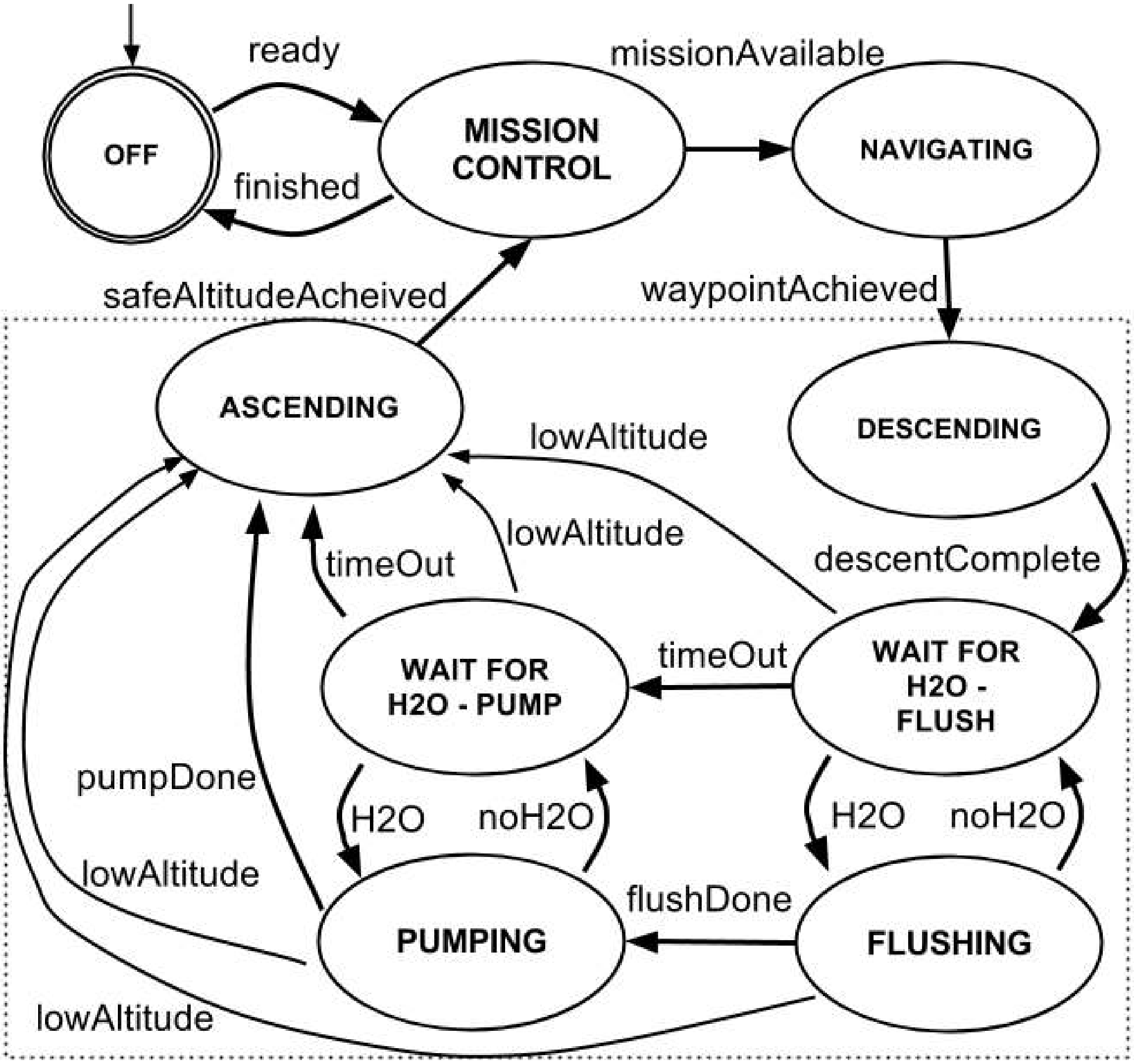


Figure 5.2: Whole system finite state automata with sampling states. Dotted Box surrounds the water-sampling portion of the state machine.

represent state transitions, labeled with high-level descriptions of the event that triggers a transition from one state to another. The flow of activities is clockwise starting from the 'OFF' state in the upper-left corner. The software is designed to recognize the high-level event causing each transition, and the system changes its behavior based on the 'state'.

From 'OFF', the system starts when the UAV and ground station are ready, and then the control flows to 'MISSION CONTROL'. If a mission is available, the system transitions to the ‘NAVIGATING’ state, where the UAV takes off and goes to a GPS location. Once the UAV arrives at the sample location (‘waypointAchieved’), the system moves to the ‘DESCENDING’ state. When the target height has been reached, the system tries to detect water (‘WAIT FOR H2O - FLUSH’), and once water is detected, water is flushed through the tube to clean it. After flushing completes, the systems starts ‘PUMPING’, where it captures water in a vial, and starts or stops the pump based on whether the conductivity sensors report ’H2O’ or ’noH2O’. After pumping, or if the system takes too long to pump, or if the altitude ever goes to low (‘lowAltitude’), then the system transitions to the ’ASCENDING’ state, and moves up away from the water’s surface and the possible danger of getting wet. After ascending, the system returns to ‘MISSION CONTROL’ and either starts a new mission or returns to base and ends the program.

The ground station software is organized and segmented to implement this FSA.

We now move from the ground station software, to the software on the embedded system, which is attached to the UAV.

5.2 Embedded Software: Pumping, Flushing, and Sensing

The purpose of the embedded system is to control the water sampling mechanism onboard the UAV and transmit sensor data. The embedded system turns the water pump on and off, moves the servo, times how long the pump has been wet, and builds data packets to be sent over the XBee radio. In order to accomplish these tasks, we use a sequence of smaller computer programs, or methods, which run one after another in an infinite control loop. We program this code on a laptop or desktop computer, and then cross-compile the code to run on the embedded system. The Atmega microcontroller carried by the UAV (described in Sec. 4.3.9) is compiled using the SEAMos build system [84] created by Dr. Detweiler and programmed using the SCONS software construction tool [85].

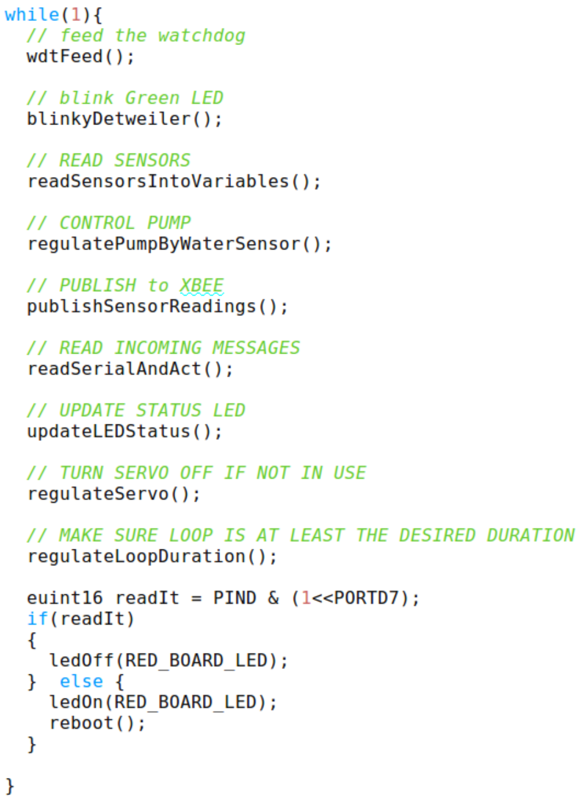
The software consists of two phases: initialization and a main control loop. The initialization phase configures various subsystems, including the LEDs, the pulse width modulation (PWM) timer registers for the servo, the serial port speed, enables interrupts, starts the system clock, and sets the default position of the servo. Interrupts are required by the system timer, but are otherwise not used by the program. This initialization code is run once, and specifies the behavior of the system until it is powered off or reset. The main control loop has four main goals: 1) read the sensors; 2) transmit sensor readings; 3) read incoming commands; and 4) act on the sensor readings and incoming commands. An overview of the main loop is shown in Fig. 5.3.

wdtFeed()

The main loop begins by 'feeding the watchdog timer' (wdt), which is a failsafe mechanism designed to restart the system (jump to instruction address 0) in case the loop takes too long. This backup mechanism has not been observed to restart the system in practice, and would cause the LED to turn red for eight seconds.

blinkyDetweiler()

This blinks the small green LED on the board (not the water status tri-color LED). This lets users know the boards is working, and is also known as a 'heartbeat'. This method is named in honor of my thesis advisor, Dr. Carrick Detweiler, who taught me how to turn on LEDs.



```
while(1){
  // feed the watchdog
  wdtFeed();

  // blink Green LED
  blinkyDetweiler();

  // READ SENSORS
  readSensorsIntoVariables();

  // CONTROL PUMP
  regulatePumpByWaterSensor();

  // PUBLISH to XBEE
  publishSensorReadings();

  // READ INCOMING MESSAGES
  readSerialAndAct();

  // UPDATE STATUS LED
  updateLEDStatus();

  // TURN SERVO OFF IF NOT IN USE
  regulateServo();

  // MAKE SURE LOOP IS AT LEAST THE DESIRED DURATION
  regulateLoopDuration();

  euint16 readIt = PIND & (1<<PORTD7);
  if(readIt)
  {
    ledOff(RED_BOARD_LED);
  }  else {
    ledOn(RED_BOARD_LED);
    reboot();
  }

}
```

Figure 5.3: Embedded control loop after initialization. Note the 'while(1)'.

readSensorsIntoVariables()

This method performs seven analog-to-digital (ADC) conversions: five on the water conductivity sensors and two on the ultrasonic sensors. These values are loaded into globally scoped variables which are later read by the pump monitor and radio packet serializer.

regulatePumpsByWaterSensor()

This code uses in the information gathered by the water sensors and uses it to govern the pump. If the pump has been commanded 'on' by the ground station, this function monitors a timer that counts how long the bottom sensor has been continuously wet. Once we reach this time threshold, we power the pump by sending a PWM signal to the MOSFET (power regulator). If the bottom water sensor is ever 'dry' then the timer is reset. The sensor can read 'dry', for example, when the vehicle gains altitude after a gust of wind.

publishSensorReadings()

This method gathers the sensor readings into a packet (a byte array) and sends it over the serial port to the XBee radio that transmits it to the ground station. This packet includes two 'guard bytes', a short code which informs the receiver it should attempt to read the following bits. The packet includes a standard checksum and is 21 bytes.

readSerialAndAct()

This function reads incoming control packets from the ground station. It receives commands which govern the position of the 'needle' servo (see Sec. 4.3.3), commands the pump, and sets the definition for the ADCs for 'wet' (see Sec. 4.4.2 for details). This packet starts with two 'guard bytes' and ends with a checksum, which must be verified against the received data before the packet can be parsed.

updateLEDStatus()

This reads the status of the water sensors and controls the color displayed on the tri-color

LED, as discussed in Sec. 4.4.2.

regulateServo()

This method sets the PWM signal controlling the position of the servo based on the logical value of the 'needle' position variable. This position determines which vial will be filled with water or whether the pumped water is jettisoned overboard in the 'flushing' phase. The 'needle' position variable is set initially during initialization and can be modified by incoming commands from the ground station. The state machine dictating the sequence of 'needle' positions is encoded in the software of the ground station.

regulateLoopDuration()

We must regulate the length of the control loop to ensure we send the outgoing data packet at the desired broadcast rate, $\approx 60Hz$. If faster, the outgoing packets overwhelm the data channel and the ground station receives no data. Therefore, this method looks at how long has elapsed since this loop began and waits (idles) until a total of 30 ms has passed since the previous loop began. The loop might be longer or shorter depending on whether the serial buffer contains data that must be processed.

5.3 Summary

We have examined the system software architecture of both the ground station and the embedded system. We extended our generic UAV control code, written in ROS, to run a water-sampling mission and direct the high-level behavior of the UAV. The low-level behavior of the water sampling system, including the pump and servo, is run by an embedded system. The embedded system code runs on a single control loop onboard the UAV. Now that we have a physical mechanism and software to control it, we can test them together as whole subsystems, as described in the next chapter.

Chapter 6

Altitude Estimation and Water Sampling Effectiveness

Now that we have an electro-mechanical system and software to control it, we can turn our attention to two critical tasks: altitude estimation and water sampling effectiveness. We cannot expect the system to succeed in an unstructured environment in the field, if we cannot first succeed in a structured environment in the lab. Therefore we approach these two problems as challenges to be solved in lab experiments prior to moving the system into field experiments, discussed in Chap 7. In this chapter, we discuss our approach to altitude estimation, and validate it in lab tests in which the altitude mean absolute error is 0.017 m. We then measure the effectiveness of the water sampling system, and demonstrate that it is 90% effective while flying, collecting 81 full sample vials out of 90 attempts.

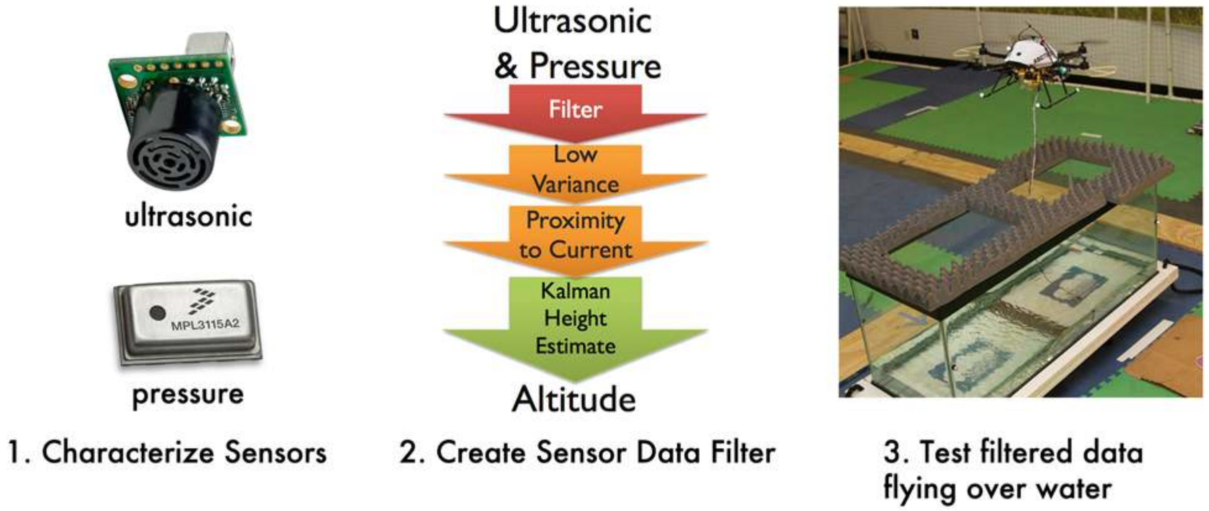


Figure 6.1: Approach to building altitude estimate.

6.1 Altitude Estimation Over Water

Flying near water is dangerous to the UAV because it can damage the electrical and mechanical systems, yet flying near water is absolutely necessary to sample water. It's difficult because the UAV does not come equipped with sensors to detect its surroundings, specifically, anything below it. Even after we add sensors to sense below the UAV, the sensors might read incorrectly because we have added a tube and pump. The sensors could read incorrectly when then dangling tube swings in front of the sensor during flight. We could encounter wind gusts, low battery, weak radio signals, software bugs, sensor noise, or the pump could become entangled with the environment. These problems might appear individually or in any combination. We knew from the beginning that if we adopted a method in which we do not land in the water intentionally, then altitude estimation to sample water would be critical so we do not land in the water unintentionally. This is why the physical design emphasizes redundant range sensors with additional conductivity sensors so that multiple measurements confirm our altitude estimate and increase the likelihood of a successful sampling result.

As shown in Fig. 6.1, we address these problems by breaking our altitude problem into three pieces: 1) characterize sensors: pressure (6.1.1) and ultrasonic (6.1.2); 2) pre-filter the data and use a Kalman Filter (6.1.3); 3) test our altitude estimation in the lab. For high altitude, we use the built-in pressure sensor, which is sufficiently accurate when the total altitude is $> 8\ m$ and the pressure sensor might drift $\pm 1\ m$.

6.1.1 Characterization of Pressure Sensors

We started by taking a closer look at the Firefly's altitude sensor, a kind of pressure sensor. The Firefly's pressure sensor is a piezoelectric microelectromechanical system (MEMS) that measures the change in electric charge caused by the mass of the atmosphere above the UAV. Pressure sensor data is useful because it is reliable over short periods of time ($< 1\ s$). The pressure sensor that ships with the Firefly samples at 500 Hz, fast enough to be the primary way the Firefly maintains altitude. By comparison, GPS altitude is not very accurate, perhaps $\pm 4\ m$ in good conditions, therefore utilizing pressure data improves results over using GPS alone. However, the pressure sensor drifts nearly continuously over longer periods of time, due to atmospheric pressure changes and gusts of wind.

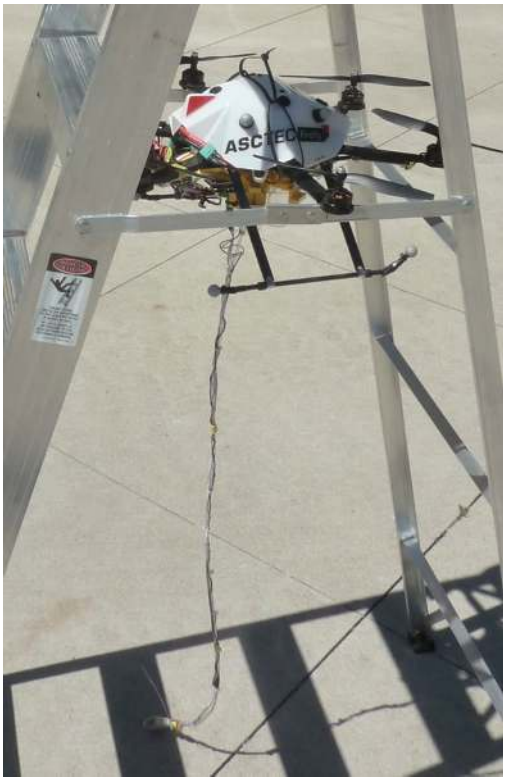


Figure 6.2: UAV at fixed altitude for pressure characterization.

To characterize the pressure sensor drift, we conducted an experiment outside. We turned the UAV on while it sat on the ground to set a baseline pressure altitude reading.

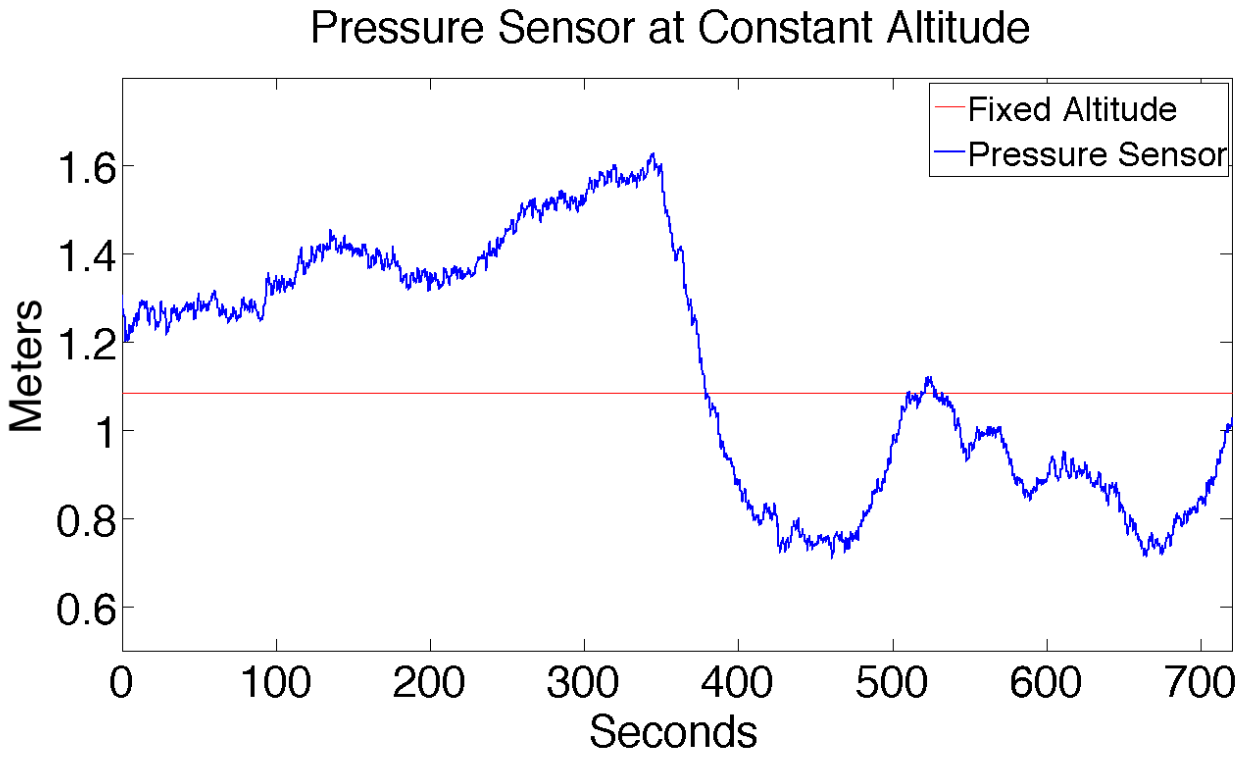


Figure 6.3: Ambient Pressure Drift Over 12 minutes.

Then we lifted it to a fixed altitude and affixed it to a ladder at 1.08 m, to ensure that proximity to the ground did not impact the readings. We recorded the pressure sensor for 720 seconds. At every 60-second mark during these 720 seconds, we recorded the wind speed at the UAV with a hand-held anemometer. It was a calm day, with and average wind speed of 1.3 ms^{-1}.

The results are shown in Fig. 6.3, over the first 300 seconds, the pressure altitude moves up slowly from 1.2 m to 1.5 m. Then suddenly, near 360 seconds, the pressure altitude drops 0.90m in 90 seconds. This steep drop did not appear to be connected to any ambient wind change, which was never observed to be greater than 2.0 ms^{-1}. An altitude drift of 90 cm is more than enough to mean the difference between a wet and dry UAV. This demonstrates that the pressure sensor data is subject to drift enough to disqualify it for low-altitude reliability, even in calm conditions.

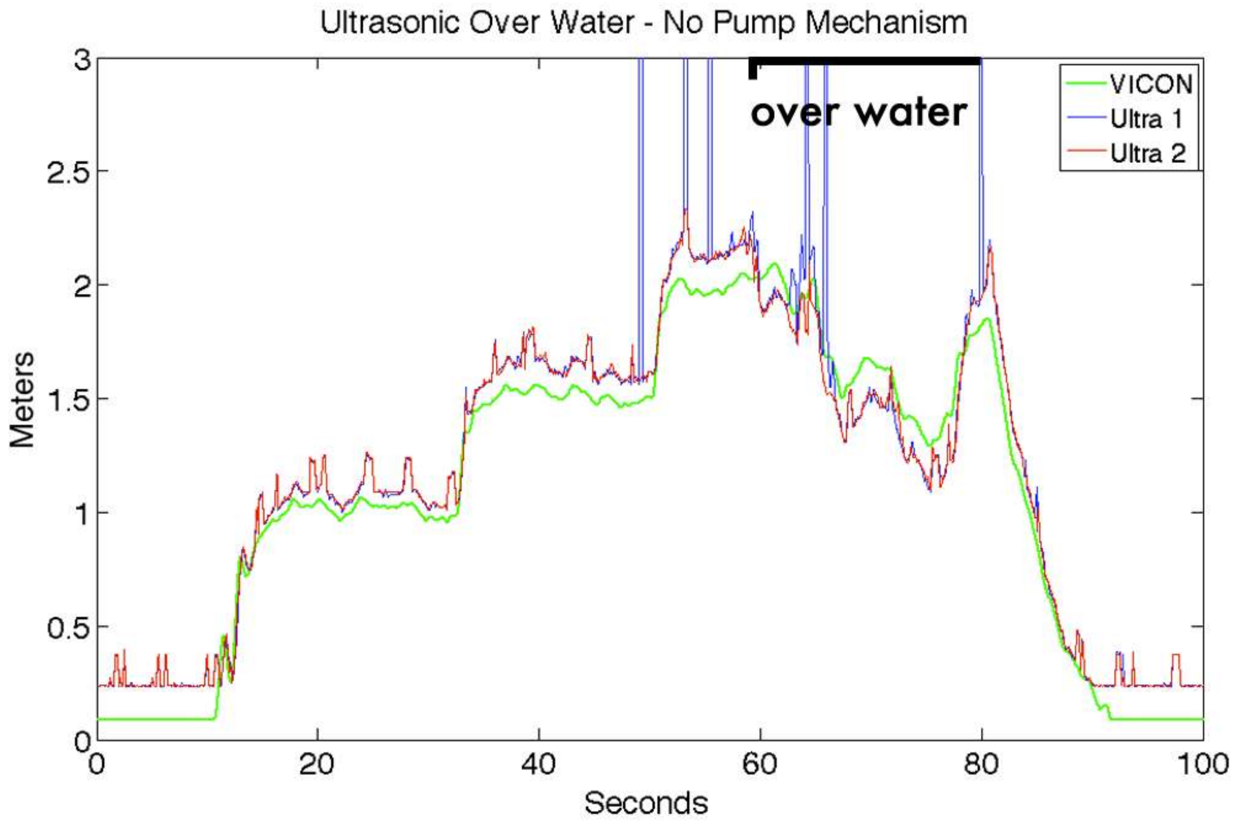


Figure 6.4: Ultrasonic and VICON Altitude.

Since pressure sensor alone is insufficient, we considered a number of other sensor types, including laser and radar. Laser altimeters are attractive, because they can be light enough for UAV applications and used simultaneously for 3-D mapping. However, point lasers read poorly over water and 3-D mapping is a large problem domain and we wanted to focus on water sampling. Radar is becoming more affordable and compact as automotive manufacturers incorporate small, lightweight radar into cars. However, radar is still heavy for a UAV and not accurate enough (± 0.5 m) for our application, as discussed in Sec. 2.4. Therefore we decided to use ultrasonic sensors.

6.1.2 Characterization of Ultrasonic Rangefinders

We took a closer look at ultrasonic sensors because they are not perturbed by lighting conditions or temperature and are designed to work well in the range $0.2 - 2.0\ m$. Also, ultrasonic sensors are necessary because the pressure sensor alone drifts over time due to wind or changes in atmospheric pressure. Further, one high-end ultrasonic sensor is fairly reasonable at \$40 USD, and the manufactures specifies an accuracy of $\approx 1\ cm$.

The ultrasonic sensors detect solid objects, but we were not sure how they would perform over water while flying. Therefore, we characterized the ultrasonic sensors over water by conducting indoor flight tests without the dangling tube and with ground truth from a VICON motion capture system [86]. Our VICON system is specified by the manufacturer as accurate to within $\pm 1\ mm$ and measured at $200\ Hz$. As mentioned in Sec. 4.4.1, we augmented the UAV with two ultrasonic sensors, pointing straight down, flanking the tube at $\approx 10\ cm$ from the center axis. We use two ultrasonic sensors in case the dangling tube occludes one sensor.

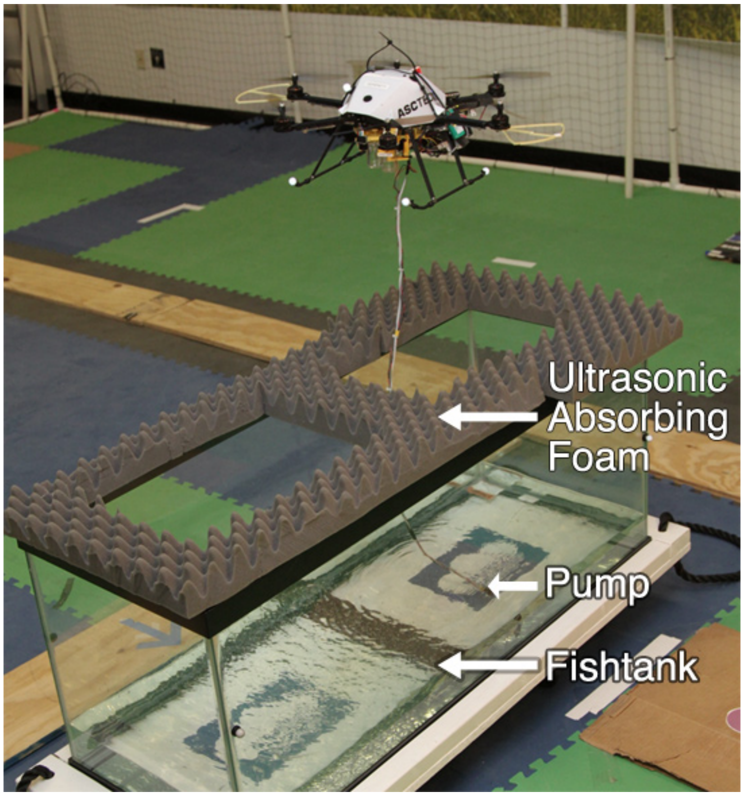


Figure 6.5: Indoor Testbed for Water Sampling.

To simulate being over a river or lake, we purchased a fish tank and filled it with $10\ cm$ of water. We placed acoustic foam over the edge of the fish tank (Fig. 6.5) to absorb the ultrasound waves and ensure the tank is not detected. Then we created a program to fly the UAV over the fish tank so we could compare the ultrasonic to VICON altitudes. A plot of the altitude over 100 seconds is shown in Fig. 6.4. In the first 10 s, the UAV launches to 1 m, then in the next 20 s, the UAV ascends to 2 m, and moves over the water at 30 s. For the next 18 s, the UAV descends to 1.5 m. Notice that the ultrasonic estimates cross above the VICON during the 'over water' period. This happens because the water in the fish tank is 0.21 m above the floor and the ultrasonic sensor measure the distance to the water. As shown in the plot, the ultrasonic readings follow the VICON altitude closely.

The larger spikes, seen six times during the period from 50 to 80 seconds, occur when the sensor fails to get a reading and returns the maximum value, ≈ 7 m. This discontinuity can be caused by the tilt (attitude) of the UAV causing the ultrasonics not to receive a reflected signal. These large spikes are usually brief and rarely affect both sensors simultaneously, so having more than one sensor is important to filter sporadic noisy readings. Also, during this experiment we used a different, older model of Maxbotix ultrasonic sensor which is less resilient to noise. Our current newer ultrasonic sensors almost never have big spikes when sensing distances less than 2.0 m.

As shown in Fig. 6.4, both ultrasonic sensors return small sensor noise every few seconds over the entire run. During prototyping, we used an Arduino microcontroller to power the ultrasonics, and found that noise on the power caused sensor noise. By adding the custom microcontroller with a better power regulator, we greatly reduced this noise.

A more detailed view of the 20 seconds over water is shown in Fig. 6.6 During the time over water, and excluding the two large spikes before 66.25 s, the ultrasonic sensors yield a mean average error of 0.038 m, reasonable values over water. As seen in Fig. 6.6, the ultrasonics closely follow VICON ground truth, although they lag $0.2 - 0.3$ s behind as the UAV changes altitude at $\approx 65 - 67$ s, and again near $68 - 70$ s. The lag is caused

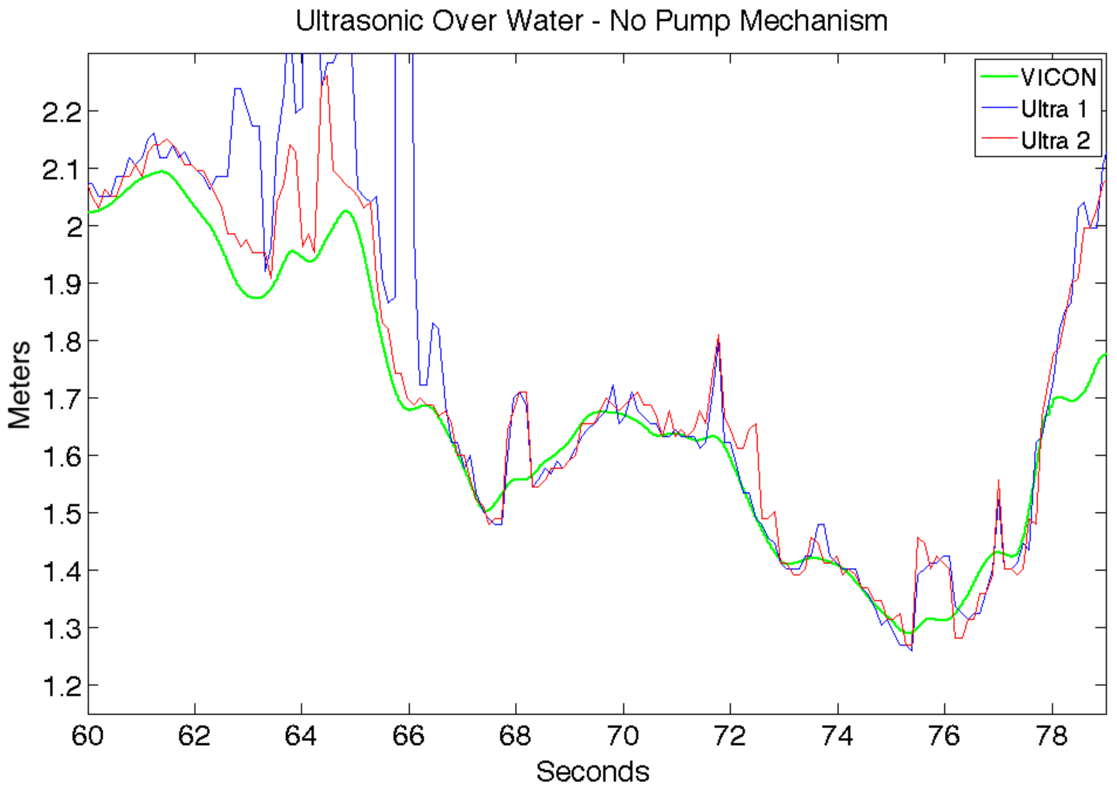


Figure 6.6: Ultrasonic and VICON Altitude Over water.

by the latency of the ultrasonics, as data is received, registered, and available to the system. If it were critical to have higher accuracy while translating in space, then we might have to compensate for this lag. However, this latency is less important for our system since we are most concerned with accurate readings when the UAV is hovering and we limit the descent velocity so that the system has more time to detect the water's surface. As a result of these tests, we learned to cap all altitude values to 1.85 m, the limit of reasonable values during flight. These experiments show that the ultrasonic sensors perform well over water on a flying UAV, especially when the UAV is hovering near the water's surface.

Ultrasonics, therefore, provide good, albeit noisy readings over water. We found the sensor data was reliable and acceptable enough to try to recover the signal information from the noise using filtering.

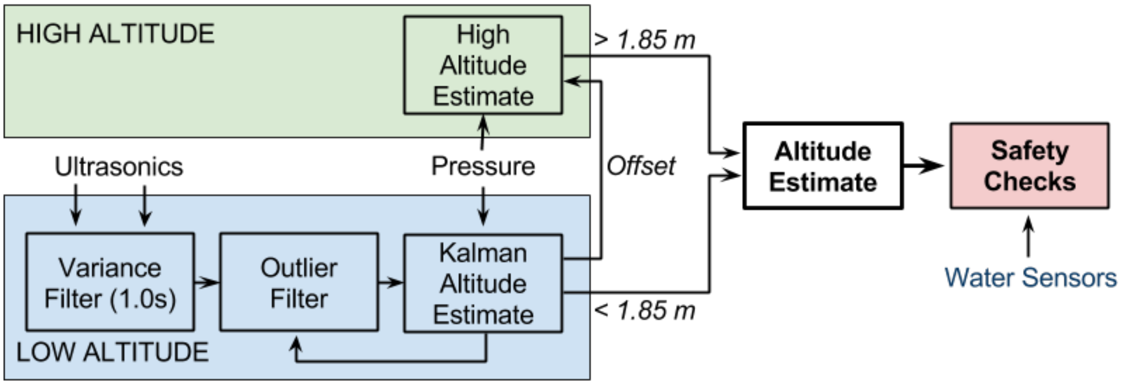


Figure 6.7: Altitude Estimation Information Flow.

6.1.3 Kalman Filter Low-Altitude Estimation

Fig. 6.7 shows our approach to altitude estimation by filtering ultrasonic sensor data and combining it with pressure data through a Kalman Filter. The left side of the figure shows two methods for altitude estimation: high altitude and low altitude. We switch between sensor suites and methods based on whether we are close enough to get reliable ultrasonic data. When the ultrasonic sensors yield readings less than 1.85 m, then they are the most accurate source for altitude. When the ultrasonics read above 1.85 m, the pressure sensor guides the system. If the pressure drift causes the UAV to descend, and the ultrasonic sensors are in range, then the system corrects the pressure offset and the UAV ascends back to the target height. As shown in the figure, the ultrasonic sensor data flows through a series of filters before the final altitude estimate is reported.

Listing 6.1: Sensor Scoring for Kalman Estimate

```
double u1 = ultrasonic_range_msg->UltrasonicFront;
double u2 = ultrasonic_range_msg->UltrasonicBack;
double u1var = getVariance(&ranges,1.0, 1);
double u2var = getVariance(&ranges,1.0, 2);
double proximityThreshold = 0.075;   // heuristic, centimeters
double varianceThreshold  = 0.08;    // heuristic, centimeters^2

// PROXIMITY
if (abs(currentKalmanEstimate - u1) < proximityThreshold) {
    isNear_1 = true; }
```

```
  if (abs(currentKalmanEstimate - u2) < proximityThreshold) { isNear_2 = true; }

  // VARIANCE
  if (u1var < varianceThreshold) { isVarGood_1 = true; }
  if (u2var < varianceThreshold) { isVarGood_2 = true; }

  // INITIAL SCORING
  msg_out.good_1 = 2*isNear_1 + isVarGood_1;
  msg_out.good_2 = 2*isNear_2 + isVarGood_2;

  // CHOOSE SENSOR VALUE WITH BEST VARIANCE AND PROXIMITY -- OTHERWISE AVERAGE
  if (msg_out.good_1 > msg_out.good_2) {
    bestUltrasonicReading_ = u1;
  } else if (msg_out.good_1 < msg_out.good_2) {
    bestUltrasonicReading_ = u2;
  } else {
    bestUltrasonicReading_ = (u1 + u2) / 2;
  }

  // REJECT OUTLIERS
  if (bestUltrasonicReading_ > MAX_range_ultrasonic_z_)
  {
    bestUltrasonicReading_ = MAX_range_ultrasonic_z_;
  }
  // ADD TO KALMAN ESTIMATE
  kalmanCallback();
  }
```

As shown in code listing 6.1, we perform a scoring algorithm to choose the best value each time new sensor information is received. Since we have two sensors each of which might have useful information, we devised a scoring method to choose one sensor's reading over the other. The sensor with the highest score is preferred. This scoring method is used throughout the series of filters, as follows: A variance filter, which calculates the variance σ^2 during the last second, and if the sensor reading has a variance below a heuristic threshold (0.08), then this sensor gets a point. We filter by variance because we noticed that when the ultrasonic ranges are noisy, they tend to exhibit high variance over a short period of time. The second filter is the outlier filter, which determines if the current reading is within a certain threshold (0.075 m) of our estimate. If so, then this sensor gets two points. Since we assume the system is in a stable, nearly hovering state, we give preferences to readings closer to our current estimate. In case of a tie, the readings are averaged. If the best possible reading is greater than the maximum allow value, then the max value is used instead of the sensor reading. Readings might suddenly go to the maximum when the vehicle tilts relative to the water surface and the reflected signal is not detected. We then add this value to the Kalman Estimate, which we discuss next.

We use a plain Kalman filter, a kind of Bayesian filter on continuous systems, to improve our altitude estimation. The Kalman filters and predicts sensor values. There are two sets of equations [58]: the time update in Eqs. 6.1-6.2 and the measurement update Eqs. 6.3-6.5. The time update equations model change in the physical system that occur between sensor readings. The measurement update models the likelihood of the current sensor measurement give our current estimation. The key part of the Kalman filter is during the measurement update and is called the *innovation*, which is the difference between the predicted value $H\hat{x}_t^-$ and the measurement z_t, in Eq. 6.4.

We use a simplified formulation of Eqs. 6.1-6.2, without modeling the control input. We get good results without modeling the control input because we assume the system is nearly hovering at low altitude.

We begin by defining and initializing the variables:

INITIALIZATION

$$\hat{x}_t^- = \begin{bmatrix} pressure \\ ultrasonic \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$P_t = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} \text{Covariance}$$

$$Q = \begin{bmatrix} 0.0007 & 0.0 \\ 0.0 & 0.00005 \end{bmatrix} \text{Process Noise}$$

$$H = \begin{bmatrix} 1.0 & -1.0 \\ 1.0 & 0.0 \end{bmatrix} \text{Combination}$$

$$R = \begin{bmatrix} 0.15 & 0.0 \\ 0.0 & 0.004 \end{bmatrix} \text{Gain Weighting}$$

TIME UPDATE

$$\hat{x}_t^- = \hat{x}_{t-1} \tag{6.1}$$

$$P_t^- = P_{t-1} + Q \tag{6.2}$$

MEASUREMENT UPDATE

$$K_t = P_t^- H^T (H P_t^- H^T + R)^{-1} \tag{6.3}$$

$$\hat{x}_t = \hat{x}_t^- + K_t (z_t - H \hat{x}_t^-) \tag{6.4}$$

$$P_t = (I - K_t H) P_t^- \tag{6.5}$$

These equations run every time we have new information, from either the ultrasonic sensors (which arrive as a pair) or from the pressure sensor. Using this Kalman filtered estimate, we designed an experiment to test the altitude over water using the aforementioned fish tank and **with** the dangling pump attached. Fig. 6.8 shows the VICON position (ground truth), the ultrasonic readings, and the Kalman estimate while the vehicle is flying over water with the dangling tube and using the Kalman Estimate as the control input to the altitude controller. Note how although the ultrasonic sensors yield less-frequent results which have much less fidelity than the VICON position, the Kalman

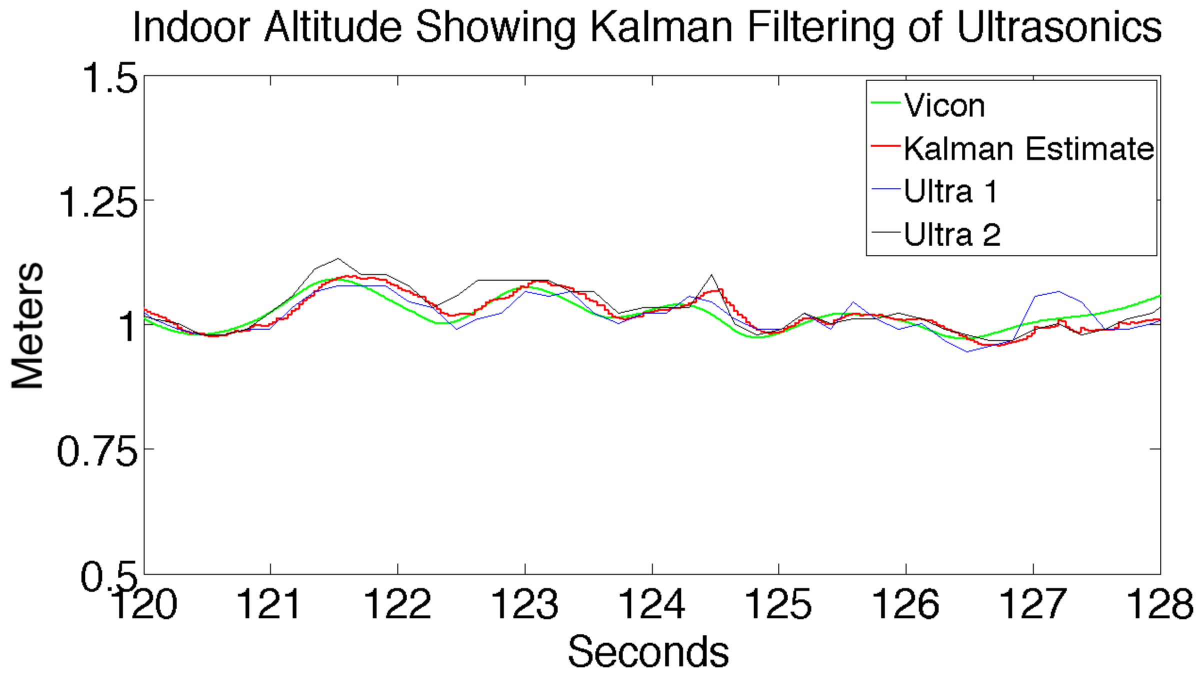


Figure 6.8: Kalman-filtered Height Estimate With Vicon and Ultrasonics

estimate tracks the VICON position much more accurately since it also incorporates the pressure sensor readings, which are not shown in Fig. 6.8 because they have drifted more than 0.75 m. By using the Kalman filter, we estimate an altitude similar to VICON using only onboard sensors. The Mean Average Error (MAE) during the time period shown in Fig. 6.8 is 0.017 m, nearly as accurate as the ultrasonic sensor accuracy of 0.01 m. While its rare to have faulty readings from both sensors, experimentally we determined that even if there is continuous faulty data from the ultrasonics, the Kalman estimate quickly converges to a good estimate once a single sensor yields accurate readings.

6.1.4 Final Altitude Estimate

Although the Kalman estimate is the best information when the ultrasonics are in range, the overall altitude estimate is governed by a higher-level process, as shown in Fig. 6.9. The final altitude estimate uses the Kalman estimate at low altitude and the pressure sensor with an offset at high altitude. At low altitudes, the Kalman estimate is accurate enough to assure vehicle safety, while at high altitude, the pressure sensor is sufficient and if sensor drift forces the system below 1.85 m, the low-altitude controller will take over. Since the pressure sensors initializes to 0.0 m when it is turned on, if we send the UAV lower than the starting height, which can happen if we launch from a hilltop and then sample down in a valley, then the altitude can go negative as the UAV descends. Then, when the UAV is within 1.85 m of the surface, the low altitude controller takes over and the altitude is suddenly higher. This can cause a discontinuity in the altitude, which we handle by adjusting the target waypoints at the same time the altitude is changed, so that the vehicle continues to the target sampling height. The discontinuity that is possible during descent is not possible during ascent, because the altitude reading from the pressure sensor is offset by the last Kalman estimate from the ground. This ensures that whenever we have good information about the ground, we use it to improve the altitude estimate when we ascend. Anytime the vehicle transitions from low to high altitude, the pressure sensor is offset with the last best estimate from the Kalman Filter. When descending, we limit velocity so that the UAV can stop before

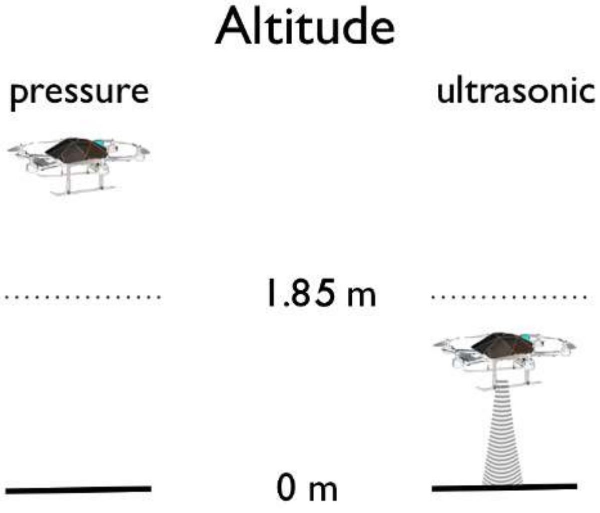


Figure 6.9: Altitude estimation at low and higher altitudes.

coming within one meter of the water.

Listing 6.2: Water sensors safety code

```
def getWaterSensorConsensus(self, level):
    m = self.water_sampler_board_processed_msg
    if level == 1:
        return m.H2O_1
    if level == 2:
        return m.H2O_1 + m.H2O_2
    if level == 3:
        return m.H2O_1 + m.H2O_2 + m.H2O_3
    if level == 4:
        return m.H2O_1 + m.H2O_2 + m.H2O_3 + m.H2O_4
    if level == 5:
        return m.H2O_1 + m.H2O_2 + m.H2O_3 + m.H2O_4 + m.H2O_5

def WaterSensors_check(self):
    m = self.water_sampler_board_processed_msg
    # CHECK WATER SENSORS - WITH 4 BEING THE SECOND TO TOP
    if (self.getWaterSensorConsensus(4) > 2) and m.H2O_4:
      print "Water Sensor Warning"
    # TOP SENSOR
    if (self.getWaterSensorConsensus(5) > 3) and m.H2O_5:
      print "Water Sensor Abort"
      self.triggerAbort()
```

Code listing 6.2 shows how we enforce additional safety checks with the water sensors on the tube. If the water sensors indicate that the tube is too deep, then the UAV ascends to a safer altitude. The routine **getWaterSensorConsensus()** checks how many sensors are wet, with level 1 being the lowest sensor at the pump. Therefore, when sensor 5 is wet, the vehicle is too low and the system triggers an abort. With our redundant sensors, we have yet to see this abort mechanism trigger in the field. The water sensor data is not directly added to the Kalman Filter both because they are slow (0.5 s) and also because water droplets from the pump occasionally cause false readings. We validate this approach with field experiments in Chap. 7.

6.2 Sampler Effectiveness Experiments - Indoor

Now that we have an accurate method for estimating altitude, we can use this information to verify that the water sampling mechanism works effectively while using this altitude. We built a water sampling system, including a pump, tube and vials, but before we took the system into the field, we wanted confidence that the system would work as intended. Therefore, we designed an experiment that compared the success rate of the sampling system while changing altitude control from VICON to the ultrasonics with Kalman-filtering. In these experiments, the only thing we changed was the source of the altitude information, the x and y positions were provided by VICON. In either case, we ran a script or pre-programmed set of actions in which the UAV turns on, takes off and flies to the fish tank to obtain three water samples, and then lands. A 'Sample' event means the UAV descended from 1.85 m to 1.0 m, submerged the tube into the water, and pumped. After each sample the script directed the UAV to ascend back to 1.85 m. By repeating this script, and counting the number of vials that come back completely full, we built an expectation of the likelihood of successful sampling with VICON altitude control vs. ultrasonic control. We counted as 'full' every vial that was full of water at least to the 'neck' of the vial. We also noted if the vial was at least half full, or less than half full.

We repeated each method 15 times, three samples per trial, for a total of 45 samples by VICON and 45 samples by ultrasonic control, for a grand total of 90 indoor samples. Each trial took 4-5 minutes flying, with an additional 5-10 minutes to set up the system, empty the vials, and periodically change batteries. The results are shown in Table 6.1. Notice that the VICON and ultrasonic success rate is within 2.2% of each other, and that the sampling system returns a full vial about 9 times of 10. Because the success rates are so similar, we infer that the reason 1 of 10 is **not** full is independent of the method of estimating altitude. Of the nine unfilled vials, 66% were at least half full. Of those

Table 6.1: Sampling Success Rate Indoors

Altitude	Trials	Samples	Full	$> \frac{1}{2}$	$< \frac{1}{2}$	% Full
Vicon	15	45	41	3	1	91.1
Ultrasonic	15	45	40	3	2	88.9
Total Indoor	30	90	81	6	3	90.0

that did not fill, we believe the timing on priming the pump is the primary fault. We wait 400 *ms* after the pump is wet until we turn on the power. However, occasionally the pump still does not prime, and perhaps in the future we will add sensors to detect the presence of water in the vials, and use this information to diagnose when the pump does not prime. If the sampling success rate is the same for VICON and ultrasonic method of estimating altitude, and we can use the ultrasonic method outdoors, then we can move the system to outdoor field trials, since we believe a 90% success rate is sufficient for this first version. This experiment demonstrates that the water sampling subsystem is more than 90% effective using ultrasonic-based altitude.

In addition to these experiments, we performed indoor sampling an additional 26 times during demonstrations with a 92% success rate.

6.3 Summary

In this chapter we have demonstrated the effectiveness of two critical subsystems: ultrasonic altitude estimation and the water sampling mechanism. We tested these subsystems in controlled, laboratory environments where we could isolate the variables and test them independently. By establishing accurate altitude without a motion capture system and effective water sampling, we are ready to test the whole system outdoors, in the field in a less structured environment.

Chapter 7

Field Experiments

Field experiments validate the principles and theories developed in the lab by exercising them in real-world conditions. Field tests are critical to convince yourself and the community that your ideas are viable. This is especially true in field robotics, where we want to develop systems which have addressed or at least encountered all the challenges in the gap between theory and our proposed solution. Our indoor testing methodology is designed so that tests directly address the challenges of an outdoor, semi-unstructured environment, but they are still not a replacement for *in situ* tests.

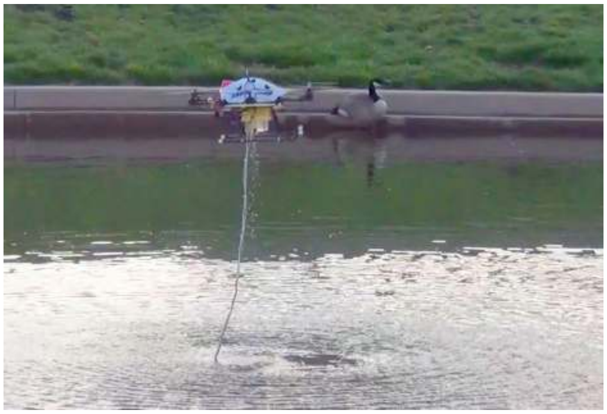


Figure 7.1: Sampling at Antelope Creek, Lincoln NE

To test the system outside, we designed an experiment which repeated the sampling methodology used in the lab, only replacing VICON position in x and y with GPS. We then select outdoor locations which are representative of the kinds of places where we might sample, but which simplify the challenges by minimizing obstacles and allowing

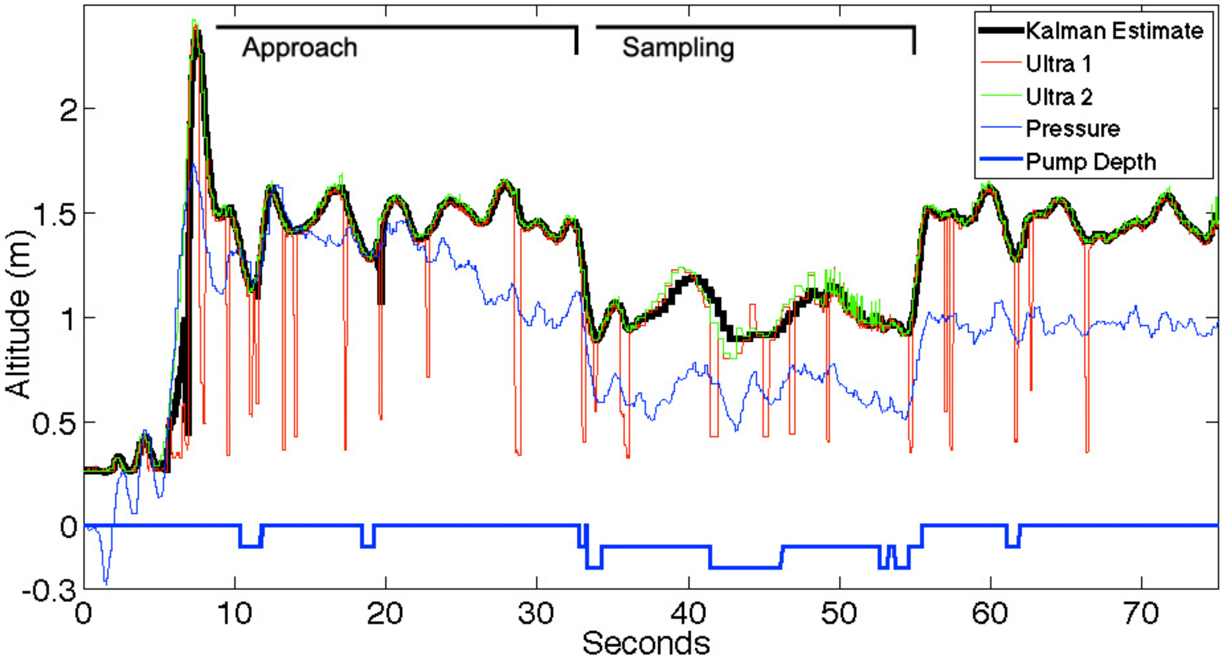


Figure 7.2: Vehicle Altitude and Pump Depth While Sampling Outdoors.

easy take-offs and landings. This chapter describes the field experiments used to validate the altitude estimation in Sec. 7.1, effectiveness of the water sampling system in Sec. 7.2, and to verify that the water sampling system does not induce bias in the water properties that scientists wish to measure in Sec. 7.5. We discuss the causes and solutions to a UAV water crash in Holmes Lake, Lincoln, NE in Sec. 7.3. In this chapter we also describe a field demonstration we performed at the Blue Oak Ranch Reserve near San Jose, California in Sec. 7.4.

7.1 Outdoor Altitude Estimation

In Chap. 6, we validated our water sampling system and tested a method for estimating altitude over water in the lab We use the same methodology now to test the system in the field. To compare the altitude estimation versus indoor experiments, we recorded the ultrasonic, pressure sensor, and Kalman-filtered height estimate, as shown in Fig.7.2.

During this experiment the UAV always flew at low altitude ($< 2.5m$). This figure shows the UAV while it ‘approaches’ the sample destination and the critical ‘sample’ stage when the UAV descends and maintains altitude to pump water. Compared with altitude tests indoors, the ultrasonic sensor readings had more spikes, indicating additional noise but the dual ultrasonics still allowed for successful altitude control. The noise from Ultrasonic 1 in Fig. 7.2 is an extreme example, as there was faulty cabling. However, the altitude estimate tracks in spite of this noise.

The figure also shows the depth of the pump, as detected by the water conductivity sensors on the tube. Both the first and second conductivity sensors activated during sampling, but never the ones above. We noticed that the water sensor skimmed the surface as the UAV approached the sample location, which is reflected in Fig. 7.2. During the outdoor altitude tests, we observed a larger variation in x and y during sampling due to GPS inaccuracy, which impacts height as the UAV tilts as it tries to adjust its location. These tests confirm that our filtered altitude estimate works well at near proximity to water in calm conditions. Future tests will stress the system with higher winds and waves.

7.2 Outdoor Water Sampling Effectiveness

We repeatedly exercised the sampling system with a trial consisting of three consecutive sampling events at one location. After the vehicle samples, we examined the vials and recorded the quantity of water in each vial. Any quantity at or above the ‘neck’ of the vial is full, less than the neck but more than half full is ‘more than half full’, and anything less is ‘less than half full.’ We intended to run the trials 15 times as we did indoors, but we ran out of batteries and were only able to conduct 13 trials.

For the initial field tests, we sought a location with few obstructions, shallow water, easy access, and a good place to take off and land. Fortunately, there is a location close to the University of Nebraska that met those needs. We chose a human-made waterway along Antelope Creek in Lincoln, Nebraska, USA. The water feature has a flat, concrete patio adjacent to it, and the depth at this location is $1-2\ m$ deep. The water is fresh and clear, and this depth ensured that we would be able to retrieve the vehicle in case of catastrophic failure. Additionally, this site was chosen in part because the concrete area allows access to the water with little loss in altitude. Fig. 7.1 depicts the system operating outdoors at the Antelope Creek location.

We performed outdoor experiments to test the effectiveness of the sampling system when controlled autonomously over water. We programmed the system to navigate to GPS waypoints and obtain three samples. The results of this test are shown in Table 7.1. The success rate for fully filled vials was 69%, with 7 of 12 failures caused by a faulty lid mechanism that we have now fixed. Three of the remaining five "failures to fill" occurred on the third vial when the backup pilot took over control after perceiving that the UAV was trending too close to water, especially as the wind increased during the experiment. We believe pilot aborts will occur less frequently in the future as we improve hover stability in gusty conditions and as the safety pilot's confidence increases. Thirteen total sample trials were conducted, until all available batteries were discharged. Overall, within the wind and environmental constraints, the system demonstrated the ability to maintain altitude and retrieve samples.

Table 7.1: Sampling Success Rate Outdoor with Grand Totals

Altitude	Trials	Samples	Full	$> \frac{1}{2}$	$< \frac{1}{2}$	% Full
Outdoor	13	39	27	4	8	69.2
Total Indoor	30	90	81	6	3	90.0
Grand Total	43	129	108	10	11	83.7

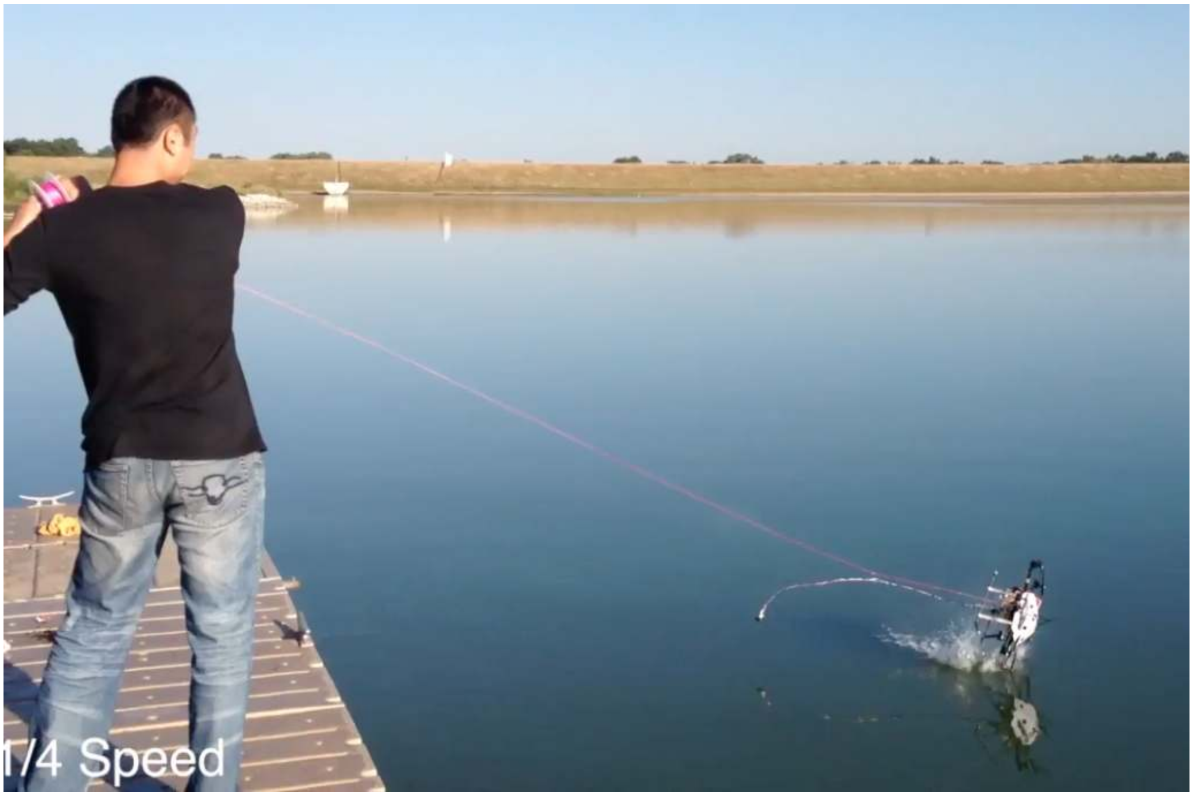


Figure 7.3: Water Crash at Holmes Lake

7.3 Water crash at Holmes Lake, Lincoln NE

As shown in Fig. 7.3, we gained valuable experience from a water crash into Holmes Lake on 3 Sept. 2013. The crash revealed a fault in the low-level control, which reported ″MOTORS-ON″ while one motor was stuttering on start. In our launch sequence, our control system commands the UAV to start it's motors, and then waits for the response "MOTORS-ON" from the UAV. Once the UAVs motors are on, we start sending 'stick commands' to increase thrust, and the UAV takes off. In this instance, our control system started sending thrust commands before all the motors were ready. One motor provided no thrust, and the imbalance between the forces accelerated the UAV in a sharp horizontal path, directly into the water.

Although the Firefly only requires five of six props for flight, the AscTec algorithm for reduced motor flight does not apply to startup. We retrieved the vehicle immediately from the fresh water, and after disassembling and thoroughly drying the vehicle, we rebuilt it and it has worked without failure since. From the crash we learned several lessons: 1) Introduce a delay after ”MOTOR-ON” to give the motors more time to spin up before adding thrust; 2) Start further away from water, so that launch problems result in trajectories that are confined to the ground; 3) Warm the motors up with more test flights, especially on cool mornings; 4) The safety tether is useful for more than keeping the UAV from flying away; and 5) Our UAV platform **can** survive total immersion with the battery on, if it is recovered quickly. We had the same system in the field over water within two weeks.

7.4 Demonstration at Blue Oak Ranch Reserve

We conducted an additional field demonstration at the Blue Oak Ranch Reserve (BORR) near San Jose, California, as shown in Fig. 7.4. The purpose of the demonstration was to show our collaborators from the University of California Berkeley how the system behaves at their field station. The BORR is part of the University of California reserve system, and was cattle-grazed until 1990, and has many small ponds. The water source of these small ponds is not fully understood, whether from rain run-off or from groundwater sources. Hydrologist Dr.

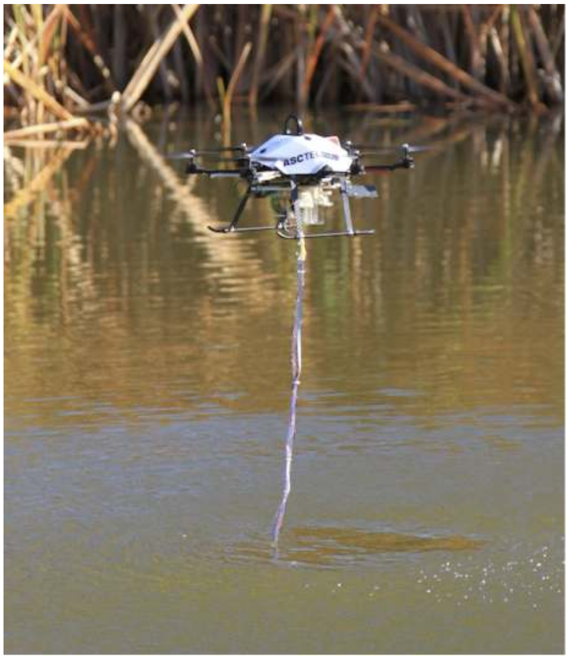


Figure 7.4: Sampling at the Blue Oak Ranch Reserve, near San Jose, CA, USA.

Sally Thompson of UC Berkeley envisions autonomous water sampling augmenting an existing wireless sensor network installed across the BORR. Regular water sampling combined with isotope analysis would help determine the source of water feeding these ponds. The current sensor network measures humidity, temperature, wind speed, and provides video feeds from the sensor location. Further, the ponds suffer from invasive frogs and algal blooms.

This demonstration utilizes the altitude controller that switches from high-altitude 'pressure-sensor' mode to low-altitude 'ultrasonic-sensor' mode during descent. This allows the vehicle to fly a pre-programmed 'script' which flies high up over obstacles (vegetation near shore), and then descend down to the water's surface for sampling. The demonstration took place in the morning of 21 November 2013 for an audience of Dr. Michael Hamilton, Dr. Sally Thompson, Dr. Sebastian Elbaum, Dr. Carrick Detweiler, Mr. Eric Viik, and the author. Two flights of three samples each were conducted over a small pond, approximately $15m$ by $40m$. Wind speeds of $5m/s$ were measured with a hand-held anemometer. Both flights returned three full sample vials.

7.5 Water Science

Separate from altitude estimation and sampler effectiveness, and of particular importance to our limnologist collaborators, is validating that the system introduces no bias in water properties. Potential differences include those caused by pumping, transit through the tube, agitation during flight, and changes in water properties during the delay between sample acquisition and sample measurement on land. We conducted an experiment to ensure that water samples collected by the UAV-mechanism exhibit similar water chemical properties as samples obtained through traditional hand sampling methods. The UAV was not flown, but rather held by a human operator in a kayak to ensure that both the

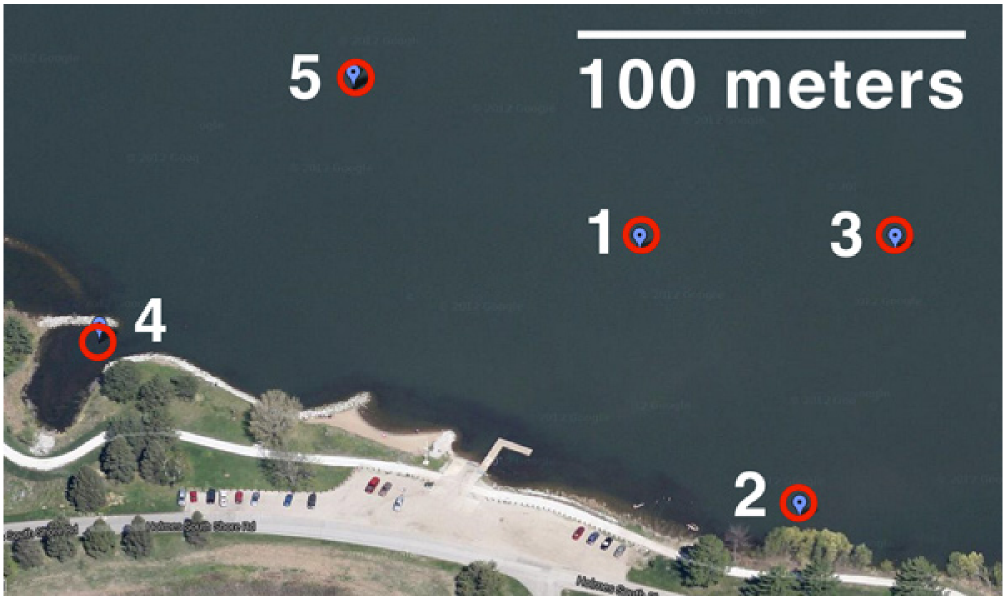


Figure 7.5: Holmes Lake Water Property Sample Locations

hand and UAV samples were taken at the same time and place. The human operator collected samples using both the UAV mechanism and the 'grab sampling' technique.

We sampled at five locations on Holmes Lake, Lincoln, NE, USA. We collected two samples near shore and three closer to the middle of the lake, as shown in Fig. 7.5. At each location, we took three samples by hand and three with the UAV-mechanism for a total of fifteen samples by each method. Overall it took approximately 2 hours to collect this data due to the time to kayak, collect manual and UAV-mechanism samples, and to perform some on-site analysis and filtering. We estimate that collecting the same samples with the flying UAV would take 20 minutes.

At each location we measured temperature, dissolved oxygen[1] (DO), sulfate, and chloride. By sampling both a dissolved gas and representative ions we can assess the suitability of the UAV-mechanism for scientific water sampling. Our collaborators, experienced water scientists, measured temperature and at the sample location for

[1] For DO and temperature a single reading was obtained with the hand sensor at the location, but for the UAV-mechanism it was tested on each of the three samples.

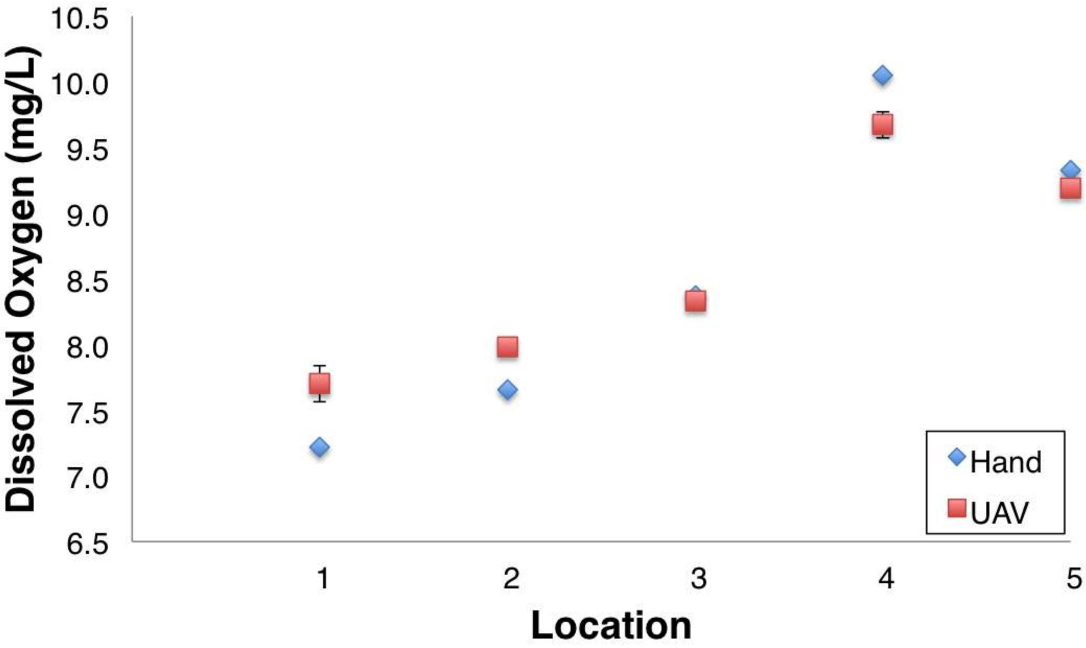


Figure 7.6: Dissolved Oxygen

the manual measurements and at shore for the UAV-mechanism samples, since these properties change rapidly. Chloride and sulfate ions are measured in the lab using equipment [2] that is not easily portable and these properties don't change rapidly after sampling and filtering.

7.5.1 Dissolved Oxygen

We measured DO because it is a key indicator of biological activity and because we suspected the UAV-mechanism might bias the measurement through degassing during pumping or continued photosynthesis during transit. Sulfate and chloride ions occur naturally in most water and their ratio in freshwater can indicate proximity to a saltwater source. But inland, chloride comes from many sources including lawn fertilizers and road salt.

[2] Lab measurements use a Dionex Ion Chromatograph AS14A, made by ThermoFisher

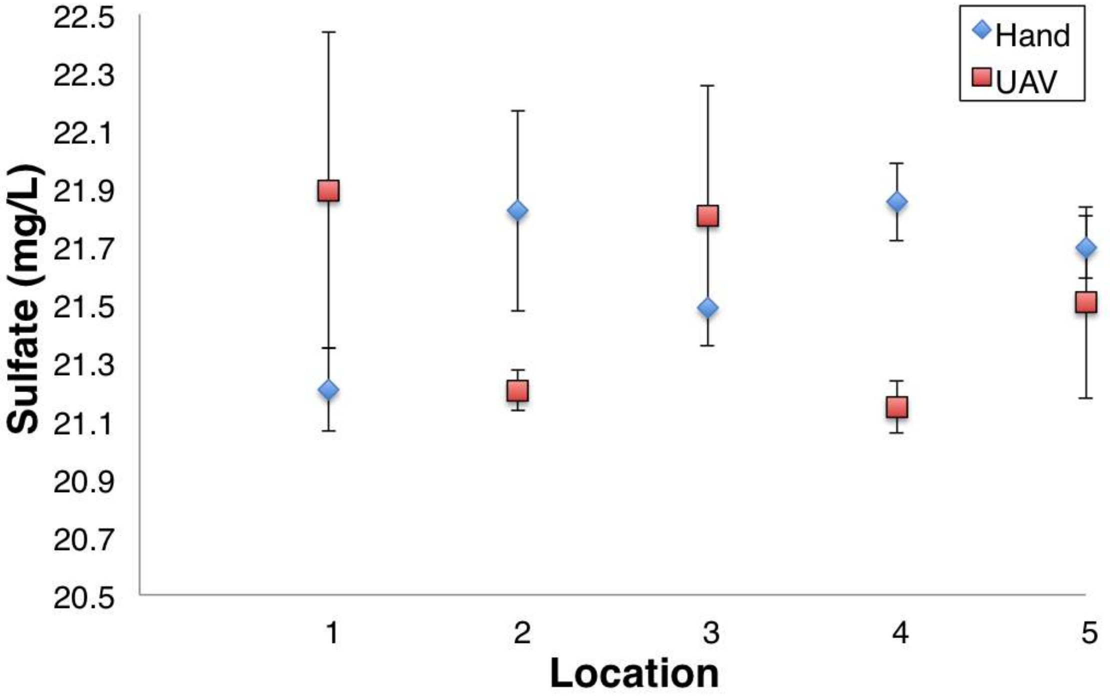


Figure 7.7: Sulfate

We are primarily interested in verifying that the UAV-mechanism does not induce a bias in the measurements. Fig. 7.6 shows the DO as measured by hand at the location and with the UAV-mechanism. The values at the five sample locations are close and show the same general trend in all five locations, implying that the UAV-mechanism and delay (longer by kayak than by flying) has little impact on the DO. Also visible in this figure is the general upward trend between the sample locations. Our collaborators suggest this was caused by increased photosynthesis over the two hours of data collection, although sample location may also play a role in this variation. For instance, location 4 is probably higher than the general trend because it is closer to an enclosed bay and therefore likely to have more plants near the surface.

Obtaining samples quickly and at higher spatial resolution by UAV could help to disambiguate these factors.

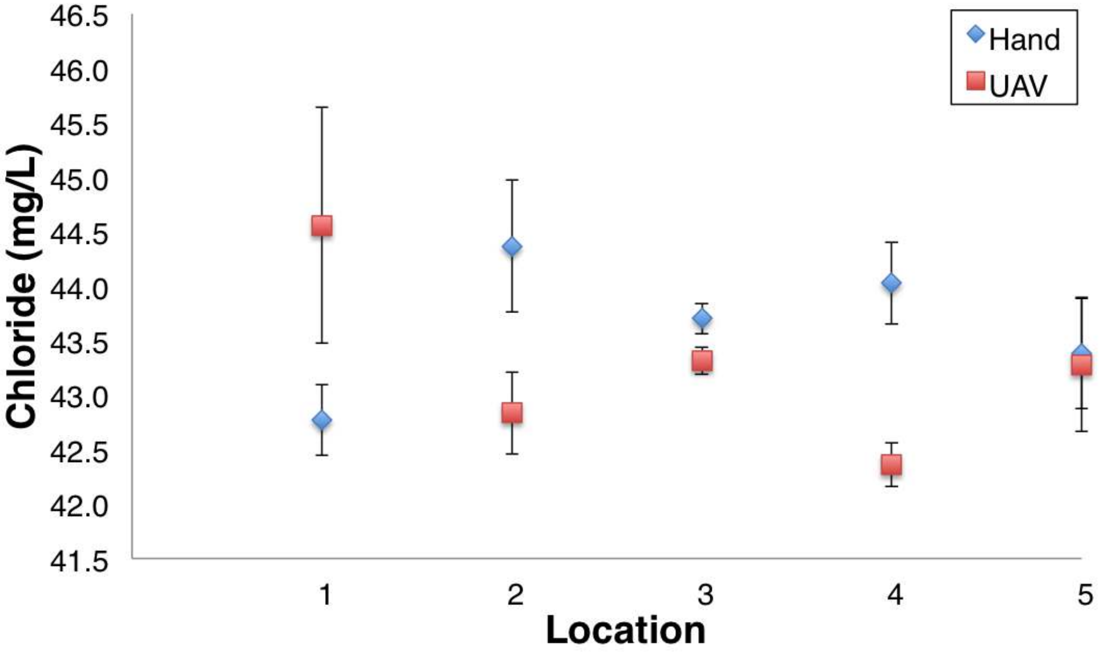


Figure 7.8: Chloride

7.5.2 Dissolved Gasses: Sulfate and Chloride

Sulfate and chloride concentrations shown in Fig. 7.7-7.8 revealed some differences between hand methods and the UAV-mechanism. These differences, however, can likely be attributed to typical sampling variation and neither indicates a strong bias induced by the UAV-mechanism. Further, the typical range for sulfate in lakes is between $10 - 60\ mg/L$ [87] and for chloride varies seasonally but usually is between $10 - 100\ mg/L$ [88], so the observed variation is minimal. We plan to perform additional field and lab tests to verify that these measurements are unbiased.

7.5.3 Temperature

Temperature is the sole exception in the water properties in that it shows a clear bias by the delay induced by recovering the sample by UAV. In contrast to the other measurements, Fig. 7.9, shows that the temperature measured by hand at the sample location is nearly

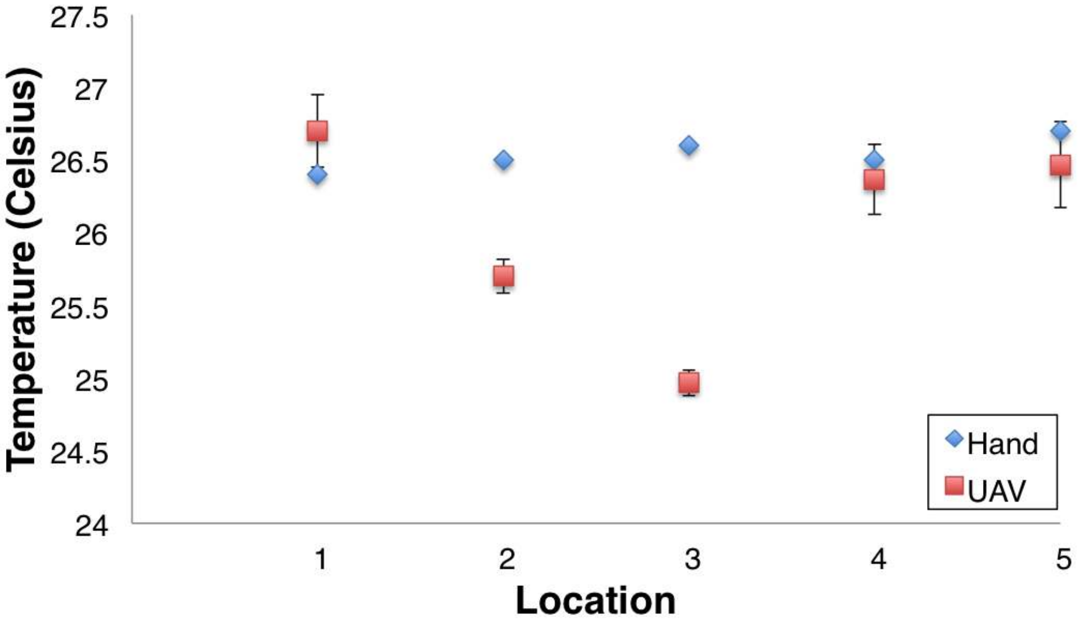


Figure 7.9: Temperature

constant, while the temperature measured in samples from the UAV-mechanism changed during transit, especially at locations two and three. Future versions of system should measure water temperature at the sample location by mounting a temperature probe at the end of the pumping tube. We believe measuring temperature will be easy because temperature sensors are small, light, and fast.

7.6 Summary

These experiments validate our altitude estimation method for near-water flight outdoors, and show that our system can successfully acquire water samples from scientifically important field locations. The UAV system greatly reduces the effort and time to collect samples. These experiments demonstrate that the UAV-mechanism can collect samples that replace those collected be hand. This permits water scientists to obtain more samples within a single lake or river to develop a high-resolution map, for instance, after a rainstorm to identify the source of the influx of chemical or biological contaminates. In addition, reducing the collection time is critical since many water properties, such as DO, fluctuate within hours and using our UAV system would reduce collection time by nearly an order of magnitude.

Chapter 8

Conclusion

Water sampling is a key activity in effectively managing our fresh water resources and maintaining public health. Developing approaches and systems for efficient and effective water monitoring will increase in importance over the coming decades. Co-robotic techniques, where human scientists and robotic systems work together, can help meet future water monitoring needs by combining robot efficiency with human expertise in the field. Field scientists require a system that carries multiple samples, is light and small, reliable and safe, and does not bias sensor information. The cost of the sensors and water sampling subsystem is reasonable and as the price of the aerial platforms descends, as it is expected to as mass-manufacturing increases, the cost of the overall system will be affordable to water scientists. These requirements guided our efforts, and our contributions address these requirements. We have built a system, consisting of an electromechanical system for taking samples and a software control system to guide the entire flow of activities.

The current system design is scalable and could carry more samples or larger quantities of water in each sample. The present design uses only 300 g of the total 600 g payload, and therefore several additional samples could be added to the current design with small modifications. Carrying much more water (0.5 *L* or more) would require a larger UAV, which will make the overall system harder for a single scientist to carry. Otherwise, the current design could be scaled up to carry 1 or more liters of water. Another way the system could be scaled is for one ground station to control multiple vehicles. The limiting factor is the number of simultaneous radio connections and the computationally expensive parsing of serial connections, but the control software uses only $< 10\%$ of the CPU on a 2.3 *GHz* laptop computer, much of which is due to the computational demands of writing a debug file to disk. Overall, the system is immediately scalable to twice the capacity (6 20 *ml* vials or 3 40 *ml*) with straightforward modifications, and further scalable with some re-engineering.

In this thesis, we demonstrated a novel mechanism for sampling water autonomously from a UAV that requires significantly less effort than existing techniques and is nearly an order of magnitude faster. We discussed how a UAV aerial water sampler has a wide range of potential applications, from pure science to humanitarian endeavors. We developed a Kalman filter-based approach using ultrasonic sensors to form an altitude estimate near water with a mean average error of 0.017 *m*. Using this altitude estimate, the system can safely fly close to water and collect three 20 *ml* samples per flight, yielding a full vial 90% of the time. We have developed an embedded system to control the on-board water sampling system, and control software to autonomously sample water at GPS locations specified by field scientists. We verified that the water properties of the samples collected by the UAV match those collected through traditional manual sampling techniques. This shows that this system can be used by water scientists to develop high-resolution maps and improve the spatiotemporal quality of water sampling data sets. Finally, in addition to lab tests, we performed a number of *in situ* experiments with our science partners.

Our future efforts include further operation and evolution of the system in the field, especially in the presence of varying wind speeds and wave sizes, as well as with moving water. We intend to explore ways to enable control of the sampling depth. We are in the process of implementing and evaluating the usability of a user interface for the limnologists and non-expert operators that balances manual control with autonomous behavior with the goal of maintaining system and operator safety. We intend to provide tools for end-user-programming so that the scientist can guide and customize the behavior of the system. We also intend to explore how this platform might be used with adaptive sampling, and in combination with other sensing and sampling mechanisms deployed in bodies of water. We plan to examine the duration of the 'flushing' phase with our collaborators to ensure clean samples. Further, we would like to push some water analysis onto the platform to avoid collecting samples that do not meet required criteria. In addition, we will explore a line of inquiry pertaining to operational safety, as these systems are intended to be reliable tools in the hands of field scientists. We also plan to conduct a risk analysis by repeating representative missions many times and recording the quantity and modes of failure. This risk analysis will quantify the reliability of the system, with the caveat that not all kinds of failure are foreseeable. Finally, we have received approval from the US Federal Aviation Administration for our Certificate of Authority to conduct larger-scale outdoor tests at critical test sites identified by water scientists.

Bibliography

[1] D. S. Pilliod, C. S. Goldberg, M. B. Laramie, and L. P. Waits, *Application of Environmental DNA for Inventory and Monitoring of Aquatic Species*. Fact Sheet FS 2012-3146 Corvallis, Oregon, U.S. Geological Survey, 2012. (document), 3.1, 3.4

[2] W. K. Dodds, W. W. Bouska, J. L. Eitzmann, T. J. Pilger, K. L. Pitts, A. J. Riley, J. T. Schloesser, and D. J. Thornbrugh, "Eutrophication of U.S. freshwaters: Analysis of potential economic damages," *Environmental Science & Technology*, vol. 43, pp. 12–19, Jan. 2009. 1

[3] E. Corcoran, C. Nellemann, E. Baker, R. Bos, D. Osborn, and H. Savelli, "Sick water? the central role of waste-water management in sustainable development," in *United Nations Environment Programme, UN-HABITAT*, Jan. 2010. 1

[4] E. The United Nations Children's Fund (UNICEF)/World Health Organization (WHO), Johansson and T. Wardlaw, "Diarrhoea: Why children are still dying and what can be done," in *WHO Library Cataloging-in-Publication Data*, Jan. 2009. 1

[5] F. D. Wilde, D. B. Radtke, and G. S. (US), *National Field Manual for the Collection of Water-quality Data: Field Measurements*. US Department of the Interior, US Geological Survey, 1998. 1, 3.1

[6] A. J. Erickson, P. T. Weiss, and J. S. Gulliver, “Water sampling methods,” in *Optimizing Stormwater Treatment Practices*, pp. 163–192, Springer New York, Jan. 2013. 1, 2.1

[7] M. Dunbabin, A. Grinham, and J. Udy, “An autonomous surface vehicle for water quality monitoring,” in *Proc. Australasian Conference on Robotics and Automation (ACRA)*, vol. 13, December 2009. 1, 2.1

[8] N. A. Cruz and A. C. Matos, “The MARES AUV, a modular autonomous robot for environment sampling,” in *OCEANS 2008*, pp. 1–6, IEEE, 2008. 1, 2.1

[9] J.-P. Ore, S. Elbaum, A. Burgin, B. Zhao, and C. Detweiler, “Towards autonomous aerial water sampling,” in *Robotic Science And Systems (RSS) - Workshop on Robotics for Environmental Monitoring, Berlin, Germany*, p. TBA, 2013. 1.2

[10] J.-P. Ore, S. Elbaum, A. Burgin, B. Zhao, and C. Detweiler, “Autonomous aerial water sampling,” in *Proc. of The 9th Intl. Conf. on Field and Service Robots (FSR). Brisbane, Australia*, vol. 5, p. TBA, 2013. 1.2, 2

[11] M. Dunbabin and L. Marques, “Robots for environmental monitoring: Significant advancements and applications,” *Robotics & Automation Magazine, IEEE*, vol. 19, no. 1, pp. 24–39, 2012. 2

[12] K. Anderson and K. J. Gaston, “Lightweight unmanned aerial vehicles will revolutionize spatial ecology,” *Frontiers in Ecology and the Environment*, vol. 11, no. 3, pp. 138–146, 2013. 2, 3.1

[13] A. Lucieer, D. Turner, D. H. King, and S. A. Robinson, “Using an unmanned aerial vehicle (UAV) to capture micro-topography of antarctic moss beds,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 27, pp. 53–62, 2014. 2

[14] P. P. Neumann, S. Asadi, A. J. Lilienthal, M. Bartholmai, and J. H. Schiller, "Autonomous gas-sensitive microdrone: wind vector estimation and gas distribution mapping," *Robotics & Automation Magazine, IEEE*, vol. 19, no. 1, pp. 50–61, 2012. 2

[15] A. Hodgson, N. Kelly, and D. Peel, "Unmanned aerial vehicles (UAVs) for surveying marine fauna: A Dugong case study," *PloS one*, vol. 8, no. 11, p. e79556, 2013. 2

[16] B. Zhang and G. Sukhatme, "Adaptive sampling for estimating a scalar field using a robotic boat and a sensor network," in *Robotics and Automation, 2007 IEEE International Conference on*, pp. 3673–3680, IEEE, 2007. 2.1

[17] G. S. Sukhatme, A. Dhariwal, B. Zhang, C. Oberg, B. Stauffer, and D. A. Caron, "Design and development of a wireless robotic networked aquatic microbial observing system," *Environmental Engineering Science*, vol. 24, no. 2, pp. 205–215, 2007. 2.1

[18] M. Dunbabin and A. Grinham, "Experimental evaluation of an autonomous surface vehicle for water quality and greenhouse gas emission monitoring," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 5268–5274, IEEE, 2010. 2.1

[19] J. Bellingham, C. Goudey, T. Consi, J. Bales, D. Atwood, J. Leonard, and C. Chryssostomidis, "A second generation survey AUV," in *Autonomous Underwater Vehicle Technology, 1994. AUV'94., Proceedings of the 1994 Symposium on*, pp. 148–155, IEEE, 1994. 2.1

[20] H. Singh, A. Can, R. Eustice, S. Lerner, N. McPhee, O. Pizarro, and C. Roman, "Seabed AUV offers new platform for high-resolution imaging," *Eos, Transactions American Geophysical Union*, vol. 85, no. 31, pp. 289–296, 2004. 2.1

[21] G. Dudek, P. Giguere, C. Prahacs, S. Saunderson, J. Sattar, L. A. Torres-Mendez, M. Jenkin, A. German, A. Hogue, A. Ripsman, *et al.*, "Aqua: An amphibious autonomous robot," *IEEE Computer*, vol. 40, no. 1, pp. 46–53, 2007. 2.1

[22] B. Allen, R. Stokey, T. Austin, N. Forrester, R. Goldsborough, M. Purcell, and C. von Alt, "Remus: a small, low cost AUV; system description, field trials and performance results," in *OCEANS'97. MTS/IEEE Conference Proceedings*, vol. 2, pp. 994–1000, IEEE, 1997. 2.1

[23] E. Fiorelli, N. E. Leonard, P. Bhatta, D. A. Paley, R. Bachmayer, and D. M. Fratantoni, "Multi-AUV control and adaptive sampling in Monterey Bay," *Oceanic Engineering, IEEE Journal of*, vol. 31, no. 4, pp. 935–948, 2006. 2.1

[24] M. Dunbabin, J. Roberts, K. Usher, G. Winstanley, and P. Corke, "A hybrid AUV design for shallow water reef navigation," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pp. 2105–2110, IEEE, 2005. 2.1

[25] R. Camilli, B. Bingham, M. Jakuba, H. Singh, and J. Whelan, "Integrating in-situ chemical sampling with auv control systems," in *OCEANS'04. MTTS/IEEE TECHNO-OCEAN'04*, vol. 1, pp. 101–109, IEEE, 2004. 2.1

[26] V. Hombal, A. Sanderson, and D. R. Blidberg, "Multiscale adaptive sampling in environmental robotics," in *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2010 IEEE Conference on*, pp. 80–87, IEEE, 2010. 2.1

[27] M. Walter, F. Hover, and J. Leonard, "SLAM for ship hull inspection using exactly sparse extended information filters," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pp. 1463–1470, IEEE, 2008. 2.1

[28] J. Melo and A. Matos, “Bottom estimation and following with the MARES AUV,” in *Oceans, 2012*, pp. 1–8, IEEE, 2012. 2.1

[29] J. Gould, D. Roemmich, S. Wijffels, H. Freeland, M. Ignaszewsky, X. Jianping, S. Pouliquen, Y. Desaubies, U. Send, K. Radhakrishnan, *et al.*, “Argo profiling floats bring new era of in situ ocean observations,” *Eos, Transactions American Geophysical Union*, vol. 85, no. 19, pp. 185–191, 2004. 2.1

[30] M. Rahimi, R. Pon, W. Kaiser, G. Sukhatme, D. Estrin, and M. Srivastava, “Adaptive sampling for environmental robotics,” in *Proc. IEEE Int. Conf. on Robotics and Automation*, vol. 4, pp. 3537–3544, 2004. 2.1

[31] “Jefferson project at Lake George - http://news.rpi.edu/content/2013/10/30/jefferson-project-lake-george-kicks-scientific-study-lake-high-tech-aerial-survey.” 2.1

[32] M. Hamilton, T. Dawson, and S. Thompson, “The Very Large Ecological Array,” in *AGU Fall Meeting Abstracts*, vol. 1, p. 08, 2011. 2.1

[33] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, and D. Estrin, “Habitat monitoring with sensor networks,” *Communications of the ACM*, vol. 47, no. 6, pp. 34–40, 2004. 2.1

[34] T. Koo, F. Hoffmann, H. Shim, B. Sinopoli, and S. Sastry, “Hybrid control of an autonomous helicopter,” in *Proceedings of IFAC Workshop on Motion Control*, pp. 285–290, 1998. 2.2

[35] T. Merz, P. Rudol, and M. Wzorek, “Control system framework for autonomous robots based on extended state machines,” in *Autonomic and Autonomous Systems, 2006. ICAS'06. 2006 International Conference on*, pp. 14–14, IEEE, 2006. 2.2

[36] D. Harel, “Statecharts: A visual formalism for complex systems,” *Science of computer programming*, vol. 8, no. 3, pp. 231–274, 1987. 2.2

[37] J. H. Gillula, H. Huang, M. P. Vitus, and C. J. Tomlin, “Design and analysis of hybrid systems, with applications to robotic aerial vehicles,” in *Robotics Research*, pp. 139–149, Springer, 2011. 2.2

[38] T. Merz and F. Kendoul, “Dependable low-altitude obstacle avoidance for robotic helicopters operating in rural areas,” *Journal of Field Robotics*, vol. 30, no. 3, pp. 439–471, 2013. 2.2

[39] S. Scherer, J. Rehder, S. Achar, H. Cover, A. Chambers, S. Nuske, and S. Singh, “River mapping from a flying robot: state estimation, river detection, and obstacle mapping,” *Autonomous Robots*, vol. 33, no. 1-2, pp. 189–214, 2012. 2.2, 2.4

[40] F. Amigoni and V. Caglioti, “An information-based exploration strategy for environment mapping with mobile robots,” *Robotics and Autonomous Systems*, vol. 58, no. 5, pp. 684–699, 2010. 2.3

[41] I. Vasilescu, C. Detweiler, and D. Rus, “AquaNodes: an underwater sensor network,” in *Proceedings of the second workshop on Underwater networks*, pp. 85–88, ACM, 2007. 2.3

[42] C. Stachniss and W. Burgard, “Exploring unknown environments with mobile robots using coverage maps,” in *IJCAI*, pp. 1127–1134, 2003. 2.3

[43] F. Kendoul, “Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems,” *Journal of Field Robotics*, vol. 29, no. 2, pp. 315–378, 2012. 2.4

[44] “Smartmicro.de - http://www.smartmicro.de/index.php/en/airborne-radar/micro-radar-altimeter.” 2.4

[45] S. Jain, S. Nuske, A. Chambers, L. Yoder, S. Scherer, and S. Singh, "Autonomous river exploration," in *Proc. of The 9th Intl. Conf. on Field and Service Robots (FSR). Brisbane, Australia*, vol. 5, p. TBA, 2013. 2.4

[46] S. Achar, B. Sankaran, S. Nuske, S. Scherer, and S. Singh, "Self-supervised segmentation of river scenes," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 6227–6232, IEEE, 2011. 2.4

[47] C. Poll and D. Cromack, "Dynamics of slung bodies using a single-point suspension system," *Journal of Aircraft*, vol. 10, no. 2, pp. 80–86, 1973. 2.5

[48] M. Bernard and K. Kondak, "Generic slung load transportation system using small size helicopters," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pp. 3258–3264, IEEE, 2009. 2.5

[49] P. R. Dahl, "Solid friction damping of mechanical vibrations," *AIAA Journal*, vol. 14, no. 12, pp. 1675–1682, 1976. 2.5

[50] K. Sreenath, N. Michael, and V. Kumar, "Trajectory generation and control of a quadrotor with a cable-suspended load a differentially-flat hybrid system," in *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 4888–4895, 2013. 2.5

[51] A. Faust, I. Palunko, P. Cruz, R. Fierro, and L. Tapia, "Learning swing-free trajectories for uavs with a suspended load," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 4902–4909, IEEE, 2013. 2.5, 4.3

[52] I. Palunko, R. Fierro, and P. Cruz, "Trajectory generation for swing-free maneuvers of a quadrotor with suspended payload: A dynamic programming approach," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 2691–2697, IEEE, 2012. 2.5

[53] G. Pisanich and S. Morris, “Fielding an amphibious uav: development, results, and lessons learned,” in *Digital Avionics Systems Conference, 2002. Proceedings. The 21st*, vol. 2, pp. 8C4–1, IEEE, 2002. 2.6

[54] G. Meadows, E. Atkins, P. Washabaugh, L. Meadows, L. Bernal, B. Gilchrist, D. Smith, H. VanSumeren, D. Macy, R. Eubank, *et al.*, “The flying fish persistent ocean surveillance platform,” in *AIAA Unmanned Unlimited Conference*, 2009. 2.6

[55] “Aquacopters - http://www.aquacopters.com.” 2.6

[56] “QuadH2O Amphibious UAV - http://www.rtfdrones.co.uk/product/rtf-quadh2o-waterproof-quadcopter/.” 2.6

[57] “rcgroups forum - http://www.rcgroups.com/forums/showthread.php?t=1753245.” 2.6

[58] G. Welch and G. Bishop, “An introduction to the Kalman filter,” 1995. 2.7, 5.1.2, 6.1.3

[59] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005. 2.7

[60] E. J. Lefferts, F. L. Markley, and M. D. Shuster, “Kalman filtering for spacecraft attitude estimation,” *Journal of Guidance, Control, and Dynamics*, vol. 5, no. 5, pp. 417–429, 1982. 2.7

[61] F. L. Markley, “Attitude error representations for Kalman filtering,” *Journal of guidance, control, and dynamics*, vol. 26, no. 2, pp. 311–317, 2003. 2.7

[62] M. Euston, P. Coote, R. Mahony, J. Kim, and T. Hamel, “A complementary filter for attitude estimation of a fixed-wing UAV,” in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pp. 340–345, IEEE, 2008. 2.7

[63] A. H. Göktoğan, S. Sukkarieh, M. Bryson, J. Randle, T. Lupton, and C. Hung, "A rotary-wing unmanned air vehicle for aquatic weed surveillance and management," *Journal of Intelligent and Robotic Systems*, vol. 57, pp. 467–484, 2010. (document), 2.1, 2.8

[64] "ITOPF - sampling and monitoring of marine oil spills - http://www.itopf.com." 3.1

[65] W. H. Organization, *Guidelines for drinking-water quality: recommendations*, vol. 1. World Health Organization, 2004. 3.1

[66] A. D. Eaton and M. A. H. Franson, *Standard methods for the examination of water & wastewater*. American Public Health Association, 2005. 3.1

[67] "Keeping Fremont lake 20 beaches open and toxin-free - http://water.epa.gov/lawsregs/lawsguidance/cwa/tmdl/nebraska.cfm." 3.1

[68] "201013 Haiti cholera outbreak - http://en.wikipedia.org/wiki/20103.3

[69] "Fukushima Daiichi nuclear disaster - http://en.wikipedia.org/wiki/fukushima_daiichi_nuclear_di
3.3

[70] "T-Hawk UAV enters Fukushima danger zone, returns with video. - http://www.engadget.com/2011/04/21/t-hawk-uav-enters-fukushima-danger-zone-returns-with-video/." 3.3

[71] "Reactor 3: Water leak most likely from inside pressure vessel. - http://fukushimaupdate.com/reactor-3-water-leak-most-likely-from-inside-pressure-vessel/." 3.3

[72] G. F. Ficetola, C. Miaud, F. Pompanon, and P. Taberlet, "Species detection using environmental DNA from water samples," *Biology Letters*, vol. 4, no. 4, pp. 423–425, 2008. 3.4

[73] J. C. Venter, K. Remington, J. F. Heidelberg, A. L. Halpern, D. Rusch, J. A. Eisen, D. Wu, I. Paulsen, K. E. Nelson, W. Nelson, *et al.*, "Environmental genome shotgun sequencing of the sargasso sea," *science*, vol. 304, no. 5667, pp. 66–74, 2004. 3.4

[74] C. L. Jerde, A. R. Mahon, W. L. Chadderton, and D. M. Lodge, "sight-unseen detection of rare aquatic species using environmental DNA," *Conservation Letters*, vol. 4, no. 2, pp. 150–157, 2011. 3.4

[75] P. THOMSEN, J. Kielgast, L. L. Iversen, C. Wiuf, M. Rasmussen, M. T. P. Gilbert, L. Orlando, and E. Willerslev, "Monitoring endangered freshwater biodiversity using environmental DNA," *Molecular Ecology*, vol. 21, no. 11, pp. 2565–2573, 2012. 3.4

[76] "Ascending Technologies - http://www.asctec.de." 4.2

[77] "SolidWorks - http://www.solidworks.com." 4.3.1

[78] "MakerBot - http://www.makerbot.com." 4.3.1

[79] "McMaster-Carr 20 ml glass vial screw-top plastic lidded model 4417T48 http://www.mcmaster.com/#catalog/120/1700/=rmtdni." 4.3.2

[80] "HiTEC - http://hitecrcd.com." 4.3.3

[81] "TCS Micropumps, UK. Model M200S-SUB - http://micropumps.co.uk/TCSM200range.htm." 4.3.5

[82] "Maxbotix - http://www.maxbotix.com." 4.4.1

[83] "Robot Operating Systemi - http://www.ros.org." 5

[84] "seamos - https://code.google.com/p/seamos." 5.2

[85] "scons - http://www.scons.org/." 5.2

[86] "VICON http://www.vicon.com." 6.1.2

[87] W. H. Orem, *Impacts of sulfate contamination on the Florida Everglades ecosystem*. Fact Sheet FS 109-03. Reston, Virginia, U.S. Geological Survey, 2004. 7.5.2

[88] W. K. Dodds, *Freshwater ecology: concepts and environmental applications*. Academic Press: San Diego, California, 2002. 7.5.2