System Design of an Unmanned Aerial Vehicle (UAV) for Marine Environmental Sensing

by

Joshua Leighton

S.B., Massachusetts Institute of Technology (2010)

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2013

Author . . . .

Department of Mechanical Engineering
November 26, 2012

Certified by . . . .

Franz S. Hover
Finmeccanica Career Development Professor of Engineering, MIT
Thesis Supervisor

Certified by . . . .

Nicholas M. Patrikalakis
Kawasaki Professor of Engineering, MIT
Thesis Supervisor

Accepted by . . . .

David E. Hardt
Chairman, Department Committee on Graduate Students

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE
2013

2. REPORT TYPE

3. DATES COVERED
00-00-2013 to 00-00-2013

4. TITLE AND SUBTITLE
System Design of an Unmanned Aerial Vehicle (UAV) for Marine Environmental Sensing

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Massachusetts Institute of Technology,Department of Mechanical Engineering,Cambridge,MA,02139

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Technological improvements over the past decade have led to the widespread use of autonomous surface and underwater vehicles for data collection in marine environmental sensing and modeling in coastal environments. However, these vehicles and their sensors still have limitations, especially when tasked with observing highly dynamic or transient processes. We investigate the application of a small unmanned aerial vehicle (UAV) to the study of two such phenomena: Harmful Algal Blooms (HABs) and thermal plumes. A complete field-operable system was developed to identify and characterize HAB events through a human-monitored supervisory control system. This capability was extended with an infrared imaging camera for remote sensing of thermal plumes, enabling future work to augment the in-situ measurements of surface craft with thermal imagery from a UAV. Experiments in Singapore have led to the successful identification and subsequent study of algal blooms on multiple occasions and demonstrated the potential for observation and modeling of thermal plumes.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**70**	

System Design of an Unmanned Aerial Vehicle (UAV) for Marine Environmental Sensing

by

Joshua Leighton

Submitted to the Department of Mechanical Engineering
on November 16, 2012, in partial fulfillment of the
requirements for the degree of
Master of Science in Mechanical Engineering

Abstract

Technological improvements over the past decade have led to the widespread use of autonomous surface and underwater vehicles for data collection in marine environmental sensing and modeling in coastal environments. However, these vehicles and their sensors still have limitations, especially when tasked with observing highly dynamic or transient processes. We investigate the application of a small unmanned aerial vehicle (UAV) to the study of two such phenomena: Harmful Algal Blooms (HABs) and thermal plumes. A complete field-operable system was developed to identify and characterize HAB events through a human-monitored supervisory control system. This capability was extended with an infrared imaging camera for remote sensing of thermal plumes, enabling future work to augment the in-situ measurements of surface craft with thermal imagery from a UAV. Experiments in Singapore have led to the successful identification and subsequent study of algal blooms on multiple occasions and demonstrated the potential for observation and modeling of thermal plumes.

Thesis Supervisor: Franz S. Hover
Finmeccanica Career Development Professor of Engineering, MIT

Thesis Supervisor: Nicholas M. Patrikalakis
Kawasaki Professor of Engineering, MIT

Acknowledgements

I would like to first thank my thesis advisors, Prof. Franz Hover and Prof. Nick Patrikalakis. It has been a privilege to work with Prof. Hover the past four years and I look forward to continuing my work with him. Prof. Patrikalakis has allowed me many opportunities through the Singapore MIT Alliance for Research and Technology and has always ensured a practical focus on experimentation.

Thanks to my labmates Mei Yi Cheung, Audren Cloitre, Brendan Englot, Eric Gilbertson, Matt Gildner, Rob Hummel, Paul-Arthur Jeantaud, Georgios Papadopoulos, Brooks Reed, Lynn Sarcione, Josh Taylor, Tom Trapp, and Changkuan Yu. Special thanks to Tawfiq Taher in Singapore for his work coordinating sea trials, Jacob Izraelevitz for his help interfacing the thermal camera, Mike Benjamin for his always quick responses helping with MOOS, and Maria Riefstahl for putting up with my many parts orders and complex travel arrangements.

Work was supported by the Singapore National Research Foundation through the Singapore MIT Alliance for Research and Technology and the Office of Naval Research, Grant N00014-06-10043.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Marine Environmental Sensing on a Global Scale

Understanding our oceans and coastal environments is a concern to countries throughout the world for multiple reasons. The oceans play a significant role in both local and global climates and modeling climate change depends on modeling the oceans themselves. Many countries, including some in southeast Asia, depend on the tourism their coastal environments bring them. Yet despite the importance of these marine environments our understanding of them remains limited because studying them can be so difficult. The vast scale of oceans and coastal waters is well suited to autonomous sensing methods and autonomous vehicles have seen wide adoption in that marine community over the past decade.

1.1.1 Autonomous Platforms

There are five commonly used platforms for marine environmental sensing: fixed sesnors, drifters, gliders, autonomous underwater vehicles (AUVs), and autonomous surface vehicles (ASVs). These platforms can be equipped with a wide array of sensors and varying levels of autonomy. Environmental sensing networks will often employ a combination of several different platforms to achieve the best results.

Fixed platforms: Floating buoys anchored in place or seafloor-based packages can provide continuous observations in key locations. In light of recent disasters, pressure monitoring nodes mounted to the seafloor now provide advanced tsunami warning in countries including Malaysia, India, and Greece. Surface buoys can be equipped with numerous environmental sensors to monitor conditions both above and below the surface. Winch-mounted sensor packages can even provide complete water column coverage. (Fugro OCEANOR, 2012) The Monterey Bay Aquarium Research Institute (MBARI) maintains several moorings in the vicinity of Monterey Bay that provide real time observations via a two-way radio link. The moorings are used to inform environmental models, test new sensors, and plan future experiments. (MBARI, 2004)

Drifters: Ocean drifters are simple platforms deployed in large numbers to cover a wide area while they move with the ocean currents. A single drifter consists of a surface buoy attached to a subsurface drogue, or sea anchor. The buoy communicates any sensor measurements using a satellite transmitter, which also allows the position of the buoy to be tracked over time. The drogue is designed to dominate the hydrodynamic drag of the drifter and ensure it follows the upper ocean currents. While a single drifter‘s capabilities are limited, their simplicity and long life (over a year) mean many can be deployed together and provide useful data for climate modeling.

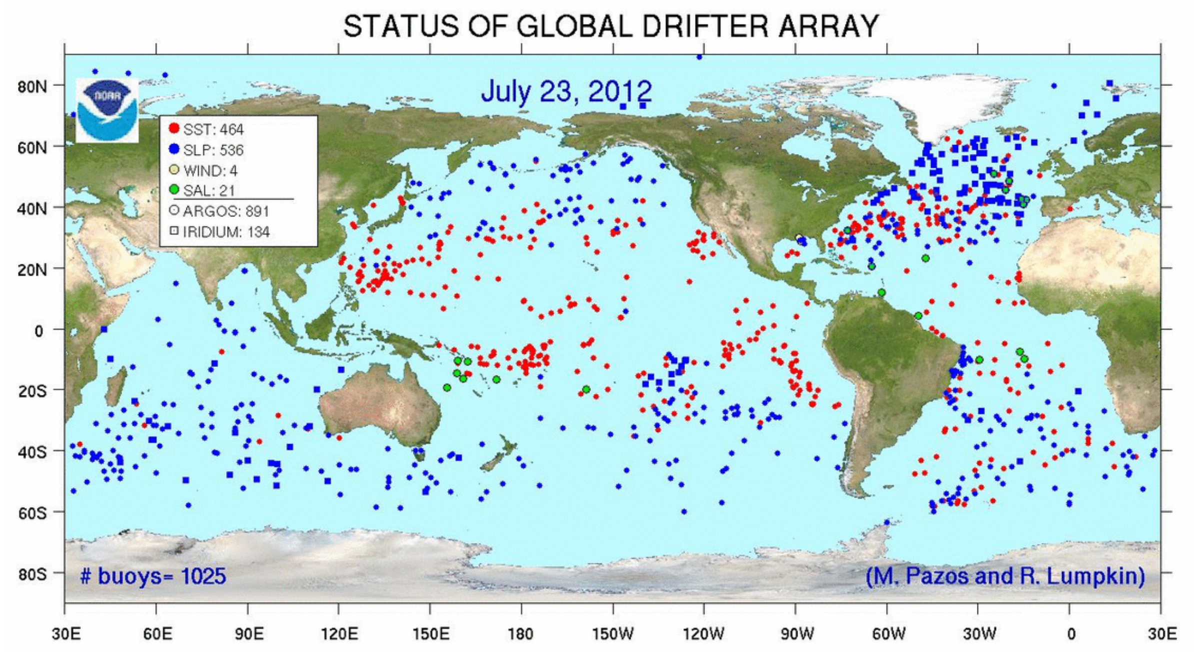

Figure 1-1: Distribution of drifters in the Global Drifter Array. Different colors represent drifters with different sensor capabilities. (image source: Atlantic Oceanographic and Meteorological Laboratory)

The Global Drifter Program (GDP) is a component of NOAA's Global Ocean Observing System consisting of over 1000 drifters. The network of drifters became fully operational in 2005 following the deployment of 1250 drifters and has been in continuous operation ever since. Drifters are continuously replaced or strategically deployed in advance of large weather events such as hurricanes. Originally designed to measure sea surface temperature while following ocean currents, recent drifters can also measure atmospheric pressure, winds, and salinity. Data is available to scientists to support both short-term climate predictions and long-term climate research. (AOML, 2012)

Gliders: Gliders strike a balance between drifters and AUVs in terms of endurance, mobility, and computing power. The vehicle buoyancy is controlled to be slightly positive or negative through the use of a buoyancy engine, usually consisting of an oil filled bladder or piston evacuated cylinder. Wings provide horizontal lift while the vehicle descends and ascends through the water column. Additional control, such as steering and pitch, can be accomplished using external control surfaces or by positioning internal ballast, often the batteries.

The resulting track follows a zigzag pattern, as in Figure 1-2, with periodic surfacing to obtain an updated GPS fix and communicate with operators. Propulsion power is only required at the peaks of the zigzag when changing the vehicle's buoyancy. With a sufficient depth range to operate over a glider can descend or ascend for hours at a time with almost zero power consumption, though speeds are limited. The Bluefin Spray Glider has a maximum speed of just 35 cm/sec, but can operate for up to six months at a time, allowing it to cover thousands of kilometers. (WHOI, 2012) Newer and larger gliders however, such as the Liberdade class being developed through the Office of Naval Research (ONR) can achieve speeds of several knots with their wing-body design. (ONR)

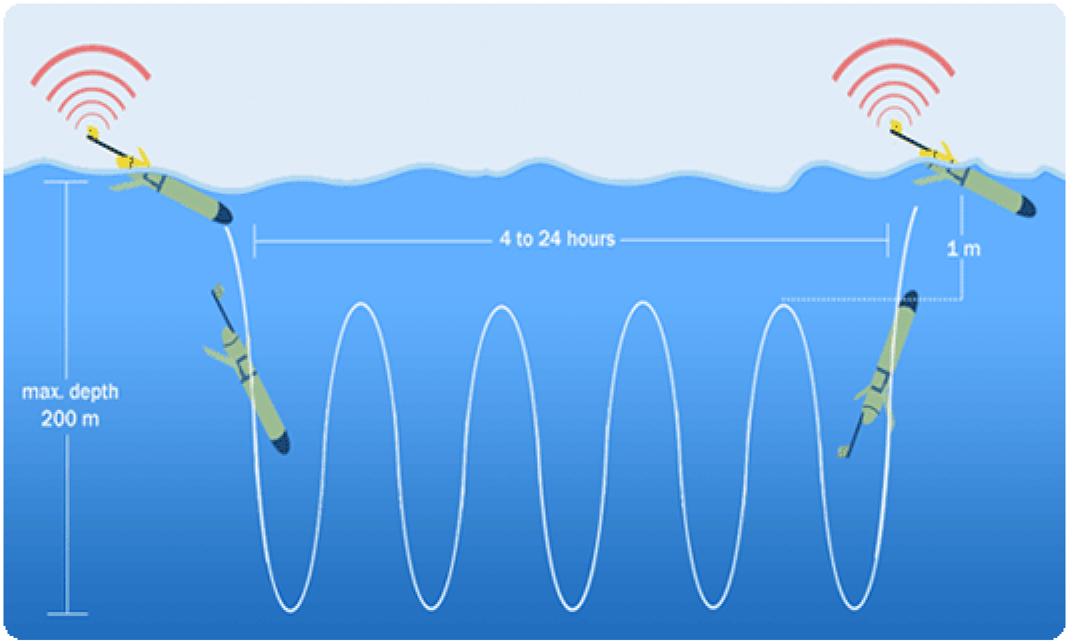

Figure 1-2: Diagram detailing the operation of a Slocum Electric Glider. The vehicle may perform multiple ascents and descents collecting data before surfacing for communication and localization. (image source: International Coalition of Ocean Observing Laboratories)

Regardless of design, gliders depend on vertical space to operate efficiently and cannot contend with strong currents, making them better suited for deployment in deep ocean waters as opposed to shallower coastal regions. Onboard computation power is often limited to simple navigation and sensor logging to maintain minimal power consumption. With only small forward thrust generated by the wings, sensor options are also limited and cannot significantly affect the vehicle's hydrodynamics. Despite these limitations though, gliders can be an effective and relatively inexpensive platform for sensing over long distances. Gliders were deployed extensively following the 2010 Deepwater Horizon accident and collected valuable fluorometric data used to determine oil concentrations throughout the Gulf of Mexico. (Shepard, 2011)

Autonomous Underwater Vehicles (AUV): AUVs come in multiple shapes and sizes to fulfill a wide variety of roles, from oceanographic data collection to ship hull inspection. The torpedo shaped AUV is the standard design for oceanographic work where straight line performance is prioritized over maneuverability. Individual manufacturers employ different designs, such as free-flooded vs. fully sealed hulls and vectored thrust vs. control fins, but the simple cylindrical shape with stern mounted propeller remains the same. Ongoing miniaturization of sensors and computers allows these AUVs to be equipped for missions including undersea archaeology and exploration, environmental monitoring, and acoustic surveillance.

Figure 1-3: An OceanServer Iver2 operating on the surface.

At one end of the spectrum lie small AUVs such as the one man portable OceanServer Iver2 , a six inch diameter torpedo shaped AUV. It can operate for several hours in shallow (<100m) coastal waters while equipped with standard sensors such as a CTD or ADCP. (OceanServer, 2012) At the other extreme are deep-water AUVs such as the Bluefin 21. Capable of operating several thousand meters deep for a day or more, these AUVs can carry advanced bottom scanning sonars or tow multi-transducer arrays. (Bluefin Robitics, 2012)

Autonomous Surface Vehicles (ASV): While not as well suited to offshore deployments where surface conditions can be too rough, autonomous surface vehicles are ideal for operations in harbors, lakes, and shallow coastal zones. They can provide payload and computation capabilities similar to larger AUVs while being much cheaper and easier to operate. Capable of speeds of several meters per second, they can operate in much stronger currents than many AUVs. Their ease of use combined with the availability of RF communications also makes them an ideal platform for the development of control and autonomy software.

Figure 1-4: A SCOUT equipped with a pan/tilt mounted ADCP being deployed from MV Mata Ikan on the Johor Strait, Singapore.

The SCOUT (Surface Craft for Oceanographic and Undersea Testing) is an ASV built around a standard kayak hull. Initially developed at MIT in 2004, the SCOUT was designed as a platform for testing AUV algorithms that could be deployed independently or in conjunction with AUVs. (Curcio J., 2005) It has since been commercialized and has been used in a wide variety of field trials, from oceanographic studies to experimental hydrodynamics research. (Zheng, 2009) Equipped with a winch, the vehicle can deploy oceanographic sensors at various depths, providing an AUV's data product for shallow waters without the usual hassle and expense. (McGillivary, et al., 2006)

1.1.2 Sensors

The sensors described here are just a few of the most common sensors typically found on autonomous marine platforms.

Conductivity, Temperature, and Depth (CTD) Sensors: Possibly the most common instrument used in oceanographic experiments, a CTD can measure water temperature, salinity, and density. Modern CTDs are small enough to be mounted on almost any vehicle while still providing excellent accuracy. A CTD is usually the first step in studying any environmental phenomenon. Additional sensing capabilities for parameters such as pH and dissolved oxygen can also be built into CTD instruments.

Mass spectrometers and flourometers: At their most fundamental level, mass spectrometry is used to measure the mass-to-charge ratio of charged particles and determine elemental composition or the chemical structure of molecules. Underwater mass spectrometers provide this functionality in a small package that operates on samples of water, usually provided by an internal pump and filter apparatus.

They are useful for identifying low molecular weight hydrocarbons and volatile organic compounds associated with pollutants. Flourometers perform similar measurements by examining the intensity and wavelength of electromagnetic emissions from a sample after excitation. They identify higher molecular weight compounds including organic entities such as chlorophyll. Together the two techniques can give us a more precise picture of the various chemicals present in a water sample.

Water sampling: Even with the in-situ measurement capabilities of today's sensors, it is still valuable to collect water samples for analysis under the microscope in a lab. Traditionally water samples are collected by humans working on a ship using Niskin bottles, a device designed to collect a water sample from a certain depth. However this capability is making its way onto AUVs with devices such as the Gulper, developed at MBARI. The Gulper water sampling system consists of ten syringe-like samplers mounted in a cylindrical AUV section. Software onboard the vehicle can trigger sampling based on readings from other onboard sensors, such as a CTD. At the end of a mission the samples are collected and analyzed onboard the research vessel or preserved for study onshore. (MBARI, 2010)

Figure 1-5: The GULPER water sampling system shown mounted in the midsection of a 21" diameter AUV. (photo credit: MBARI)

Acoustic Doppler Current Profiler (ADCP): An ADCP can perform remote current measurements by sending acoustic pulses and measuring the Doppler shift in the return reflected by particles in the water. Through the use of multiple beams, an ADCP can determine the three-dimensional current flow at multiple distances from the sensor. Range can vary from just a few meters to hundreds of meters, with accuracy typically decreasing with range as lower frequencies must be used. ADCPs are frequently mounted facing downwards from an AUV or surface vehicle to measure current throughout the water column. Some units can also function as a Doppler Velocimetry Logger (DVL), locking onto the seafloor and measuring the velocity of the vehicle rather than water currents.

Imaging sonar: While less applicable to environmental sensing, imaging sonars still bear mentioning here because of their widespread use on underwater vehicles. Side-scan sonars are commonly used on AUVs or towed behind surface ships to image a wide area of seafloor to either side of the vehicle. Varying frequency, usually in the several hundred kilohertz range, provides a tradeoff between swath width and resolution. Other types of sonar are designed to more accurately image smaller areas or even characterize the sediment or rock under the seafloor.

1.2 Specific Issues in Coastal Environments

In this thesis we will focus on the vehicles and methods used to study two marine sensing problems present in Singapore, harmful algal blooms (HABs) and industry-related thermal plumes. Neither of these phenomena is unique to Singapore and the methods explored here for observing them are widely applicable. Both contain robotics and sensing problems that prove challenging for traditional marine environmental sensing platforms. A closer look at the specifics of these two phenomena, including their causes and observable parameters, will reveal the need for innovative sensing techniques.

The work discussed here was performed as part of the Center for Environmental Sensing and Modeling (CENSAM), a department of the Singapore MIT Alliance for Research and Technology (SMART). CENSAM's goal is to combine environmental sensing and modeling to demonstrate the importance of pervasive sensing in real world applications. To study algal blooms, thermal plumes, and fulfill other marine sensing missions CENSAM has developed a tiered sensing network primarily composed of autonomous surface and underwater vehicles. These vehicles can be equipped with sensors suitable to the task at hand, such as CTDs and mass spectrometers for algal blooms or temperature probes and ADCPs for thermal plumes. Other ongoing work is focused on more tightly integrating the sensing and modeling aspects of these problems to build a truly adaptive sensing network. As we will see however, algal blooms and thermal plumes can still present significant challenges for this sensing network. (censam)

1.2.1 Harmful Algal Blooms

Harmful algal blooms have many impacts, cause significant damage, and are also difficult to observe and study. An algal bloom is simply a rapid population increase of algae in a marine environment, but the causes and effects vary greatly. A harmful algal bloom is one that has negative effects, which may occur through oxygen depletion, the release of toxins, or other means depending on the species of algae involved. Various algae species also react differently to external conditions, so knowing what types of algae are involved in a bloom is essential to predicting and reacting to bloom events.

In the US a number of algae species are associated with harmful algal blooms in different parts of the country, such as *Karonia brevis* in the Gulf of Mexico or *Alexandrium fundyense* along the New England coast. These blooms are frequently known as "red tides" for the reddish brown discoloration that commonly occurs in the water. These algal species release toxins that, during a bloom event, can reach high enough levels to kill fish and other marine organisms in huge number. Human health can also be impacted through the consumption of shellfish or other seafood contaminated with bloom-related toxins. Shoreline quality is also deteriorated by large algal blooms, affecting the local tourism economy. In 1999 the National Oceanographic and Atmospheric Administration estimated that HABs had caused over $1 billion in damages in the United States. (Bushaw-Newton, 1999), (Center for Disease Control, 2012), (Massachusetts Department of Publc Health, 2012)

Figure 1-6: A harmful algal blooom of the species *Lingulodinium polyedrum* near San Diego, California. (photo source: NOAA National Ocean Service)

Since the algal species responsible for red tides have been identified and their effects are well documented, it is possible to monitor bloom pre-conditions, predict their occurrence, and respond accordingly. In Hong Kong the Red Tide Information Network combines periodic water sampling, food sampling and reports from local fisheries to monitor the likelihood of a harmful algal bloom event. In response to reports of elevated algal populations or red tide sightings, the government can take action to determine if a bloom is in fact hazardous and if so, respond accordingly. Plans are in place for protecting fish farm populations through early harvesting or relocation and monitoring the safety of public beaches and seafood sources. (Hong Kong Agriculture, Fisheries and Conservation Department, 2012)

In situations like Hong Kong's, knowledge of the nature of local HABs makes monitoring, prediction, and reaction possible, but in other locations our understanding is limited. With countless species of algae worldwide, scientists still don't know what effects different species have on humans or marine organisms, much less understand what causes most algal blooms. Singapore is one such location where recent events have prompted closer investigation in HABs. In January of 2010 more than 200,000 fish were killed at fish farms near Pasir Ris and Pulau Ubin in Singapore, the biggest reported loss in 10 years with damages exceeding $2 million. (Quek. C., 2010) The deaths were attributed to a plankton bloom that decreased oxygen levels in the water, suffocating the fish. Without identifying the species responsible though, it's impossible to know what environmental factors contributed to the bloom or how future blooms could be predicted.

CENSAM's aim in the study of HABs in Singapore is to better inform decision makers so that future harmful algal blooms can be predicted or even prevented, and damages minimized. Oceanographic models need to be combined with measured data to better understand how local algal blooms develop. One immediate goal is to associate algal blooms in Singapore with easily measured environmental variables such as temperature, salinity, and dissolved oxygen. ASVs using CTDs and mass spectrometers can measure these key parameters and associate them with the chemical compounds present in the water. Collecting data in the vicinity of algal bloom events in conjunction with lab analysis of water samples can tell us what harmful algae species we need to be concerned about and what pre-conditions can lead to their blooming.

Despite these sensing capabilities, data collection from algal bloom events is predicated on finding algal blooms when they do occur. Since so little is currently known about blooms in Singapore, it is difficult to predict when they will occur in the future. Furthermore algal blooms are often very localized and transient, making them that much harder to find and study. (Richardson, 1996) Time scales can be on the order of hours to days and while some blooms might cover hundreds of kilometers, others can be just a few hundred meters across. Blooms will also move with water currents, making continued observation difficult while physically separating the bloom from its original location and the factors that contributed to its formation there. ASVs can only gather measurements at their current location and their slow speed combined with strong currents around Singapore limit their ability to efficiently search for algal blooms. A more effective solution for finding algal blooms and guiding the deployment of ASVs is needed, especially in these early stages of HAB study when prediction is difficult.

1.2.2 Thermal Plumes

Heavy industry plants through the world use local water sources extensively for cooling, in many cases without fully understanding the potential ramifications. Many plants were constructed before the introduction of environmental regulations controlling acceptable temperature rise near hot water exhausts. In other cases the thermal efficiency of a plant is affected by interaction between cold water inflows and hot water exhausts or the operation of multiple plants in close proximity.

Singapore, an industrial hub with limited land and sea area available, faces several such issues with both old and new industrial plants. In the Johor Strait off the north coast of Singapore, the Senoko power plant's intake and exhaust flows operate in close proximity to each other, leading to potential interference depending on tidal currents. Off the southwest coast of Singapore in the Tuas area, numerous refining plants operate in close proximity causing an overall rise in local water temperatures. Existing plants seek tools that can help them redesign their outflows to both improve plant efficiency and bring them closer to compliance with new environmental regulations. New plants being constructed similarly want to maximize their efficiency while maintaining compliance when many plants are clustered together.

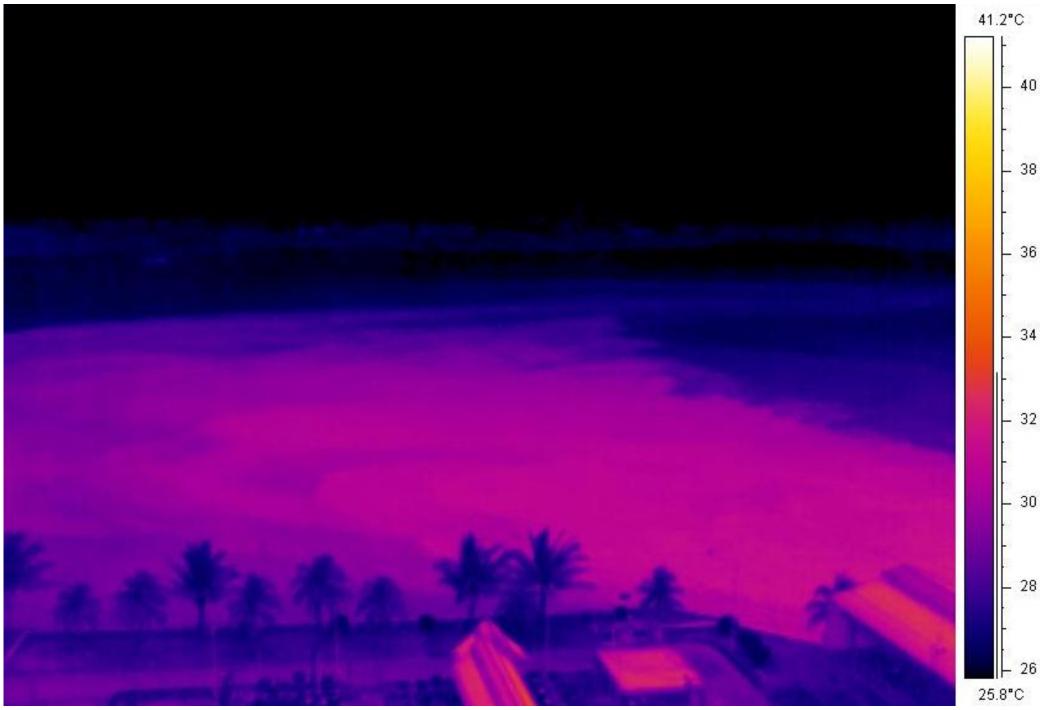

Figure 1-7: A thermal image of the Senoko power station cooling exhaust. While a camera could be mounted on a high smokestack here, the low angle of incidence affects the accuracy of temperature readings. In other locations vantage points such as this one are simply not available. (image source: Tawfiq Taher)

CENSAM is looking to develop the tools these plants need by modeling the flow of hot water away from industrial outflows and verifying the accuracy of these models on existing installations. High volume outflows can be very dynamic, with local features occurring over scales of seconds to minutes in additional to tidal variation over the course of several hours. While the location of the exhaust jets is precisely known, the powerful currents produced by the jets make it difficult to operate marine vehicles. In many cases the velocity of the jet exceeds the maximum speed of the ASVs, making measurement in some areas impossible. Local dynamics within the jet also cannot be observed by an ASV taking single point measurements. A more global measurement of the plume, such as the thermal image in Figure 1-7, can be combined with the data provided by ASVs to more accurately model the plume.

1.3 Motivation for an Aerial Vehicle

The study of both HABs and thermal plumes presents challenges that a sensing network consisting of AUVs and ASVs is not well equipped to handle. These phenomena have variations occurring over time scales much shorter than we are typically used to seeing in the ocean. In addition algal blooms can cover large areas and are difficult to predict and find, while thermal plumes are associated with strong flows that limit where an AUV or ASV can successfully operate. Surface and underwater vehicles are primarily limited in these situations by their slow speed and localized in-situ measurements.

1.3.1 Mission Definition

An unmanned aerial vehicle (UAV) can provide many of the capabilities that marine vehicles lack, namely speed and broader field measurements. When working in conjunction with surface and underwater vehicles, a UAV can improve the efficiency of the entire sensing network. To implement an effective aerial solution we first look at how a UAV would ideally complement the study of algal blooms and thermal plumes. Table 1-1 lists some of the key mission parameters that would apply to an aerial vehicle.

Table 1-1: Mission parameters for study of algal blooms and thermal plumes by a UAV.

Parameter	Algal blooms	Thermal plumes
Knowledge goal	• Determine the existence and location of potential algal blooms • Direct the deployment of AUVs and ASVs for in-situ measurements • Secondary data product	• Set of measurements that can be used to reconstruct the full field and be validated or calibrated using AUVs and/or ASVs • Primary data product
Target characteristics	• Bloom's location, size, and temporal aspects all unknown • Best identified by some color variation	• Plume's approximate location and size known in advance
Measurement characteristic	• Visual spectrum • Looking for meta-scale properties such as color contrast	• Infrared spectrum • Accurate measurement of individual pixel values
Operator interaction	• Define search area • Providing visual feedback via identification • Behavior modification after discovery	• Define fairly well known coverage area • Set and forget
Coverage and navigation requirements	• Rapid coverage of large areas • Low spatial accuracy required	• Smaller area • High spatial accuracy required
Data processing	• Mosaicking based on navigation data • Limited image processing for partial bloom recognition	• Mosaicking and field reconstruction based on navigation data and calibration measurements • Potential for next-best view or similar

Knowledge goal: The first and most important property that needs to be defined for each mission type is the goal of an aerial vehicle in the mission itself, or what it is we wish to learn by deploying a UAV. In the study of algal blooms the biggest challenge a surface or underwater vehicle faces is finding a bloom at all. In this case an aerial vehicle can cover large areas quickly to identify candidate algal blooms before directing AUVs and ASVs to the target area. The final data product will be the measurements taken by the in-situ sensors on those vehicles, whereas the information supplied by the UAV is only in support of that. On the other hand, an aerial vehicle can provide a primary data product when studying thermal plumes through the use of a thermal camera. Here the goal is a complete reconstruction of surface temperature near a hot-water exhaust that can be validated by surface vehicles and contribute to the development of a complete model.

Target features: Knowing the goal of the UAV in each mission type, we need to identify the key target features that will define how the vehicle can perform observations. The location and size of any possible algal blooms is largely unknown. Time scales are expected to be on the order of several hours, but the bloom may not be visually identifiable throughout its existence. Blooms in the Johor Strat of Singapore are expected to appear as darker brown or green patches at the surface of the water [Tkalich, personal communication], though water and bloom colors vary depending on weather, algal species, and even viewing angle. On the other hand, the location and size of thermal plumes is approximately known and the disturbance caused by an exhaust jet is visible on the surface.

Measurement characteristic: Given both the mission goal and some basic target features we can define the important measurement characteristics. For finding algal blooms a UAV needs to perform measurements in the visual spectrum that can either be processed by a computer or viewed by an operator to identify abnormal coloring in the water. Precise observations aren't important because ASVs or AUVs can be deployed for verification of potential bloom sightings. For studying thermal plumes and modeling a hot water exhaust jet, precise temperature readings are required. In-situ measurements can provide accurate reference temperature measurements but will be limited in spatial and temporal coverage.

Operator interaction: As with any system that will need to be operated in the field, especially in a marine environment, it's important to consider the level at which the operator will interact with the system. When searching for potential algal blooms we can expect an operator to take an active role even after defining some initial search area. Part of the identification process will probably depend on a human's interpretation. In the event that a potential bloom is spotted, the operator may want to interrupt the original search mission to maintain stationary observation or explore the boundaries of the bloom. The thermal plumes mission is much less subjective and the result is a more clearly defined data product, the complete surface temperature field. Here we can expect more of a set-and-forget approach where the operator defines an area and lets the mission run its course.

Coverage and navigation method: Both missions call for coverage of an area but with different focuses. Finding algal blooms requires that we cover large areas quickly though not necessarily with much accuracy. The target may not be well differentiated from the background water and measurements are somewhat subjective. Building a model of a thermal plume requires that not only the measurement itself be accurate, but also the position information associated with that measurement. The area that needs to be covered is smaller and the goal is not to perform an exhaustive search but to ultimately reconstruct a field, so observations in some areas might be more important.

Data processing and autonomy: These two missions also call for some level of automated data processing, whether it be on the vehicle or on an operator's computer. Some geo-referenced mosaicking is certainly desirable for both mission types, though the type of data varies. Algal bloom identification could be aided by image processing, but the varying appearance of different blooms could make this is a difficult task for a computer. Concrete temperature measurements are much more amenable to automated processing though and more advanced techniques such as next-best view could be implemented.

The algal blooms mission requires a more flexible approach in which the vehicle and its operator must react to any potential bloom sightings and choose a best course of action. Actions could include a new outward spiraling search pattern to find the edges of a bloom, station keeping to maintain observation, or coordinating with a second quadrotor to provide uninterrupted observation. The thermal plume mission has a narrower scope and is unlikely to change once the plume dimensions are defined. The measurements are more objective in nature though and a higher degree of confidence is needed so that the data may be directly used in modeling the thermal plume. Algal bloom sightings serve primarily to direct surface and underwater vehicles that are responsible for producing the primary data product that will be later used to inform models.

1.3.2 Existing Work in Small Aerial Vehicles

The increased availability of small, low-cost autonomous and remote aerial vehicles has seen a surge in their use for various research and remote sensing applications. More advanced control methods for careful trajectory following and maneuvering have been explored under controlled environments where precision position sensing is available via a video tracking system or similar (Mellinger, 2012). Simultaneous localization and mapping (SLAM) has been very successfully implemented in indoor environments using laser scanners and cameras (Achtelik, 2009) (Shen, 2011). Outdoor environments provide additional challenges including wind disturbances and larger scale environments that may challenge SLAM algorithms, though recent work has focused on improving the autonomous operation of quadrotors in feature-rich outdoor environments (Wendel, 2011). Remotely-operated aerial vehicles with little or no built-in autonomy are commercially available through several manufacturers and used not only for research, but also aerial reconnaissance by law enforcement agencies, rescue services, and others.

The application of an autonomous aerial vehicle to marine environmental sensing presents challenges most similar to those tackled by remotely operated commercially operated vehicles. The environment is very feature-poor, and hence not amenable to the large body of research available for navigation and control of quadrotors in a more controlled environment. The desired operation ranges also greatly exceed those typically supported by commercial systems, which typically rely on the vehicle remaining within eyesight of an operator. The unique marine environment exposes the vehicle to harsher external conditions, primarily wind, and imposes a very high penalty, complete loss, for any system failure, whether it be in software or hardware.

2 System Design

Based on our identification of mission composition and objectives for algal blooms and thermal plumes we can set a number of design requirements for an aerial vehicle system and explore various options for meeting these requirements. Design aspects range include the sensors we need to carry, the hardware of the vehicle itself, the software to control the vehicle, and the interface with the operator of the system and other vehicles in the sensing network. Thermal plumes were not being considered when the vehicle was first designed for the study of algal blooms, but we will examine the design of the vehicle with respect to both mission types concurrently and note where early design decisions for algal blooms affected operations for thermal plumes.

The scope of this thesis primarily covers the development of a system consisting of just a single vehicle operating in conjunction with an operator and his or her computer. However with the often short flight times of aerial vehicles, considerations were made for the potential operation of multiple vehicles in the future. More than one vehicle could work simultaneously for quicker coverage or in tandem to maintain a continuous presence depending on the application need.

2.1 Sensors

In both missions the goal is to support the study of an environmental phenomenon. As such, the sensors we need to observe that phenomenon will define the overall design and operation of our system. The sensor and ultimate data product also clearly differentiate the two mission types from one another.

2.1.1 Algal Blooms

As discussed during in the mission definition, algal blooms in Singapore are expected to be visible on the surface of the water as a darker brown or green patch. A bloom may not be visible on the surface for its entire lifespan, but we expect that it should be visible when the algae population is near its peak. While in-situ sensors such as CTDs carried by ASVs may detect a bloom over a greater fraction of its lifespan, water discoloration is a feature we can easily and continuous look for from the air.

While the discoloration of a bloom is visible to the naked eye, it is very difficult to spot from a ship because the water is viewed at such a shallow angle. As shown in Figure 2-1, the reflectivity of water increases sharply when the angle of incidence exceeds fifty degrees. High reflectivity will conceal any discoloration associated with algal blooms lying just under the surface, especially when working against strong sunlight reflections. With a higher vantage point, such as that provided by an aerial vehicle, a much larger area can be seen at once while still viewing at a small angle of incidence.

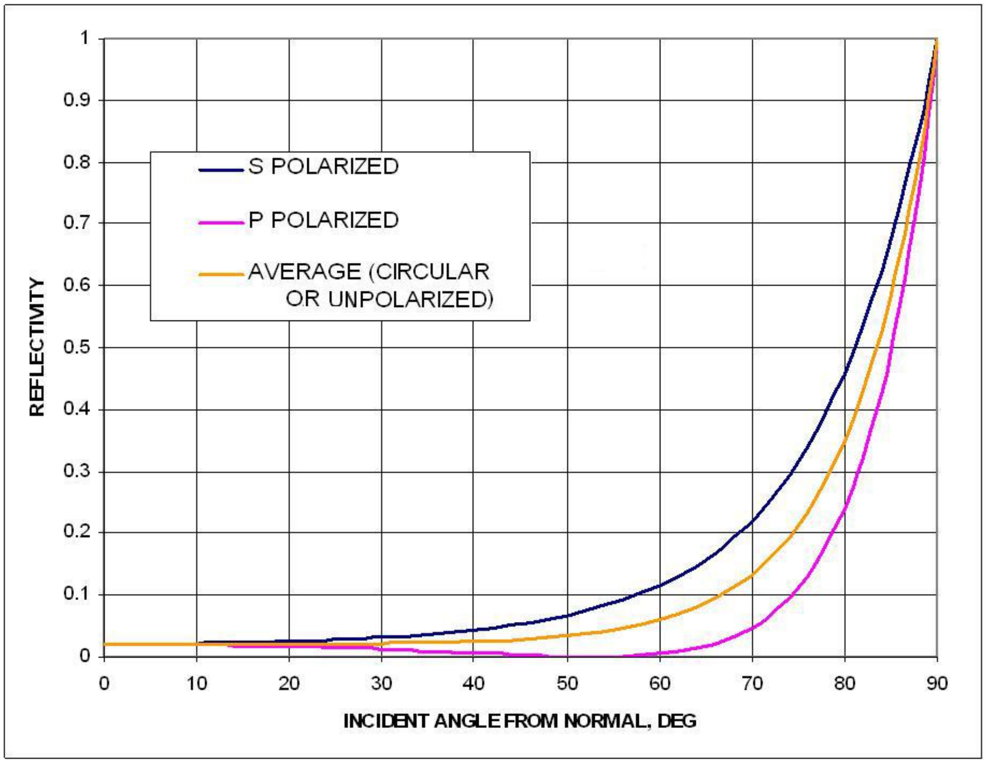

Figure 2-1: The reflectivity of water varies with the angle at which the light is reflected. Observations taken from a surface ship might be in the 70- to 80-degree range, while an aerial vehicle can view from directly above, or near zero degrees. (image source: wikimedia.org)

This relationship between reflectivity and angle of incidence also affects how a camera is mounted on the vehicle and the type of lens used. Using a wider angle lens or mounting the camera at a shallower angle increases the field of view when operating at a given altitude, but increases the angle of incidence near the edges of image. The effects of varying lens angles and mounting arrangements will become clear in later experimental results. In most applications lens distortion is a concern when using wider angle lenses where a “fish eye” effect becomes very prominent near the edges of the image. However, simply finding a bloom is more important than any accurate reconstruction or mosaicking. Increasing altitude may also seem to be a simple solution for expanding the field of view, but permitting in Singapore restricts operations of our aerial vehicles to an operating ceiling of just twenty meters.

Weight is of course a critical concern on any aerial vehicle, especially a battery powered one. Fortunately there are many small and lightweight video and still cameras available that provide acceptable image quality. Since the discoloration associated with an algal bloom should occur over a large area, resolution and general image quality can be sacrificed in favor of weight savings. When interacting with the system, an operator would like to view imagery in real time so ASVs can quickly be deployed to verify potential sightings. As will later be shown, image processing for bloom identification can be difficult, so this human-in-the-loop interaction is important to successfully detect the discoloration associated with blooms. It was also found that real-time imagery cold be very useful in determining the position of the vehicle relative to features onshore when the vehicle was operating beyond visible range.

These criteria lead us to compact video cameras that can wirelessly transmit data in real time. A variety of cameras and transmitter/receiver combinations are possible and readily available for hobby use to provide “first person video”, whereby the pilot of a hobby aircraft can fly his or her aircraft using onboard video. Cameras such as the popular KX-171 provide resolution on the order of 560x420 in a tiny package (30 grams) and provide decent imagery even in low-light conditions. Analog video transmitters provide standard definition video quality over ranges of up to several kilometers if paired with a high gain receiving antenna. 5.8 Ghz transmitters were used to comply with Singapore frequency regulations and avoid interference with other wireless links operating in the 2.4 Ghz spectrum (such as 802.11x wifi). The power draw of even higher-powered transmitters is still small compared to power required to keep an aerial vehicle aloft, and the 600 mW ImmersionRC (ImmersionRC Ltd) transmitter was found to work at ranges up to 1.5 km using only standard ‘rubber-duck’ antennas. On the receiving end, analog video can be viewed by an operator and imported into a computer using off-the-shelf video hardware. 5.8 GHz transmitters also have eight different channels available, allowing multiple vehicles to transmit video simultaneously without interfering with each other.

While analog video transmission is simple to implement, it can frequently yield video with static or other artifacts. At longer ranges transmission quality can vary significantly with a vehicle’s orientation depending on how the transmitter antenna is mounted. An operator can easily ignore static artifacts and focus on the good frames, but an inconsistent video stream does introduce additional challenges if performing remote video processing. Conversely, video processing before transmission requires considerable processing power onboard the vehicle, adding weight. For the scope of this thesis, real time video processing is not fully handled and the issue of onboard vs. shipboard processing is left for future consideration.

2.1.2 Thermal Plumes

To look at the infrared spectrum and study thermal plumes we need a very different sensor. Remote temperature measurement can be performed with relatively inexpensive spot meters, which measure the temperature at a single spot on a surface, or with thermal cameras, which can provide an instantaneous picture of the temperature over a wide area. A thermal camera is preferable in this application to gather significantly more data in each observation and to enable observation of the rapid dynamics present in a jet. Thermal cameras are typically too bulky for a small aerial vehicle though and smaller thermal cameras come with tradeoffs in accuracy and precision.

At the time of this system’s development the FLIR Tau 320 uncooled Long Wave Infrared (LWIR) thermal imaging core was the only sub-500 gram thermal camera that provided a sufficient set of features for collecting data that could be used for modeling thermal plumes. The Tau series of cameras weighs as little as 72 grams when configured with a wide angle 9-mm lens, making it small enough to be carried by many commercial battery-powered aerial vehicles. With a sensitivity of 50 mK, it can easily resolve the temperature gradients expected to be found near industrial thermal plumes, where exhaust flows can easily be 5 degrees Celsius or more above the ambient water temperature. (FLIR, 2012)

Figure 2-2: Frame capture from the Tau 320's analog video feed showing a boat and the warm water in its wake as it passes underneath the quadrotor.

The compact size of the Tau 320 comes with several shortcomings that affect how the UAV would be operated when studying thermal plumes. Larger thermal cameras will often use an imaging core cooled to a specific temperature that serves as a reference for the bolometer sensor. Since the Tau is uncooled, it lacks this absolute reference point and its measurements are expected to vary as the temperature of the camera itself varies. The error in its temperature measurements may drift over the course of a single mission as the camera itself is exposed to different temperatures, due to wind, sunlight, and heating through its own power dissipation. Temperature readings within a given image are accurate relative to one another and a periodic Flat Field Correction (FFC) is performed to maintain this relative accuracy.

Interface options are also limited on the Tau 320. Its temperature measurements have 14 bits of resolution, but only eight bits of output per pixel is possible using the standard analog video output. FLIR implements a proprietary algorithm to convert 14-bit temperature values into 8-bit grayscale or color-mapped video output. This mapping of values is constantly changing as the range of temperature in the camera's field of view varies and the camera's absolute calibration fluctuates. Video output from the camera could be wirelessly transmitted using the same hardware used for transmission of the standard camera's video feed, but this would introduce further errors into the image. Reading the precise temperature values is important if we're to build a model with this thermal data, so the analog video output cannot be used. 14-bit video data is available through a camera-link compatible output, but the hardware required to capture and save this data is far too bulky to be carried onboard a smaller aerial vehicle.

Figure 2-3 shows the importance of accessing the full 14-bit temperature values from the thermal camera. The two images were generated by mapping the same 14-bit snapshot to 8-bit grayscale values using different linear functions. The left image maps the entire range of temperature values, from the cooler water to hot parts of the ship. In the right image, only a subset of the full temperature range is covered by the mapping. The ship now appears as a solid block of high temperature, but variations in water temperature can be seen that were not previously visible, such as warmer water near the boat. Similar methods are required to study thermal plumes, especially if ships or other objects may be present in the image to affect the range of temperatures seen by the camera.

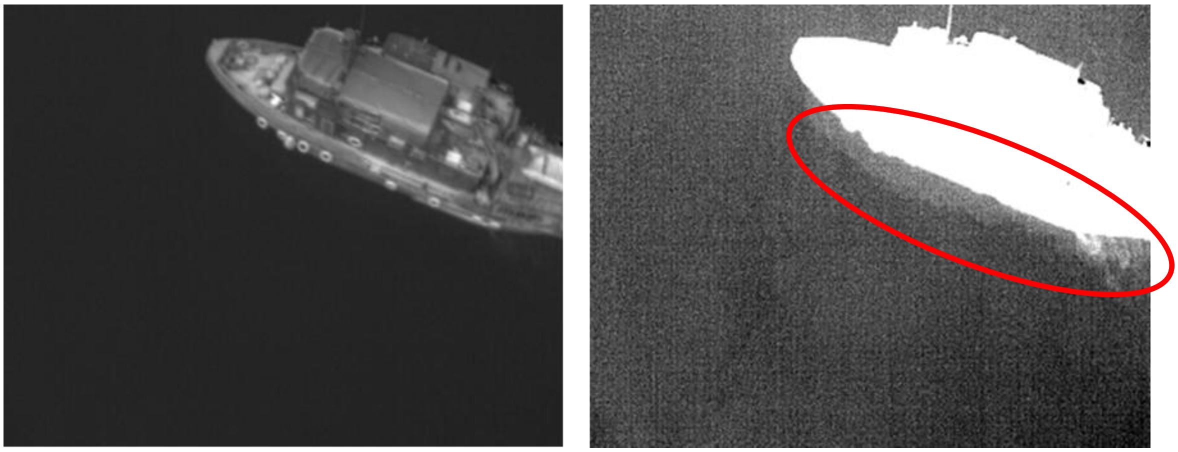

Figure 2-3: Two different 14-bit to 8-bit conversion functions applied to the same thermal snapshot. Note the warm water (circled) near the boat in the right image not visible in the left image.

Without a high speed USB or Ethernet connection available, we are forced to use the slower serial connection for downloading snapshots from the camera. Primarily designed for camera control and debugging, the serial interface can also be used to store up to three snapshots at a time in the Tau 320's internal memory. These snapshots can then be downloaded over the serial connection and erased, allowing for another three snapshots to be taken. At maximum baud rate it takes approximately 20 seconds to download and erase three snapshots. Of course some on-board computing and logging capability is required to handle communication over the serial interface.

2.1.3 Summary

The important sensor characteristics for algal bloom and thermal plume missions are summarized in Table 2-1. While both are performing field measurements with an imaging sensor, the hardware itself and the important characteristic of that measurement vary significantly. A human operator can identify an algal bloom and the sensor is designed for that type of use, whereas the temperature measurements needed for studying thermal plumes are much better handled by a computer.

Table 2-1: Summary of key sensor characteristics.

Attribute	Algal bloom	Thermal plume
Sensor	Video camera	Thermal camera (FLIR Tau 320)
Data handling	Real-time transmission	On-board logging
Supporting hardware	Analog video transmitter	(System on Chip) SoC or microcontroller
Measurement characteristic	Color variation or boundary	Temperature field with internal relative accuracy

2.2 Vehicle Hardware

The system was initially developed around the algal blooms mission, so we will first look at how that mission type influenced the design and selection of an aerial platform before discussing how it was also suited for thermal plume missions.

2.2.1 Selecting a Platform for Algal Bloom Missions

UAVs fall into two major categories: fixed-wing and helicopters. Blimp-based designs are also possible, but ill-suited for outdoor operation where winds may be present. Within the fixed wing and helicopter classes there a variety of designs and sizes available, both battery- and gas-powered. As with full-size aircraft, fixed wing vehicles typically have better endurance and faster speed than their helicopter counterparts, though they also require more provisioning for launching and landing. Many hobby-sized fixed wing aircraft can be launched overhand and even caught by a skilled operator. Helicopters have the bonuses of hovering capabilities and truly vertical takeoffs and landings. To avoid the hassle of handling fuel and potential noise and permitting issues, only battery-powered vehicles were considered.

When studying algal blooms, the expected role of an aerial vehicle is to perform initial reconnaissance, quickly searching a large area for potential sightings. Precision positioning for any given measurement is not important. Once spotted, it’s also desirable to stay on station and monitor the bloom for position changes while surface vehicles are deployed to the location. Ideally the vehicle could operate at a high altitude of fifty meters or more to achieve a wide field of view and easier coverage of the search area. In this operating case a fixed wing aircraft would be best because it can travel faster and stay aloft longer than a helicopter. Once deployed, a plane could operate for an hour or more while traveling over 15 m/s, exploring a large area or maintaining continuous coverage over a smaller one. While hovering is not possible, loitering at higher altitudes or using an actuated camera mount would still allow the vehicle to keep a single target within its field of view. The biggest challenge would be launch and recovery of the vehicle from ships of opportunity or smaller support craft.

Unfortunately regulations in Singapore concerning the operation of hobby class aerial vehicles impose a number of limitations on our design, the most restricting being a maximum altitude of just twenty meters. Operating at twenty meters limits the field of view of our sensor and could make it difficult for an operator to use if working with a fixed wing vehicle travelling at high speed. A helicopter could move slower, allowing an operator more time to interpret the images being sent back, while also allowing the vehicle to hold position at any time and maintain a static field of view. Also, Singapore regulatory bodies were more receptive to the operation of helicopters that could be accurately and slowly controlled as opposed to a plane with potentially much greater range. Operational areas are usually less than 2 square kilometers in size as well, so the reduced range is not too severe a limitation. For these reasons, a helicopter platform was chosen over a fixed wing plane.

Most hobby helicopters follow a traditional design with one main rotor and one tail rotor, but recent multi-rotor designs, such as quadrotors, are rapidly gaining popularity in both hobby and research circles. Quadrotors use four lifting rotors, two rotating clockwise and two rotating counterclockwise, mounted at the ends of a cross-shaped structure as shown in Figure 2-4. By varying the thrust of different rotors the vehicle can be tilted in any direction to provide horizontal motion, or turned in the yaw direction. Quadrotors are mechanically simple compared to traditional helicopter designs; brushless motors can be connected directly to propellers without requiring any of the complex linkages associated with a normal helicopter's main rotor. These features translate to a more reliable vehicle and relatively straightforward autonomous control, making the craft especially useful in research or other applications were full or partial computer control is useful.

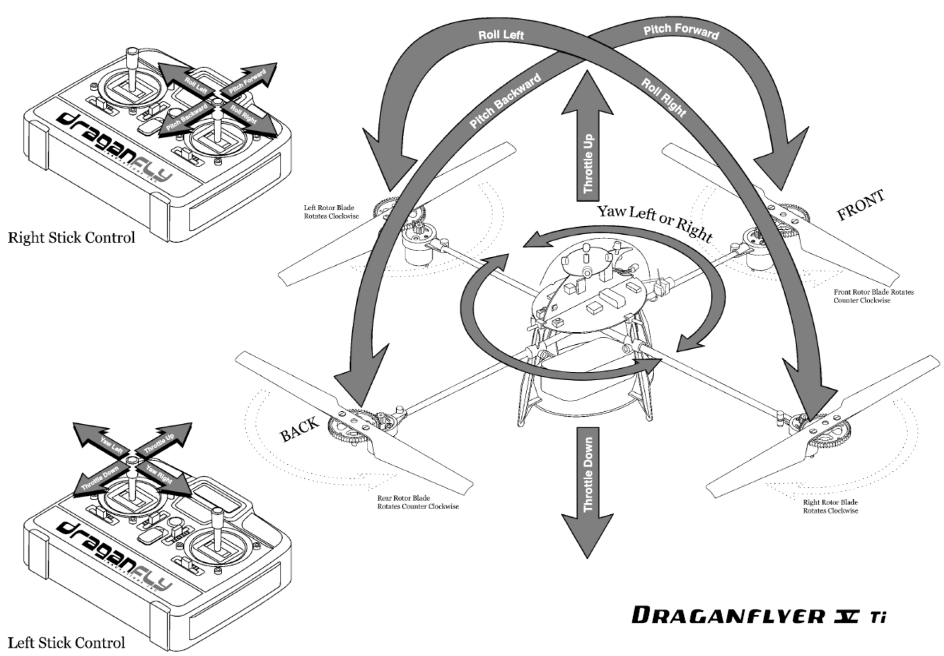

Figure 2-4: RC control diagram of a typical quadrotor. Yaw motion is accomplished by varying the speed of clockwise and counterclockwise rotating propellers, applying a net torque about the vertical axis. (image source: Draganfly Innovations Inc)

The Pelican quadrotor produced by Ascending Technologies in Germany was chosen for its research-oriented design and good payload capacity and flight time. Past experience with Ascending Technologies' vehicles also motivated the Pelican's use from an ease-of-development perspective. While small, measuring just 50 cm across and weighing approximately 1.1 kg including a battery, the Pelican can lift up to 500 g of payload. Flight times are advertised around twenty minutes, but with a higher capacity 8Ah 11.1V Lithium Polymer battery flights of up to 27 minutes have been achieved when carrying a lightweight video payload appropriate for algal bloom missions. Maximum velocity is about 13 m/s, which also means the vehicle can operate in winds of almost twenty knots. The platform includes a built-in flight controller with GPS, compass, and pressure sensors for navigation, and accelerometers and gyros for attitude control. (Ascending Technologies GmbH)

Figure 2-5: The Ascending Technologies Pelican in its standard configuration. The orange tape is used to denote the front on the vehicle. (photo source: Ascending Technologies GmBH)

Control of the Pelican is accomplished either manually through a handheld RC controller or via serial communications with the onboard autopilot. Using either interface several control modes are possible, as summarized in Table 2-2.

Table 2-2: Ascending Technologies Pelican flight modes.

Control Mode	Interface	Description
ACC	RC	Joysticks control thrust, yaw rate, and pitch and roll of the vehicle. Releasing the pitch/roll joystick causes the vehicle to return to level flight.
ACC+height	RC	Like ACC mode, but using the pressure sensor for altitude control. Centering the thrust joystick causes the vehicle to maintain constant altitude.
GPS	RC	GPS and compass are used to control horizontal velocity. Height and yaw control function the same as in ACC+height mode, but the pitch/roll joystick now controls velocity in the vehicle's frame of reference. Releasing this joystick causes the vehicle to hold position.
Serial-ACC	Serial	Thrust, yaw rate, pitch, and roll commands are transmitted via the serial interface. This is essentially a computer controlled version of the ACC mode.
Waypoint	Serial	Individual waypoints are transmitted to the autopilot, which controls the vehicle to its target location.

Regardless of operating mode, flight data such as location and orientation can always be requested via the serial interface. Ascending Technologies also provides development tools to run custom code on the included autopilot, though this capability has not been needed in our work.

2.2.2 Applicability to Thermal Plumes

While the Pelican quadrotor was originally selected based on the requirements of an algal bloom mission, it can also fulfill the needs of a thermal plume mission. The modeling of thermal plumes requires field temperature measurements that are not only accurate in terms of temperature, but also spatially so. The limitations of the Tau 320's data interface means that the vehicle must allocate individual snapshots carefully because only a set number of images can be taken and downloaded in a single mission. Therefore it's important that the vehicle is accurately positioned for each snapshot to avoid wasting valuable flight time. Unlike a fixed wing vehicle, the quadrotor can easily and stably hover in place to take individual snapshots.

The Pelican's 500 gram payload capacity also allows it to carry the approximately 70 gram thermal camera without significantly decreasing flight time. Field experiments have shown flight times around 17 minutes are possible when carrying the thermal camera and any additional hardware associated with its mounting and communication.

2.2.3 Vehicle Summary

The Ascending Technologies Pelican provides a capable aerial platform fully assembled and ready-to-fly. Using a commercially available system such as this one allows us to leverage years of quadrotor hardware and control research and development. Table 2-3 summarizes the key attributes of the Pelican quadrotor as they apply to the algal bloom and thermal plume missions.

Table 2-3: Summary of the Ascending Technologies Pelican's capabilities.

Attribute	**Algal Blooms**	**Thermal Plumes**
Vehicle type	Ascending Technologies Pelican quadrotor	
Max. Payload	500 g	
Max. Velocity	13 m/s	
Max. Wind speed	10 m/s	
Flight time	Approx. 25 min.	Approx. 17 min.
Max. Range	Approx. 15 Km	Approx. 10 Km
Sensors	GPS, compass, pressure (altitude), 3-axis accelerometer and gyroscope	
Control Interface	Various from full manual to waypoint – RC and computer interface	

2.3 Computing and Communications

Onboard computing, communication, or a combination of the two is required to control the Pelican quadrotor through its autopilot's serial link. In its standard configuration this serial link is tied directly to a Digi XBee wireless modem, allowing a paired XBee modem connected to a computer to communicate directly with the onboard autopilot. Ascending Technologies provides PC software for reading data from the vehicle and sending waypoints to the autopilot. The serial interface with the autopilot operates at 57600 baud, so a 2.4 GHz XBee that can match that communication rate is usually used for communication. However the maximum range of 2.4 GHz XBees is advertised as 1.6km line-of-sight and is typically much less in practice as many wireless devices, including 802.11x WiFi, operate in the 2.4 GHz spectrum. In operations near ASVs using powerful WiFi transmitters, 2.4 GHz XBee range was as short as 100m, hardly sufficient for operating an aerial vehicle. A more comprehensive solution is needed for reliable field operation on the vehicle.

2.3.1 Communication Options

Regardless of what additional onboard computing capability is added, some form of communication with the vehicle will always be required. Again Singapore regulations limit the available design options. In the US, 900 MHz radios are commonly used when looking for a longer range but lower bandwidth solution. Singapore frequency spectrum allocations do not allow for unlicensed use of the 900MHz band, but the XBee 868 MHz modems are an option. These devices have exceptional range (up to 40km line of sight) and reliability, but very low bandwidth averaging less than 9600 b/s. Like the 2.4 Ghz XBees, communication is done via a serial interface and the devices can be configured to function as a transparent serial link, simplifying setup and software development. (Digi International Inc.)

A second communication option is standard 802.11x WiFi, which is already used by ASVs in Singapore. Range can reach up to a kilometer but typically requires powerful, and hence sizable, amplifiers and high gain antennas. This would provide a high bandwidth link to the quadrotor, more than adequate for control, but not as reliable as the lower frequency XBees. Furthermore, 802.11x is not easily compatible with the serial interface used by the quadrotor's autopilot.

These limitations lead us to consider adding additional onboard computing power to the vehicle in the form of a small microcontroller or system on chip (SoC). An onboard computer could handle control of the vehicle by communicating with the autopilot through its serial interface while also forwarding data and commands to or from operators on a ship. Since only a subset of the data exchanged with the autopilot needs to be exchanged with the ship as well, a lower bandwidth wireless modem can be used. The 868MHz XBees give us a reliable data link with the vehicle over its operating range (as determined by battery life and speed) and has enough bandwidth to receive waypoint commands and send periodic navigation updates. The onboard computer can also provide additional safety by maintaining control of the vehicle in case of a communications failure.

Using XBees also yields benefits when extending functionality to multiple vehicles. With just two vehicles, a pair of XBees can be configured as either a transparent serial link or to simply broadcast to all modules in range. With three or more modules in use, mesh configurations can be used to automatically handle addressing and routing of messages when communicating with the modules using their API. Unlike traditional WiFi devices, the XBees can handle substantial changes in the network configuration without loss of connectivity and without additional complication to the software interfacing with the module. Acknowledgements can also be performed at the hardware level, enabling the transmitting XBee to report back to software whether or not a given message was successfully received by the specified device. These features make it easy to transition from a two-node system to a multi-node network with minimal additional software development.

2.3.2 Onboard Computing

The onboard computing equipment needs to provide at least two serial interfaces, one for an XBee and one for the autopilot, while also handling basic navigation, decision making, and data logging. A microcontroller, such as an Arduino, is simple to use but also imposes a number of programming limitations and excludes the use of any of the currently available robotics middleware software packages. On the other hand, modern SoCs can provide considerable computing power in extremely small packages and allow different operating systems and great software flexibility. Future expansion of the vehicle with different sensors or payloads is also easier with an SoC, as will be seen with integration of the thermal camera.

The Gumstix computer-on-modules (COMs) were selected as one of the few available products (at the time) in such a small form factor with readily available support for developers. Gumstix uses a system of expansion boards that mate with the computer mainboard to minimize overall size and weight while providing a diverse set of IO options. The Verdex series was initially selected due to its robotics-oriented expansion boards, but now an Overo COM is preferred for its more powerful processor and better support network. The Summit expansion board is used because it provides all the necessary serial and power connections through its 40-pin header, while USB host, console and On-The-Go (OTG) ports allow for connection to the Tau 320's USB port and USB networking for data retrieval.

Overo COMs come preloaded with Gumstix's version of the Linux kernel and the OpenEmbedded operating system, a Linux distribution focused on maintaining a small footprint suitable for embedded operating systems. The Bitbake cross-compilation system was used to produce a slightly modified bootloader and kernel for supporting USB networking and the Tau 320's USB to serial converter as well as rerouting a third serial port's signals to the 40-pin header on the summit expansion board. Bitbake was previously used to also build a stripped-down version of the OpenEmbedded OS without graphical support, but Multistrap is now used to create a Debian/Emdebian image that is compact while allowing Debian packages to be used, simplifying software development. Kernel and bootloader builds are also being transitioned away from the BitBake build system. The complete combination of bootloader, kernel, and OS image is loaded onto a microSD card from which the Overo can boot.

2.3.3 Summary

The Gumstix Overo using Debian allows us to take advantage of the large community developing for Debian while providing ample computing power in a tiny package, smaller than many Arduino boards. Onboard processing reduces the quantity and rate at which data needs to be exchanged with an operator's computer, so the 868MHz modules can be used to ensure reliable connection to the vehicle in all conditions. A minimal amount of support components are needed in addition to the Gumstix and XBee. 3V and 5V regulators power the XBee and Gumstix, while a pair of level shifters convert between the 1.8V logic of the Overo, 3V logic of the XBee, and 5V logic of the quadrotor's autopilot.

Table 2-4: Key attributes of computation and communication hardware added to the vehicle.

Attribute	**Algal Blooms**	**Thermal Plumes**
Onboard computer	Gumstix Overo Computer on Module (COM) with 800 MHz 800 MHz ARM processor	
Power consumption	< 1 Watt	
Communications	868 MHz Digi XBee	
Data rate	24 Kbps burst, 2.4 Kbps average	
Range	40 Km line of sight	

2.4 Integration and Software

So far we have defined the requirements for both algal bloom and thermal plume missions and designed the hardware for an aerial vehicle that can fulfill those needs. The resulting platform combines a small quadrotor, including built-in autopilot for flight control, with additional onboard computation in the form of a Gumstix Overo embedded linux computer. Communication is handled using 868 MHz XBee modules while analog video data is transmitted separately on a 5.8 GHz link. A small onboard video camera provides the necessary imagery data in real time via the video link and Flir's Tau 320 thermal camera can be added for thermal plume missions.

Development of a complete software infrastructure to both control the vehicle and integrate the various hardware pieces is the last step remaining in forming a complete system, but also the most complicated. Ideally the software should handle all of the design requirements initially outlined, including interfacing with the user and handling navigation, while also being flexible for future development. Since this aerial vehicle system will be operating in conjunction with surface and underwater vehicles, closer integration with those systems might be desired after proving the capabilities of an individual aerial vehicle. Deployment of multiple aerial vehicles is another potential direction for future development and imposes additional requirements, especially where the operator interacts with the system.

Software was first developed around the algal blooms missions and went through several significant design iterations that affected the overall architecture, vehicle control, and user interface. The control and interfacing requirements for thermal camera and thermal plumes were tackled after developing most of the software around the algal blooms mission type. We will split our analysis into four major sections: early approaches to system architecture, methods for vehicle control, implementation of the robotics middleware package that is used today, and extensions for thermal plumes missions. Ongoing and planned future work will also be discusses as it applies to the various components.

2.4.1 Early System Architecture

Early implementations of the control software did not perform any onboard computation and instead focused on basic flight control. Communication with the vehicle was accomplished using the standard 2.4 GHz XBees for better bandwidth at the expense of range. Control software ran on an operator's computer and transmitted control commands over the wireless link while continuously receiving new data from the vehicle. This architecture eased early development by eliminating the need to recompile code on the vehicle, usually a slow process due to hardware limitations. Various control experiments using the lower-level pitch, roll, thrust, and yaw interface were conducted using this setup. The motivation behind these experiments and their results are discussed in depth later.

Past experience and the need to do low-level serial interfacing led to the use of C++ as the primary programming language. The wxWidgets library was used to more easily build a user interface resembling the one in Figure 2-6. This form-based interface allows an operator to easily view data from the vehicle, such as altitude and position, while sending commands in absolute or relative terms. For example, a user can command an absolute altitude of ten meters and a relative position change of fifteen meters north and twenty meters east while maintaining the current heading, allowing for easy testing of various step responses when developing a controller. Data is also logged locally on the operator's computer for post-mission analysis.

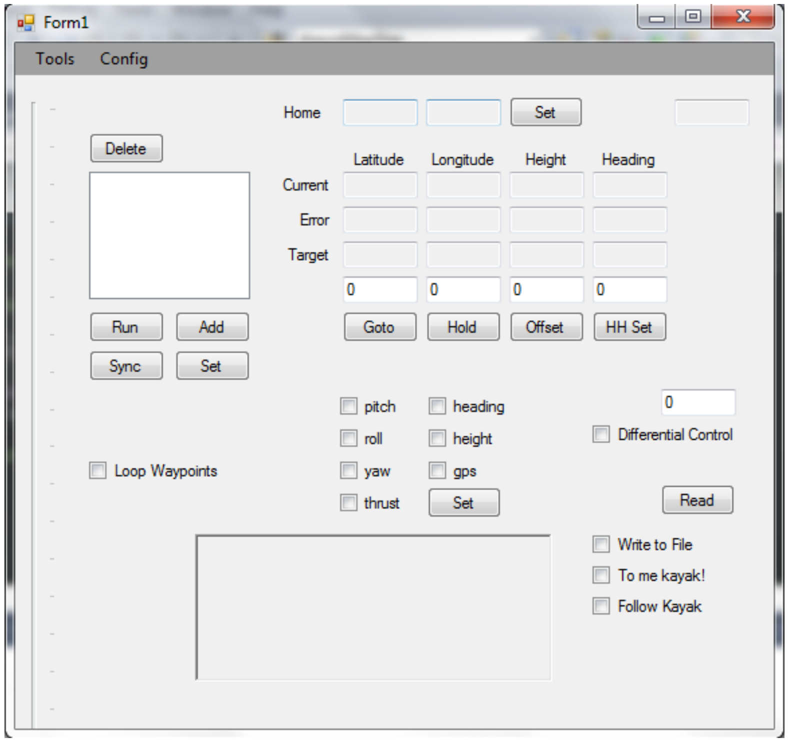

Figure 2-6: Early wxWidgets based interface with various fields for monitoring vehicle data and sending commands.

The move to onboard computing using the Gumstix necessitated substantial changes to the overall software architecture. Many of the functions previously performed on the operator's computer were moved to the Gumstix, such as vehicle interfacing, real time control, and data logging. With the lower bandwidth 868 MHz XBees, a new protocol had to be developed for communication between the Gumstix on the vehicle and the operator's laptop. The user interface needed to provide the same control and data viewing capabilities as previously, but also utilize the new communication protocol to pass information, such as waypoint lists, to the vehicle. This functionality of this split architecture is outlined in Figure 2-7. Additional sensors could also be added using the Gumstix's interfacing capabilities, and an ultrasonic rangefinder used in some control experiments is shown here as well.

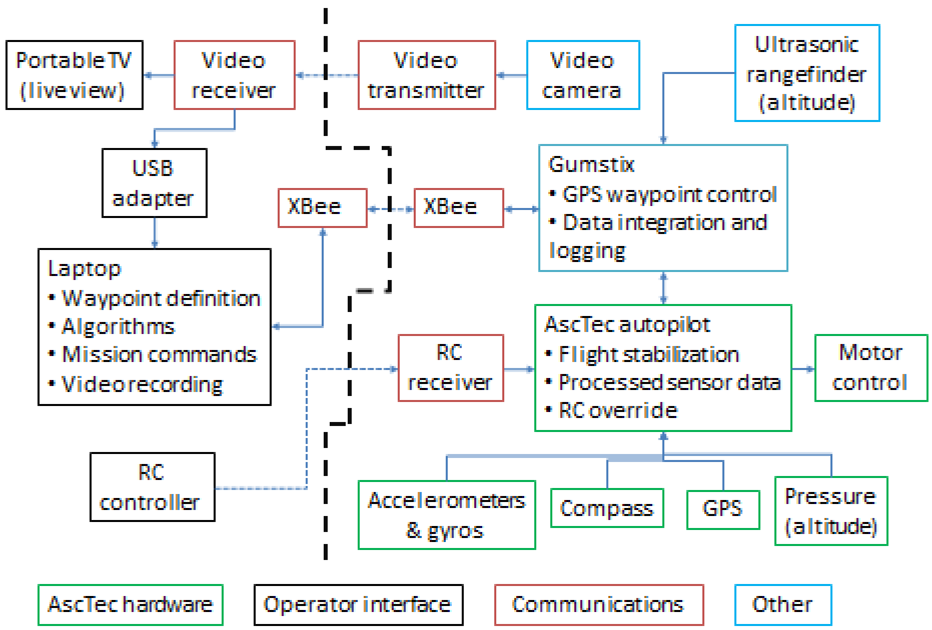

Figure 2-7: Functional diagram of the main system components. The dashed line separates hardware on the quadrotor (right) from hardware used at the base-station (left). Items are color coded according to the legend at the bottom.

C++ was still the language of choice for both the user interface and onboard the vehicle. Compartmentalization between major functional components was limited, minimizing overall code size and ensuring reliable interaction between components, but making expansion and debugging difficult. Different functional components were typically run as separate threads with very regulated methods for passing data amongst themselves. On the vehicle these components included the autopilot interface, XBee communications, vehicle control, waypoint management, and data logging. The vehicle control thread was the most important, handling navigation of the vehicle while passing data to and from both the autopilot and XBee interfaces. Onboard computing also allowed for new safety behaviors, such as setting a home location for the vehicle to automatically return to if a specified time elapses with no communication from the vehicle. The software running on the user's laptop was simple by comparison, having just XBee and GUI components.

As testing and experiments on vehicle control continued, it became clear that a map-based interface would make the vehicle much easier to operate, especially over long distances. A map with satellite or terrain imagery would allow the user to easily input waypoint commands relative to important landmarks and monitor the vehicle's location while receiving video. Such an interface was first implemented using the Google Maps API and could be viewed through Internet Explorer. A simple webpage with some JavaScript exchanged data with the C++ GUI application using several text files on the operator's computer. Waypoints set in the GUI application, whether manually or by importing from a file (generated by Matlab, for example) were written to a file and then displayed on the map a few seconds later. Conversely, mouse clicks on the map could be configured to write a location to file, which could then be used in the C++ GUI to set a new waypoint or instantly send the vehicle to that location.

Current vehicle coordinates, destination, and home location were also displayed on the map using the same method of text files to transfer information.

While not particularly elegant, using text files to interact with the Google Maps API was the easiest implementation of a mapping interface and had several advantages. Most importantly, the mapping half of the interface was completely handled by Google Maps, eliminating the need to manually produce and display a tiled map image. Zooming, panning, and other standard interface functions were smoothly handled by the Google Map's interface and required no work to implement. However, Google Maps required that there always be internet access available to download the map tiles. With internet frequently unavailable for field operations, especially at sea, an alternative solution to the mapping problem was pursued.

At this stage of the development process, the software running on the operator's computer was all running in Microsoft Windows. Based on recommendations from Tawfiq Tafer at CENSAM, C# replaced C++ for code running locally. C# allows for many of the same capabilities as C++ while natively including a powerful graphical interface design tool in Microsoft Visual Studio. The Visual Studio Integrated Development Environment (IDE) made development much easier and faster at the expense of being limited to running on Windows; code running on the vehicle was still written in C++. C# was used to write a new mapping interface, fully integrated into the GUI used to control the vehicle rather than running in a web browser on the side. Map tiles were pre-fetched from Google Maps and stored on the operator's computer where they could then be retrieved by the GUI application as needed.

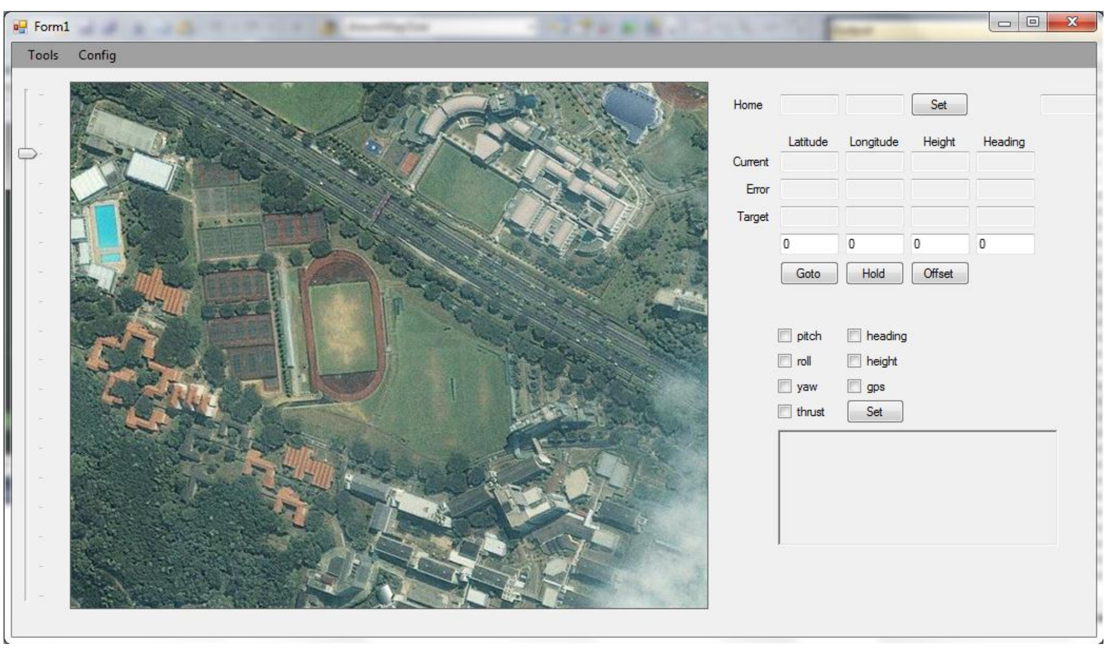

Figure 2-8: Later version of the GUI provides similar functionality, but simplifies the various fields in favor of a map based interface. Waypoints are now displayed directly on the map instead of as a list.

The final version of this software solution, similar to that shown in Figure 2-8, provides all the basic features required for algal bloom missions with only a few limitations. Waypoint control, including lists of multiple waypoints, is fully implemented and allows for autonomous area coverage. Precise coverage patterns can be generated separately in Matlab or another application and imported and displayed in the GUI. Important safety behaviors, such as automatic returning in case of communications failure, are also implemented onboard the vehicle. Wireless communications via the XBee modems is slow but reliable, and acknowledgements and retries are used on important command packets to ensure delivery. Many of the key results that will later be discussed were achieved using this software architecture.

While this software architecture is relatively simple, the approach reduces overall flexibility of the solution. Since various components are not well compartmentalized, even simple modifications to enable a new behavior can require many changes and careful testing to avoid breaking basic functionality. Adding the thermal camera, for example, would add another thread interacting with several of the threads already running on the vehicle and require a prohibitive number of variables related to that interaction. New packets would need to be hardcoded into the XBee communications software to command the camera as well. Expansion to multiple vehicles or integration with surface and underwater vehicles running different software is also difficult. These limitations eventually motivated the adoption of MOOS as is later discussed.

2.4.2 Flight Control

The overall software system starts with control of the vehicle's flight itself. As mentioned previously, the Autopilot on the quadrotor can be commanded by a computer using two different control schemes. Complete waypoints including latitude, longitude, altitude, heading, and travel speed can be sent to use the Autopilot's built-in waypoint controller. For lower-level control, pitch roll, thrust, and yaw rate commands can be sent and the Autopilot will control the individual rotors to achieve the desired settings. While the waypoint controller is obviously easier to use, initial experiments focused on developing our own controller using the lower level interface option. A successful controller using these commands could allow us to automate the launching and landing procedure, a feature not supported by the Autopilot's waypoint controller. Flight could also be optimized for the payload onboard by attempting to control the vehicle more smoothly or enforcing specific pitch and roll angles when acquiring images.

2.4.2.1 Low-level Control

The four lower level input commands offered by the Autopilot allow us to independently control the quadrotor's motion in several degrees of freedom. The symmetric design of the quadrotor means the vehicle is equally capable of moving in any lateral direction, controlled almost entirely by the pitch and roll of the vehicle. Yaw angle affects only the alignment of the body-fixed pitch and roll angles with respect to a global frame, so it can easily be corrected for. Thrust may affect lateral motion of the vehicle, in that a higher thrust at a given angle of tilt will result in higher acceleration. However we expect the vehicle to typically operate at constant altitude and hence near constant thrust, so the potential effects of different thrust on lateral motion are ignored. Just as lateral motion can be controlled solely through pitch and roll of the vehicle, altitude and heading are controlled only through the thrust and yaw commands respectively.

The accuracy that can be achieved in the various degrees of freedom is affected by both the type of control input and the sensors available to provide feedback. We also have different goals for control accuracy in each degree of freedom. For example, lateral position and velocity feedback is provided only by the GPS, which is accurate to within approximately three meters in ideal circumstances. Fortunately we expect the vehicle to be traveling between waypoints that may be 100s to 1000s of meters apart, so accuracy on the order of a few meters in more than sufficient. Only in landing or launching situations would we need to consider achieving higher accuracy. Heading control is made easier because the control input is yaw velocity. Using internal gyros, the Autopilot can maintain near-constant heading when given a zero yaw input command. Furthermore, the compass used to measure heading is accurate to within a few degrees, allowing for precise yaw control.

Since the yaw control is in the form of a yaw rate command, control in yaw was easily implemented with a constant velocity towards the desired heading with a dead band several degrees wide. GPS control was implemented using a simple PID without any track-line control for these initial experiments. Different gains were used based on distance to the target waypoint. At distances greater than 5 meters, higher proportional gain was used to increase the translational speed of the vehicle, while a smaller proportional gain reduces the response of the vehicle to GPS noise when at the target waypoint. In both cases, maximum limits were imposed on the tilt of the vehicle to effectively limit the maximum speed.

Figure 2-9 illustrates the performance of the near GPS controller during position holding about the origin (marked in red). Despite wind gusts on the order of several meters-per-second and GPS noise on the order of one to two meters, position is maintained within approximately 1.5 meters of the target. It should be noted that the only source of position measurement available is GPS, so these plots cannot fully show the effects of GPS noise over small position changes.

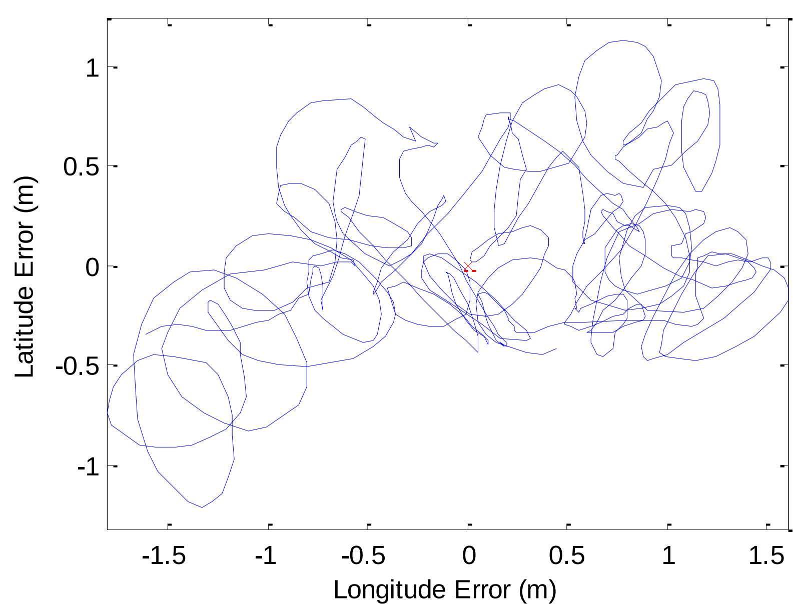

Figure 2-9: 165 seconds of position holding under autonomous GPS control.

The track of the vehicle during a fifty meter step in position is shown in Figure 2-10 and the distance from the target waypoint is plotted against time in Figure 2-11. Despite the lack of a cross-track controller, the quadrotor remains within a few meters of a straight line between its starting position and the target waypoint. Velocity towards the waypoint is maintained at a near-constant four meters-per-second the vehicle is within five meters of the target waypoint. In both plots the transition from the far to the near PID gains is clearly visible and most notable for the sudden decrease in velocity.

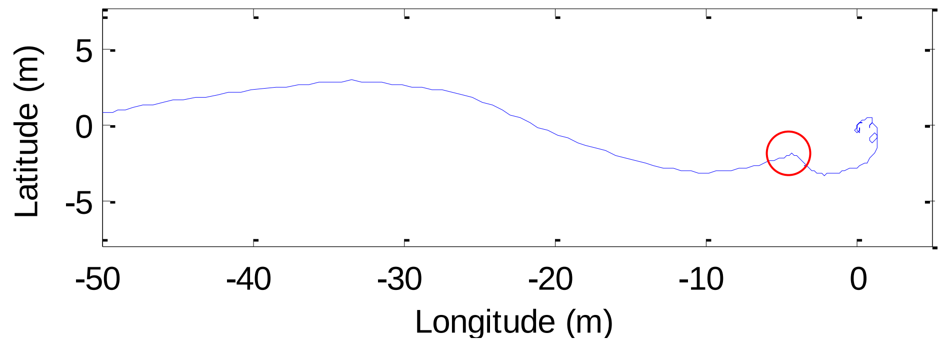

Figure 2-10: Track of the quadrotor's GPS coordinates during a fifty meter translation in longitude. The target location is at the origin. The transition region from far to near GPS controller is circled in red.

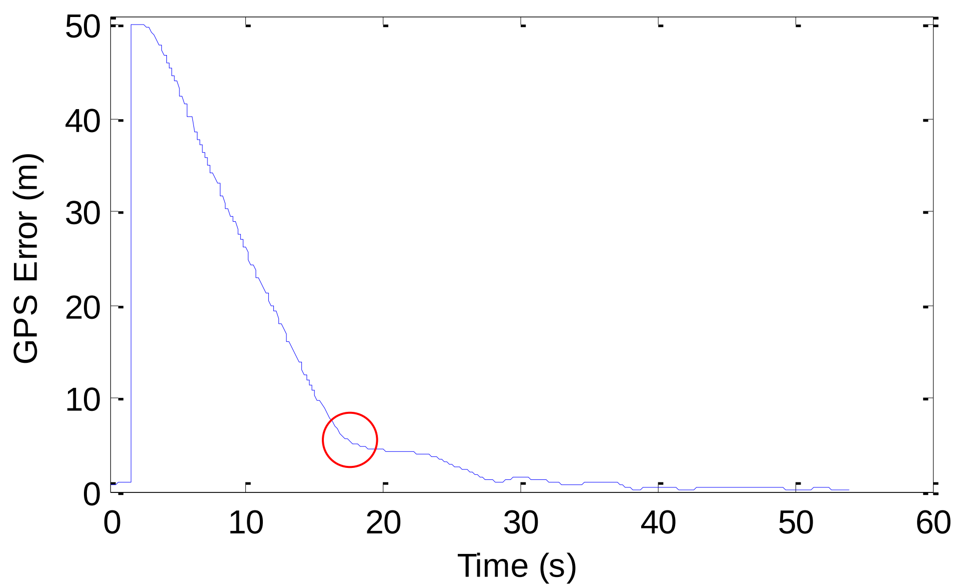

Figure 2-11: The quadrotor's distance from its target as a function of time for the track shown in Figure 2-10. Again the transition from far to near GPS controller is circled in red.

Throughout these tests heading was maintained to within approximately five degrees of the target heading. We expect about this much error due to the dead-band in the heading controller, which was designed to avoid oscillations in heading with noise in the compass measurement. With the quadrotor's omnidirectional design, the GPS controller performs similarly regardless of current heading. This is especially useful when carrying a camera payload because yaw rotation can be used solely to adjust the camera's field of view.

Altitude control is much more difficult than lateral position control because the vehicle's altitude is sensitive to small changes in thrust and sensor noise is on the same order of magnitude as the vehicle's operating altitude. GPS, a highly sensitive pressure sensor, and accelerometers contribute to sensing of altitude and each sensor has its own unique characteristics. Earlier control experiments also utilized a downward facing ultrasonic rangefinder, but the sensor proved too unreliable with respect to the type of surface underneath the vehicle. For example, surface water waves or propeller downwash over grass could cause erroneous range measurements or no range measurements at all.

GPS's vertical accuracy is much worse than its horizontal accuracy with measurement errors often exceeding 10 meters. However, over long periods of time we expect the GPS height measurement to average about the correct altitude reading. Measurements from the pressure sensor behave in exactly the opposite manner. The sensor suffers from long term drift and is subject to changes in atmospheric pressure associated with everyday weather patterns, but can accurately measure altitude changes that occur on time scales of several seconds to minutes. The pressure sensor can also be affected by wind, or even the vehicle's own velocity. The onboard Autopilot performs some initial filtering of the pressure sensor data and reports both a height and velocity measurement. Since we want to use the pressure sensor primarily to detect shorter-term dynamics we choose to use only the velocity measurement. The accelerometer provides reasonable acceleration measurements on the sub-second scale, but cannot be accurately integrated more than once without extensive filtering. Its measurements are more useful to the Autopilot for flight stabilization than they are for our own height controller.

Sensor fusion is accomplished through the use of a Kalman filter, designed in continuous time and converted to a discrete-time representation through Matlab's c2d function. The Kalman filter expects measurements to have pure Gaussian noise, but as discussed this is not a reasonable assumption for the sensors here. Instead, noise coloring is used to add non-Gaussian noise for the GPS and pressure sensors. For the pressure sensor, we use the second order low pass filter

$$\frac{colored\ noise}{white\ noise} = \frac{Y}{\omega_x} = \frac{1}{(\tau s + 1)^2} = \frac{1}{\tau^2 s^2 + 2\tau s + 1} \qquad (1)$$

to represent the presence of low-frequency noise, where $1/\tau$ is the desired cutoff frequency. In matrix form this becomes:

$$\begin{Bmatrix} \ddot{y} \\ \dot{y} \end{Bmatrix} = \begin{bmatrix} -2/\tau & -1/\tau^2 \\ 1 & 0 \end{bmatrix} \begin{Bmatrix} \dot{y} \\ y \end{Bmatrix} + \begin{bmatrix} 1/\tau^2 \\ 0 \end{bmatrix} \omega_x \qquad (2)$$

To incorporate the accelerometer measurement we need to include the $\ddot{x}$ as one of our states, where x is the altitude of the vehicle. With $\ddot{x}$ as a state, $\dddot{x}$ appears in our state equation and needs to be assigned a process noise, which will be integrated into the acceleration. Since integrating white noise is effectively a random walk, which will diverge away from zero, we also need to color the process noise just as was done for the pressure sensor such that only the high frequency noise is integrated. The resulting state equations are:

$$\begin{Bmatrix} \dddot{x} \\ \ddot{x} \\ \dot{x} \\ \ddot{a} \\ \dot{a} \\ \ddot{y} \\ \dot{y} \end{Bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2/\tau_a & -1/\tau_a^2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2/\tau_p & -1/\tau_p^2 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{Bmatrix} \ddot{x} \\ \dot{x} \\ x \\ \dot{a} \\ a \\ \dot{y} \\ y \end{Bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \omega_{a_2}/\tau_a^2 \\ 0 \\ \omega_p/\tau_p^2 \\ 0 \end{bmatrix} \qquad (3)$$

where a and y are states added for noise coloring. τ_p and τ_a are the time constants associated with the low frequency of the pressure sensor and third derivative, respectively, while ω_p and ω_{a_2}set the DC magnitude of the noise. Noise coloring is not being used on the GPS measurement in this example. For these state equations the associated sensor equations used in the Kalman filter are:

$$\begin{Bmatrix} acc \\ p_{rate} \\ gps \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \ddot{x} \\ \dot{x} \\ x \\ \dot{a} \\ a \\ \dot{y} \\ y \end{Bmatrix} + \begin{bmatrix} \nu_{acc} \\ \nu_p \\ \nu_{gps} \end{bmatrix} \qquad (4)$$

where *acc*, p_{rate}, and *gps* are the measurements of vertical acceleration, velocity, and position by the accelerometer, pressure sensor, and GPS respectively. The high frequency white noise associated with each of these sensors is defined by ν_{acc}, ν_p, and ν_{gps}. Note the “1” in the fifth column, which adds the low frequency colored noise to the pressure sensor’s rate measurement.

Measurements of the vehicle’s weight and thrust force at varying control inputs contributed to a rough model of the vehicle’s dynamics in the vertical direction. We assume small velocities in the vertical direction, rarely exceeding one meter-per-second, so aerodynamic drag is not included in the model. A Kalman filter using this simplified model and including the noise-coloring techniques discussed above was implemented on the vehicle running at 20 Hz. Similar to the heading controller, the altitude controller set a constant desired velocity in the vertical direction until the quadrotor was within a set vertical error, usually on the order of 0.25 to 0.5 meters. A PID controller operated on the state estimate of the Kalman filter to achieve the target vertical velocity by setting the thrust command on the vehicle.

The performance of the altitude controller was qualitatively compared to the altitude control provided by the onboard Autopilot when operating via RC in the ACC+height mode. Various experiments tested different process and sensor noise values, as well as adjusting the PID gains, to improve both step responses and performance when translating at constant altitude. Figure 2-12 shows a step response with relatively good results. As with tests of the GPS controller, no absolute altitude reference was available, so this plot cannot tell the complete story. Constant velocity is maintained while increasing altitude to the target and the vehicle eventually settles at the desired altitude, but oscillations about the target are slow to die down.

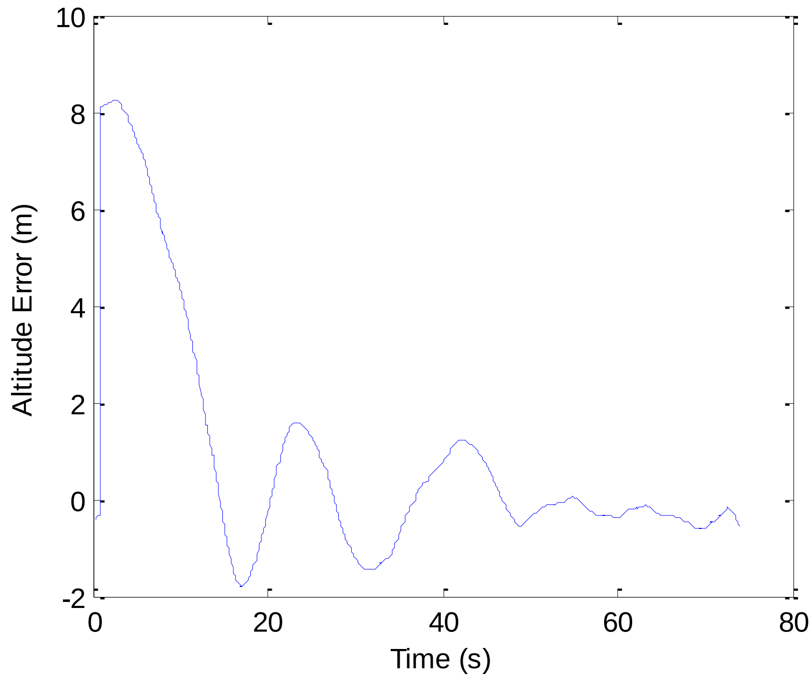

Figure 2-12: Response of the altitude controller to an 8 meter positive step response (altitude is increasing).

The altitude controller proved sufficient for initial field testing but was not reliable enough to be trusted in operations over water, nor did it meet the performance of the Autopilot's built in height controller. Our altitude controller is very dependent on the vehicle model to determine the initial hovering thrust used to fly the vehicle, which means that any payload change requires updating the model with the new mass of the vehicle. Attempts to increase the integral gain to quickly correct for errors in the starting thrust offset resulted in decreased performance during step responses, and sometimes even instability. The built in controller, perhaps through the use of gain scheduling, performs similarly for any feasible payload with no modification required.

Ultimately further development on the height controller was abandoned in favor of using the built in waypoint controller, sacrificing any capabilities requiring finer control of the vehicle. In the future, automated launching and/or recovery of the quadrotor may require additional work on the low level controller, especially for altitude, based on the methods and results reported here.

2.4.2.2 Waypoint Control

Based on the difficulty of developing of developing a controller reliable enough to be used in over water operations, we decided to develop the larger system around the waypoint controller provided by the Autopilot onboard the quadrotor. While launching and recovery must be performed manually via RC, the controller itself is backed by years of development at Ascending Technologies and can be depended on for consistent operation over water. Custom modification of the vehicle was provided by the manufacturer to allow the waypoint controller to function without the RC present, allowing for operation at the longer ranges possible with the XBee 868 MHz modems. A single target waypoint can be sent to the Autopilot and includes absolute GPS latitude and longitude, altitude in meters, a heading relative to North, and a speed in percent at which the vehicle will travel.

An early characterization of the waypoint controller's performance is shown in Figure 2-13. A capture radius of several meters is defined at each waypoint and the vehicle is required to hover within that radius for several seconds before proceeding to the next point. Our custom control software running on the Gumstix onboard the vehicle enforces these holds by monitoring the vehicle's position and deciding when the next waypoint may be transmitted to the Autopilot. The consistent offset to the left of each track line is the result of a calibration offset in the compass. Later and much longer trials, even in the presence of crosswinds, yielded very small cross track error relative to the distance between waypoints.

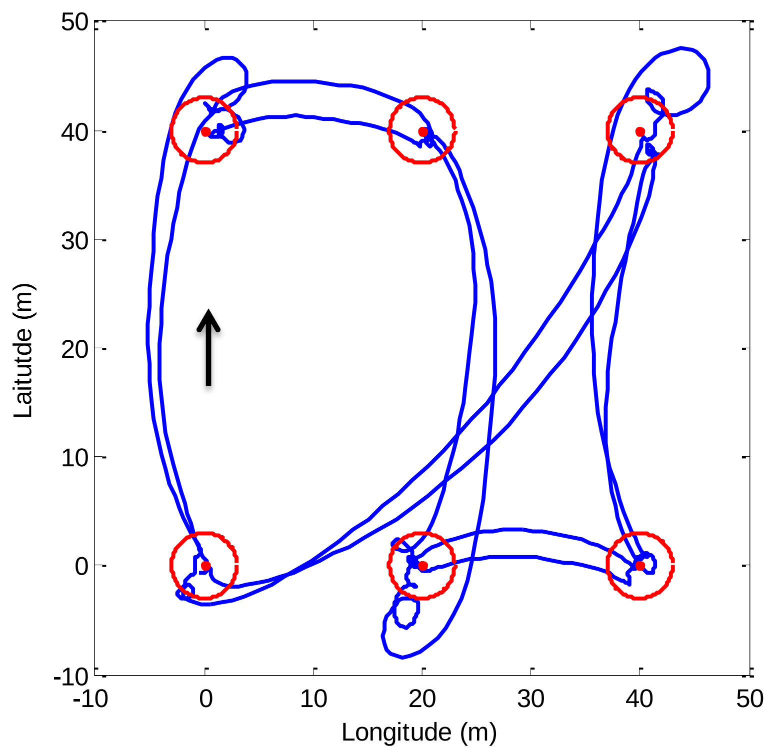

Figure 2-13: Position track of the quadrotor using the Autopilot's waypoint controller to cycle among six waypoints. The direction of travel is indicated by the arrow.

Longer tests also showed the average speed of the vehicle to be about seven meters-per-second, slightly less than the ten meters-per-second maximum advertised by Ascending Technologies. The vehicle's straight line velocity between waypoints is plotted in Figure 2-14, which shows consistent speeds within each traverse between waypoints. Different traverses exhibit different speeds because of varying wind speed and direction relative to the vehicle's course. Speed variation can be attributed not only to imperfect control, but also varying wind conditions and GPS measurement noise (affecting both the controller and our speed measurement).

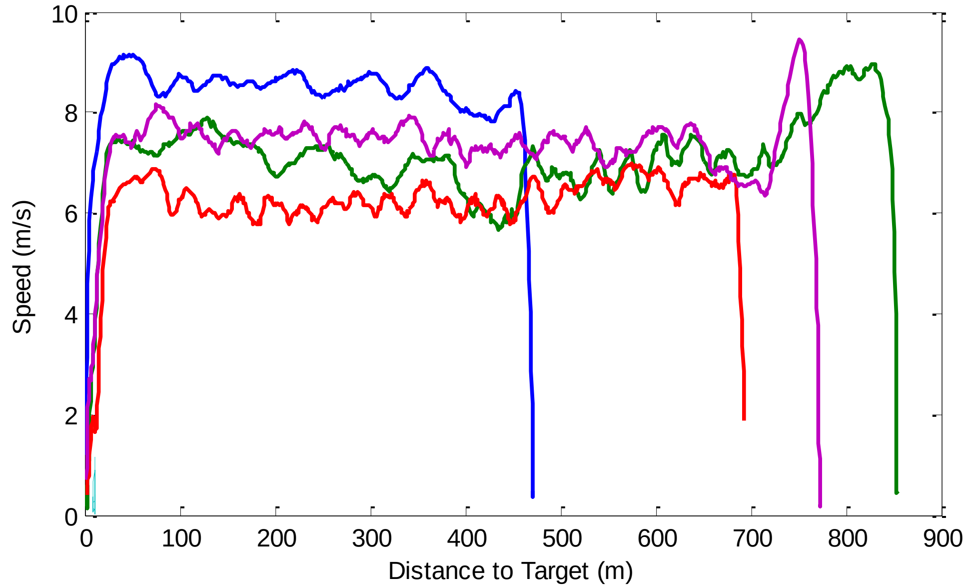

Figure 2-14: Quadrotor's speed as measured by GPS when traversing in a straight line between distant waypoints. Each color corresponds to a traverse between a different pair of waypoints.

The Autopilot's height controller appears to depend primarily on the pressure sensor for its altitude measurement. Altitude is consistently controlled to within a fraction of a meter of the target height, even with different payloads resulting from switching sensors or battery sizes. The pressure sensor is susceptible to slow drift and waypoints commands sent to the Autopilot must take into account the initial altitude reported by the vehicle at the start of a mission. The drift is typically small over the course of a mission though, never varying more than a few meters between launch and recovery and not posing a risk to operations at twenty meters. For autonomous recovery or longer missions (with a higher endurance vehicle), GPS or other absolute measurements would also be required to avoid endangering the vehicle.

2.4.3 MOOS-Based Architecture

As discussed earlier, the simplicity that made the original system architecture appealing also motivated the use of prewritten robotics software. A number of such packages are available and almost all contain some method for separate processes to exchange information. This means that different components of the vehicle's software, such as communications and control, can be developed and debugged independently of each other. Additional modules, such as drivers supporting new sensors, can also be added without necessitating changes to other software components.

Though MOOS stands for Mission Oriented Operating System, it has evolved into what is now popularly known as robotics middleware, software designed to ease development of code on robotics platforms. At its most basic level MOOS is a message-passing service that allows multiple processes running on a computer to share information. A number of prewritten MOOS applications, or MOOS-Apps, provide basic functionality such as logging and communication across multiple communities. MOOS-IvP adds applications specifically designed for operating marine vehicles. We will discuss both the basic concepts of MOOS that make it appealing for use on quadrotors and the applications that were developed to make MOOS running on the quadrotors a reality.

2.4.3.1 Exploring MOOS

MOOS is a set of cross-platform C++ libraries and applications that ease development of complete software control systems in robotic applications. The most fundamental layer, CoreMOOS, provides network communications for multiple processes running on a single or across multiple computers. All communication is passed through a communications hub called the MOOSDB. Processes cannot communicate directly to each other; instead they must follow a publish/subscribe system using the MOOSDB. A process can publish data to the MOOSDB in the form of a MOOS variable, either a single number or a string that contain almost any amount of data. When a variable is posted to, the MOOSDB notifies any applications that have subscribed to that variable of the change. Applications communicating through the same MOOSDB form a MOOS community. In most situations a single MOOS community runs on a single computer or vehicle.

Applications that implement the MOOS libraries are called MOOSApps and can be easily written in C++ using a predefined CMOOSApp class. This class provides all the basic methods an application needs to publish messages, subscribe to variables, and receive new mail from the MOOSDB. Each MOOSApp has an OnNewMail and Iterate method, which are called at a configurable frequency. This frequency and other parameters can be configured in a .moos file, which is read at runtime. This architecture is well suited to robotics where applications have to perform a specific function, such as running a controller, at a certain time interval.

Essential MOOS adds several applications for common tasks such as process control and logging. *pAntler* is a special launching application that can be configured to start multiple MOOSApps so that the user does not have to start each one individually. While called a database, MOOSDB does not behave like a true database in that it only keeps track of the most recent value of each variable that has been posted. *pLogger* can be configured to log any or all variables in the MOOSDB in either a synchronous or asynchronous format. *pMOOSBridge* allows variables to be communicated between MOOS communities running on separate communities, such as between a vehicle community and a community running on an operator’s personal computer. Other applications ease debugging by allowing a user to monitor or set values manually while MOOS is running.

MOOS-IvP is a much larger set of applications and libraries specifically designed to facilitate MOOS’s implementation on autonomous marine vehicles. The most unique component is *pHelmIvP*, a behavior-based application that performs a multi-objective optimization between competing behaviors to select a target course, speed, and possibly depth (depending on the vehicle type). The helm usually functions as a “back-seat driver”, making high-level autonomy decisions and passing its commands to a vehicle’s built-in controller, though other MOOSApps can also be used to perform low-level control of a vehicle. Behaviors can be anything from simply travelling between waypoints to attempting to explore and characterize some phenomenon such as a temperature plume.

Other MOOS-IvP applications include utilities for debugging and monitoring the helm, log parsing, simulation tools, and user interface applications for working with vehicles in the field. *pMarineViewer* is a GUI tool that can render vehicles, waypoints, and other information on a map while allowing a user to send control commands by interacting directly with the map. The mapping and other GUI components were previously a significant portion of the software development effort because of the complicated nature of any GUI software development. Using *pMarineViewer* allows us to focus software development on the control and communication components that are unique to the quadrotors.

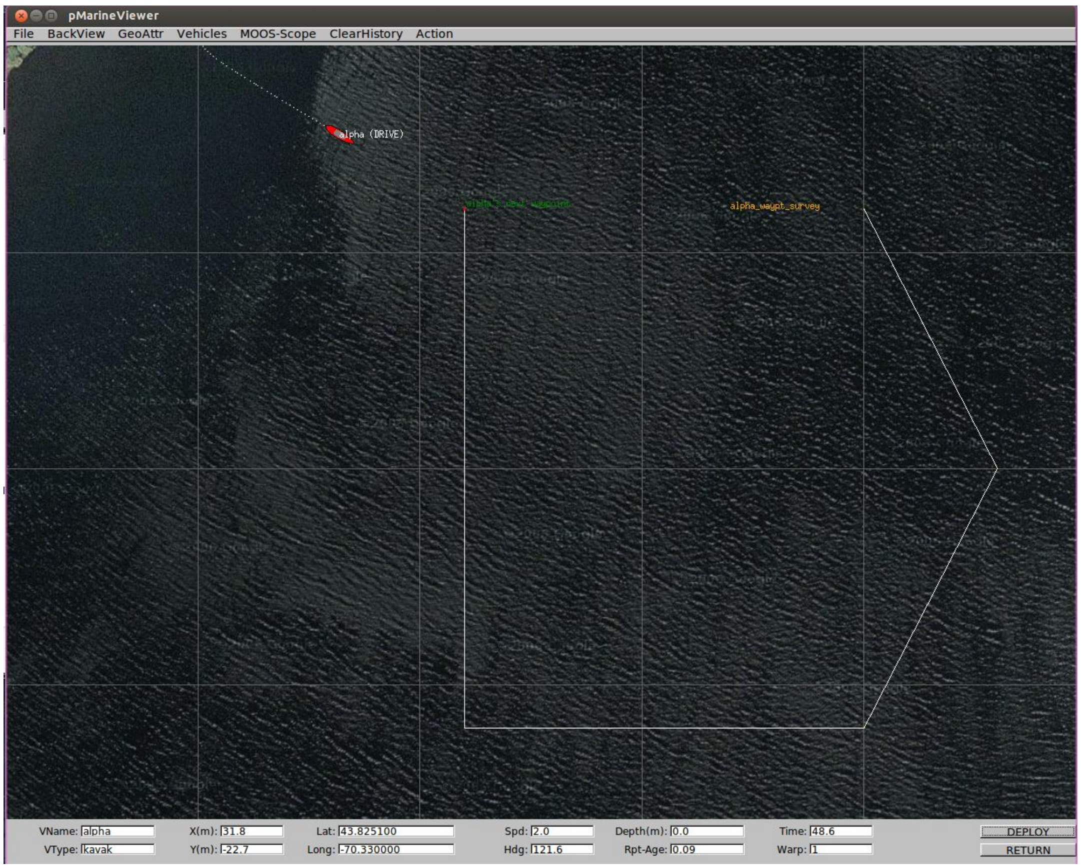

Figure 2-15: Screenshot of the pMarineViewer application being used to monitor and control a simulated vehicle.

The use of MOOS on quadrotors is appealing for several reasons beyond the use of *pMarineViewer* as a control interface and the obvious benefits of any inter-process communication architecture. The CENSAM autonomous surface vehicles used in Singapore also run MOOS, potentially simplifying communication and integration with the quadrotors. MOOS-IvP is also primarily developed at MIT making support and education more readily available.

2.4.3.2 Implementing MOOS on Quadrotors

While MOOS-IvP is designed to be as close as possible to a drop in solution in most marine vehicles, the specifics of our hardware require several customizations for basic operations. From a software architecture perspective, the quadrotor differs in two key areas: communications and control. Communication in MOOS utilized TCP (or UDP in some cases), a protocol available in standard wired or

802.11x wireless solutions. The XBees used to communicate with the quadrotor communicate via a serial interface and do not support TCP, so *pMOOSBridge* cannot be used to link the vehicles with an operator's computer. Secondly, *pHelmIvP* outputs heading and speed control commands, but we must pass complete waypoints to the Autopilot on the quadrotor. These two issues led to the development of several custom MOOS applications and libraries that allow the quadrotor to be controlled from MOOS and an operator to use *pMarineViewer* to monitor and command the vehicles.

iUFOInterface is a driver for handling communication with the Autopilot. It formats target waypoints into the correct structure and sends them to the Autopilot while also constantly requesting new sensor information and posting it to the MOOSDB. Serial communication with the Autopilot is handled via a background thread and modified versions of this application have been used on other vehicles to communicate with their respective low-level controllers.

quadHelm serves as a very limited replacement for *pHelmIvP*, providing basic waypoint and safety functionality. Due to communications limitations, only single waypoints and not waypoint lists are currently supported. *quadHelm* keeps track of the quadrotor's current target waypoint and is responsible for forwarding the target waypoint to *iUFOInterface*. It can also be commanded to hold position, generating a waypoint from the vehicle's current position on the spot. Finally, a home location is saved for safety purposes. If no communication is received from the operator for greater than 10 seconds, *quadHelm* will automatically instruct the Autopilot to return to the location from which the vehicle was launched, but traveling at an increased altitude.

The various behaviors already written for *pHelmIvP* would take considerable time to implement from scratch on the quadrotor, so future work may include adaptation of *pHelmIvP* to work on the quadrotor despite the difference in control methods. Completion of the previously discussed control work could allow heading, speed, and altitude commands to be used. Another potential solution would be to replicate the heading and speed control functionality by sending the Autopilot distant waypoints at the requested heading with a speed percentage defined to meet the required speed. We have not yet tested how the Autopilot responds to many waypoint commands in close succession, but this approach could save considerable work in improving our own controller and would be extensible to other vehicles sharing a similar low level controller.

xbeeInterface replaces *pMOOSBridge* for communication between the MOOS community on the vehicle and that running on an operator's laptop. Messages received by *xbeeInterface* are encoded and sent via serial to the XBee modem. At the other end another instance of the application receives the serial data from an XBee, decodes it, and publishes the command or data to one or several MOOS variables. Due to the limited communication bandwidth available, we did not try to replicate the *pMOOSBridge's* flexibility and easy configuration. Instead of specifying which variables should be transmitted in the .moos configuration file, most standard transmissions are hardcoded into the program using rigid data structures to reduce the overall number of bytes that need to be transmitted. Since minimum and maximum values as well as the needed precision are known for variables such as latitude/longitude or heading, 2 or 4 bytes can be used instead of the 8 usually required for a floating point value. Waypoint commands to the vehicle and position/status updates from it are all handled in this manner. To allow for some flexibility, a single MOOS variable can be transmitted, but consumes considerably more bandwidth since the complete variable name must be sent as well.

The current version of this application brings low bandwidth communications to MOOS, but still suffers from some of the same limitations as the original software implementation described in Section 2.4.1. Any packet type that is used frequently must be hardcoded into the application, making even the addition of a single status variable difficult. The application also assumes direct communication between the shore station and the vehicle via the XBee link. The deployment of multiple vehicles would require some form of destination and source addressing, either through the packets themselves or by utilizing the XBee's API, which allows each packet to be given a specific destination address. Both would require another layer of processing between the current packet encoding/decoding and the XBee hardware.

quadReporter is a simple replacement to *pNodeReport*, a MOOS-IvP application that combines important navigation and status data into a single string report. *quadReporter* publishes a single variable containing GPS latitude/longitude, speed, heading, altitude, and voltage, which is read by *xbeeInterface* and transmitted to the operator's computer once every 2 seconds.

quadTranslator runs on the operator's personal computer and serves as an intermediary between *pMarineViewer* and *xbeeInterface*. Commands in *pMarineViewer* are reflected in posting to the MOOSDB, which *quadTranslator* reformats and passes to *xbeeInterface* for transmission to the vehicle. In the other direction, *quadTranslator* reposts data reports received from the quadrotor via *xbeeInterface* in the correct format to be displayed in the *pMarineViewer* GUI.

Lib_nav_data_types defines the data structures used to pass information between applications running on the vehicle. Methods are included for serializing and parsing data to and from strings which are posted to the MOOSDB. Waypoints and quadReport types are both defined in this library.

Lib_simple_log provides a thread-safe logger used by individual applications if a potentially large number of debug messages need to be recorded. For example, *iUFOInterface* can be configured to log the success or failure of every packet exchanged with the Autopilot. While these messages could be posted to the MOOSDB and logged through *pLogger*, earlier versions of many applications already depended on this logging library and there was little reason to remove this separate logging functionality.

The complete architecture including MOOSApps and the types of data they exchange is described in Figure 2-16. Some supporting applications, such as pLogger, are omitted for simplicity. However, all of the main applications and those that differentiate this implementation of the MOOS architecture from that on a traditional surface vehicle are included. Communication between the two communities is not handled over a TCP/IP capable link such as WiFi, but through a low-bandwidth connection managed by *xbeeInterface* and further aided by *translator* and *quadTranslator*. Onboard control is managed by a simple *quadHelm* application instead of the more complex *pHelmIvP*. Finally, *iUFOInterface* handles serial communication with the quadrotor's built-in Autopilot.

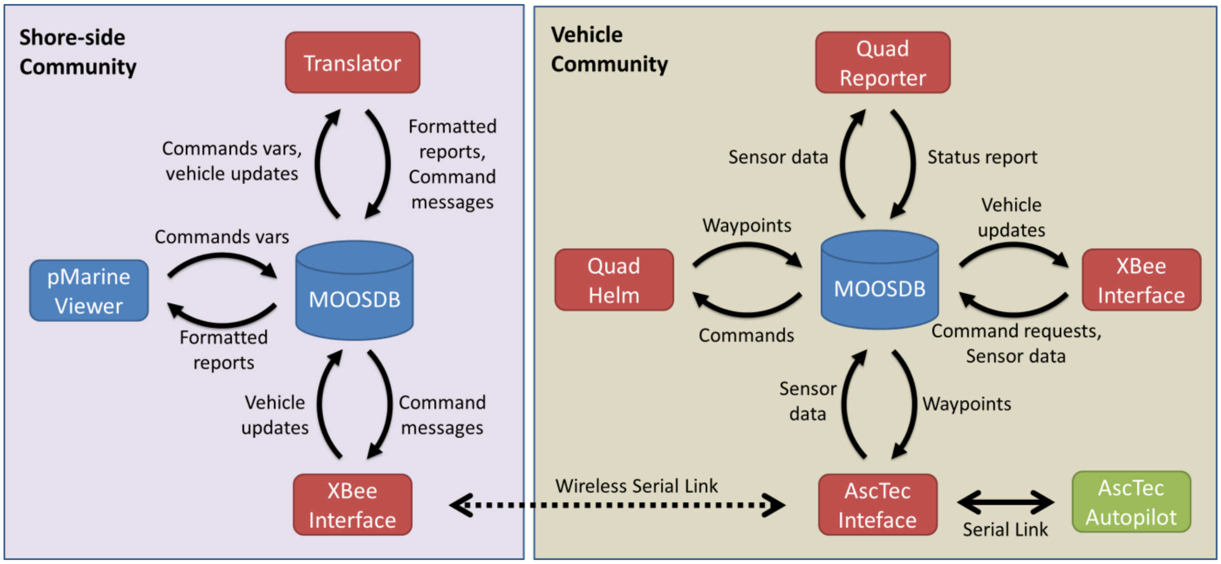

Figure 2-16: The MOOS based architecture used to control the quadrotor.

2.4.3.3 Thermal Plume Extensions

To integrate the Tau 320 thermal camera with the MOOS software framework another driver application, **iFlir** was developed with the help of fellow graduate student Jacob Izraelevitz to interface with the camera. iFlir performs all the necessary tasks to command the camera to take snapshots, download those snapshots, and save them in several different formats on the Gumstix's Micro SD card. The time at which each snapshot is taken is posted to the MOOSDB for synchronization with position and attitude information in post mission processing. Currently, snapshots from the thermal camera are not processed onboard the vehicle, nor are they transmitted over wireless due to bandwidth considerations.

Tools were also written to visually analyze the data files saved by the thermal camera by converting them to 8-bit grayscale images. The 14-bit temperature values recorded by the camera are mapped linearly onto 8-bit values to cover either the full range of 14-bit values seen (losing some temperature resolution) or a subset of the 14-bit range (clipping the values outside the range). These tools can be used to generate images such as those previously shown in Figure 2-3.

3 Algal Bloom Operations and Results

The complete system including vehicle, sensors, and software has been tested and deployed at sea trials in Singapore supporting the detection and study of algal blooms. Across several trials, operational methods have been developed that culminated in the successful detection of algal blooms on two separate dates. Some image processing techniques have also been applied to the resulting images and video to gauge the potential usefulness of real time image processing for this application.

3.1 Operational Methods

The described system is not designed to be operated standalone, but in conjunction with surface and underwater vehicles under the direction of hydrodynamic and biological models. Planning for sea trials starts early with the selection of regions and dates of interest. Marine biologists determine the best locations to look for algal blooms based on models of the process by which blooms develop. Local bathymetry, river outlets, and even manmade developments play a role in identifying locations with a higher probability of bloom formation. Tides contribute through the mixing of nutrients by tidal currents. As such, dates are usually selected to coincide with spring tides, when the sun reinforces the moon's gravitation pull and the tide's range is at its maximum.

Algal bloom sightings in Singapore have thus far been limited to the narrow Johor Strait that separates Singapore from Malaysia, so most sea trials have been performed in this area. Figure 3-1 includes the location of the fisheries where a Harmful Algal Bloom caused fish deaths in 2010. The Senoko power plant is also circled in the northwest of the image for reference; it is also a site of interest for thermal plume experiments. Many small channels and inlets border the Johor Strait and there is a large river flowing from Malaysia (just visible in the far northwest corner of Figure 3-1.) Current flows are difficult to model, but they can contribute to the formation of algal blooms and cause them to move or disperse in unpredictable ways.

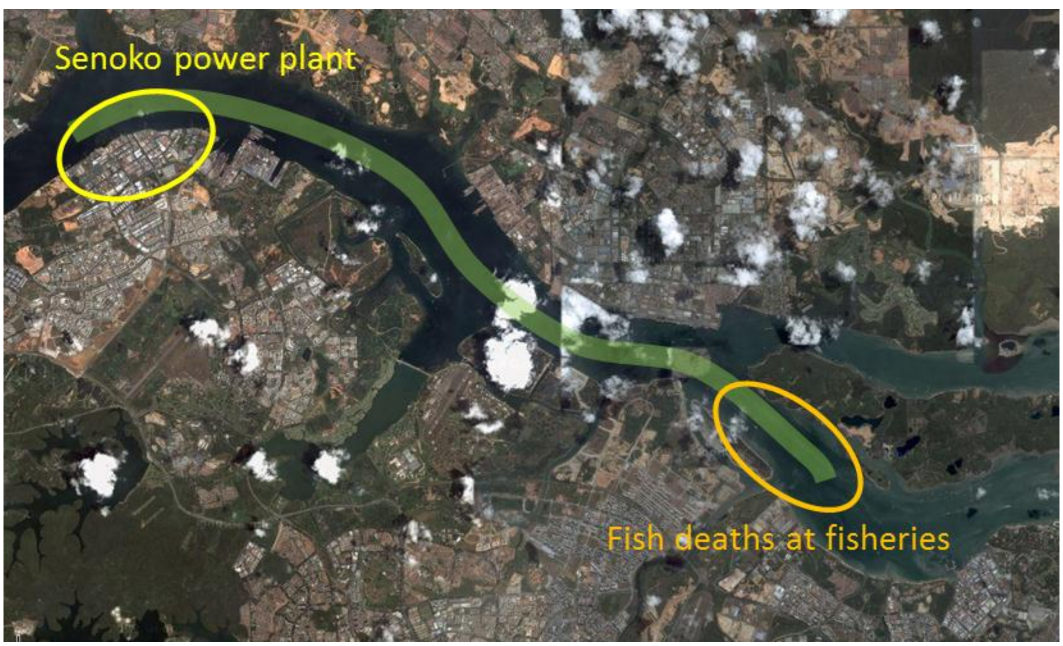

Figure 3-1: Satellite imagery of eastern Johor Strait with Singapore to the south and Malaysia to the north. Sea trials have been located through the green band.

On a typical operations day a scientist working on the biological models helps inform search areas for kayak and quadrotor deployments. Often times the regions near inlets and side channels are of particular interest. Being more cumbersome to launch, multiple kayaks are launched early, each equipped with a different sensors, such as CTDs or mass spectrometers. Ideally the surface vehicles should always be gathering data to maximize the utility of the available time at sea. Even if an algal bloom is not found, the data can still be used to verify and improve hydrodynamic and biological models.

Individual quadrotor missions are much shorter, lasting no more than 25 minutes due to the limitations of battery life, and involve greater human interaction. An area where algal blooms are believed likely to form is selected and if desired, waypoints may be generated in advance. The vehicle is launched manually by remote control before being switched over to waypoint control by the onboard computer. Once proper waypoint behavior is verified the waypoint mission is started following either the predetermined waypoints or operator commands in real time. Mounting the camera at an approximately 45-degree downwards angle increases the field of view and ease of use, but also makes image quality very dependent on orientation. The combination of ambient winds and vehicle translation mean the vehicle is almost always operating with some tilt angle, either increasing or decreasing the camera's field of view depending on heading. Sunlight reflections can also be quite strong, so the heading is continually adjusted once the mission is underway to reduce the effect of glare while maintaining a useful field of view.

The video feed provided by the camera onboard the quadrotor is monitored for any sign of algal blooms. Waypoints are often modified manually after a possible sighting, reorienting the vehicle to view the same area from a different angle or looking for edges of a patch. Since blooms are difficult to find, almost all sightings are followed up with closer investigation by surface or underwater vehicles. Kayaks already operating in the area can be redirected with new waypoint missions to explore the area where a potential bloom was sighted. Visually identifying the edges of the bloom is important because it allows surface vehicles to later cross these edges and compare key parameters inside and outside the bloom.

While the quadrotor has limited endurance to stay on station and monitor the bloom, it can be quickly returned to its base if needed and be redeployed with a fresh battery, allowing almost continual monitoring of a bloom. In-situ measurements can provide more information about the bloom's properties, but may require some processing before the in-bloom and out-of-bloom portions of the data can be differentiated. The visual image provided by the quadrotor is a very direct and easy-to-use measurement of where the bloom is and where it isn't. It is often useful to see the surface vehicles in the quadrotor's field of view to further verify their position with respect to the bloom.

3.2 Field Results

These operational techniques have been employed on several sea trials in the Johor Strait of Singapore and led to the successful identification and subsequent measurement of two algal blooms to date. In both cases the initial visual identification by the quadrotor was instrumental to finding the bloom and guiding the deployment of surface vehicles for in-situ measurements.

In January of 2011 the first sighting of an algal bloom was made with an early iteration of the quadrotor system. The vehicle was operated solely by manual RC control, using visual observations and video images to determine its position. A small support boat was used to deploy the quadrotor, allowing the operator to remain in visual range of the quadrotor while operating in shallower waters. During one of the final missions of the day a bloom was spotted less than thirty seconds after launching the quadrotor from the support boat. Sample images pulled from the video recording are shown in Figure 3-2. From an aerial perspective the bloom is very easy to spot as a brown patch at the water surface. While clearly visible here, the bloom could not be seen by a person standing on the support boat until the boat was driven nearly on top of the bloom to collect water samples. Reviewing the video feed shows how the bloom appears to suddenly materialize as the quadrotor gained altitude and flew above it. An extremely wide angle "fish-eye" lens also shows us how the bloom becomes harder to see near the edges of the image, where the viewing angle is shallowest.

Figure 3-2: Algal bloom spotted in January of 2011 is clearly visible as a brown patch on the surface of the water.

On this occasion the bloom proved transient. After approximately fifteen minutes of observation the quadrotor was returned for a fresh battery and then redeployed to the same area. As shown in Figure 3-3, the bloom that was once clearly identifiable is impossible to spot. In-situ measurements provided by surface vehicles confirmed the presence of an algal bloom even after it was no longer visible on the surface. A combination of strong currents and the short lifespan of algal blooms could both have contributed to the visual disappearance of the bloom.

Figure 3-3: Two pictures of the same region taken less than twenty minutes apart, illustrating the transient nature of algal blooms.

In August of 2011 a more fully developed version of the quadrotor system was used to successfully identify and monitor an algal bloom. Detection was accomplished at a distance of several hundred meters from the ship of operation while waypoint control allowed the vehicle to more easily explore the vicinity and return to the same location after redeployment. As seen in Figure 3-4, the bloom is again clearly visible as a splotchy discoloration on the sea surface.

Figure 3-4: Algal bloom spotted in August 2011. A pleasure craft is also present in the left image. The bloom is harder to identify in the right image, but some splotches are still visible near the bottom of the frame.

After initially detecting the bloom the quadrotor was manually controlled to identify the area over which the bloom could be clearly identified visually. Surface vehicles were directed with both GPS coordinates and visually using the quadrotor's video feed. Figure 3-5 shows such an autonomous surface vehicle passing through the bloom as recorded by the quadrotor. Visual references like these are useful in verifying the location of the surface vehicle relative to the overall bloom and determining what contributes to the visibility of algal blooms. Better modeling the bloom during its visible lifespan can help inform future quadrotor missions to improve the probability of finding algal blooms in the future.

Figure 3-5: An autonomous surface vehicle carrying in-situ sensors passes through an algal bloom.

Neither the January 2011 nor the August 2011 blooms were harmful, but both provided baseline information on algal blooms in Singapore and helped further the development of the overall sensing system.

3.3 Image Processing

As operated, the quadrotor-based system for finding algal blooms depends heavily on having a human-in-the-loop for both navigational input and more importantly for monitoring of the video feed. Looking for potential algal blooms in the video is a small burden for the operator when working with just one or two vehicles, but would quickly become infeasible if more vehicles or vehicles with greater endurance were used. Some automated image processing to pick out potential blooms could make the overall system easier to operate and allow the quadrotor to make its own decisions based on video feedback, enabling more complex behaviors such as mapping the boundaries of a bloom. Since the current system transmits video data to the base in real time, there is no concern regarding computing resources for initial image processing implementations.

Many people are performing research in the field of machine vision and we do not seek to introduce new algorithms or methods to the field. Rather, we explore the success of existing image processing algorithms applied to the specific problem of identifying algal blooms. As will be shown, this application proves quite challenging.

3.3.1 Color-Based Analysis

The most obvious approach to the problem is one based on color, since that is how a human viewing video from the quadrotor picks out a potential algal bloom. To a human the color of the bloom appears quite distinct from the normal color of the water. For example, clear boundaries are visible in Figure 3-6.

Figure 3-6: Image of an algal bloom taken by the quadrotor.

The difficulty for a computer becomes evident when the image is broken down into individual pixel colors. Figure 3-7 plots these values from Figure 3-6 in HSL (hue, saturation, lightness) space. HSL is a cylindrical coordinate representation of the traditional RGB colors designed to be more intuitive with respect to how humans perceive color. Saturation can vary from 0 to 1, but we see only a range of small saturations and hues in this algal bloom image. Furthermore, the colors form a continuous spread with no clear divisions at which to draw a dividing line. The problem is compounded by the wide viewing angle of the lens. At shallower viewing angle the color seen by the camera will be increasingly dominated by reflection, not the water or its contents. This is visible by the bluer areas in the upper right and left corners of Figure 3-6, which are also clearly visible in the HSL plot. With color also varying between different days, locations, vehicle heading, and blooms, it would be impossible to select a single determining color or range of colors to be considered as potential blooms.

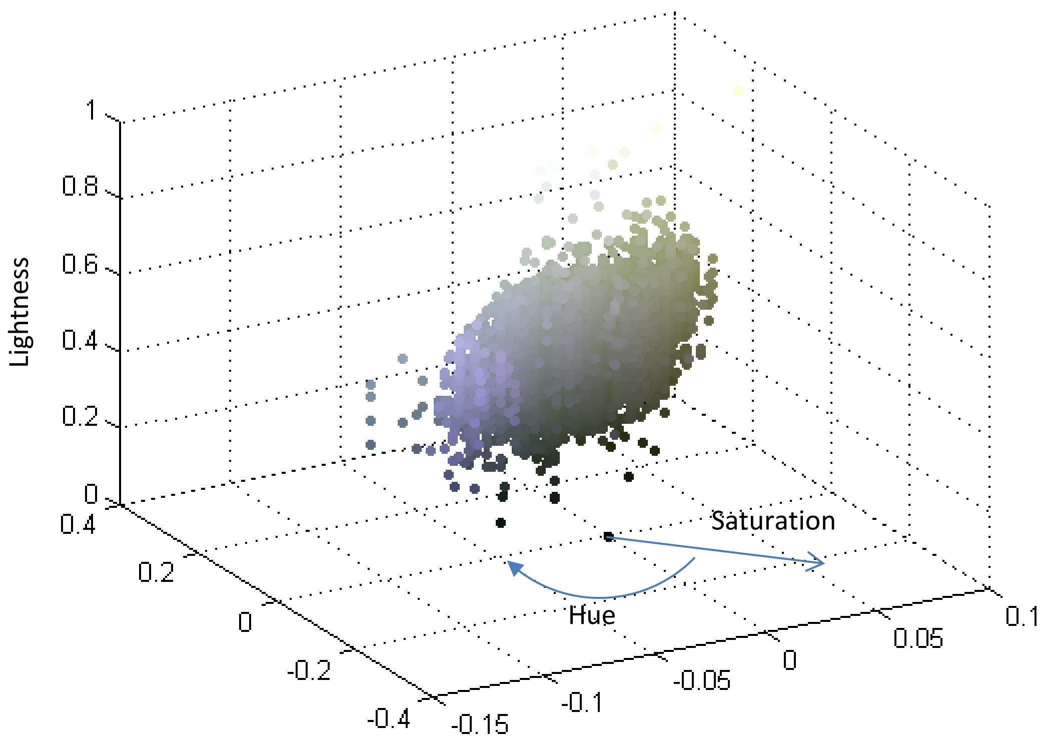

Figure 3-7: Breakdown of algal bloom image into individual pixel colors, plotted in the HSL (hue, saturation, lightness) color space.

Similar analyses using RGB representation and a conversion to YCbCr, another color space, were even less fruitful. A simple color comparison is not sufficient for identifying the green-to-brown blooms seen in Singapore.

3.3.2 GrabCut – A Segmentation-Based Approach

Many of the bloom pictures exhibit clear boundaries between the bloom itself and the surrounding, usually darker, water. Image segmentation algorithms look for such boundaries to pick out objects from an image, such as finding a person in a foreground of a picture while disregarding the background. First introduced in 2001, graph cuts use an energy minimization approach operating on the grayscale values of an image to determine whether each pixel belongs in the foreground or the background (Boykov, 2001). GrabCut improved upon the graph cut approach by making the process iterative, using Gausian Mixture Models to handle colored images, and sharpening the border between foreground and background (Rother, 2004). In our application we can apply either method by treating the bloom as the foreground and clear water as the background.

GrabCut is not a fully automated algorithm and depends on user feedback to identify areas of an image that are or aren't part of the foreground. The user is given four methods for interaction. First, a box can be drawn around the foreground object, allowing the algorithm to classify all pixels outside the box as part of the background. Two more methods allow the user to draw lines on the image identifying pixels that belong to either the foreground or the background. This can be useful to improve GrabCut's performance in difficult areas after an initial iteration. The final interaction method allows the user to highlight areas with detailed edge features, forcing GrabCut to treat those areas differently. This method is not useful in our application because we only need to identify rough boundaries (and probably could not identify exact boundaries even if we tried).

The software library OpenCV includes a self-contained implementation of the GrabCut algorithm and was used to perform two tests on images containing algal blooms as shown in Figure 3-8 and Figure 3-9. In the first test, initial user interaction marks the pixels under the blue lines as part of the background – water that does not contain an algal bloom. In the second image only a rectangle is drawn, and everything outside the rectangle is marked as part of the background. In both cases it is hard to evaluate the algorithm's performance near the top of the image where the viewing angle becomes shallow. In many of these images it is impossible, even with a human perspective, to identify the top edge of the bloom because the shallow viewing angle ultimately limits what can be seen. The second set of images does show some promising results along the upper left border of the selected foreground, where the final cut closely matches the visible browner area.

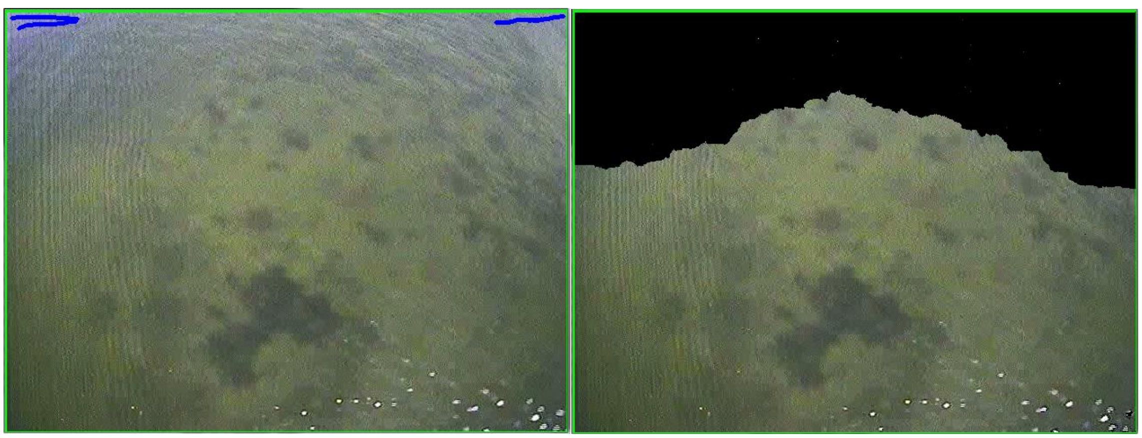

Figure 3-8: GrabCut algorithm applied to an aerial image of the algal bloom. The only user input provided came from defining the pixels under the blue lines as belonging to the background.

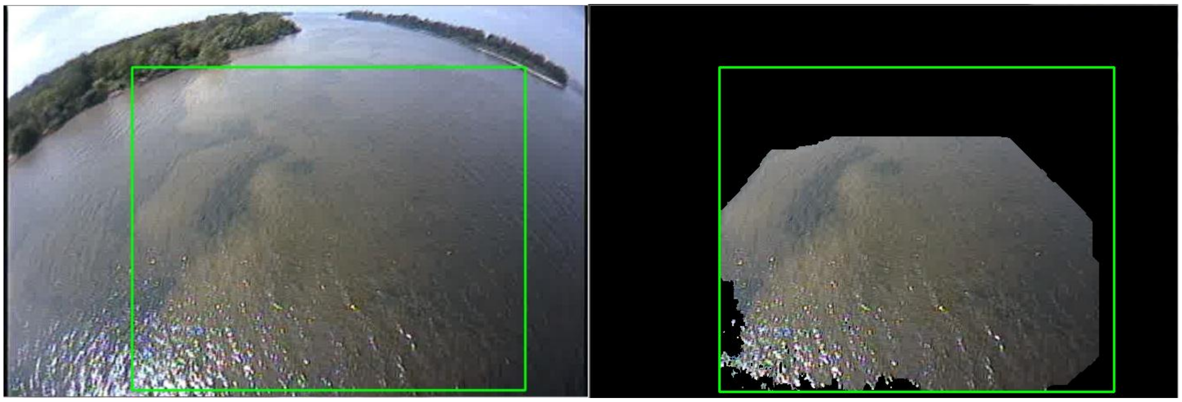

Figure 3-9: The wide viewing angle leads to worse performance of the GrabCut algorithm. Initial user input takes the form of a green rectangle, marking everything outside it as part of the background.

GrabCut shows some success on individual images, but it is not clear how it could best be applied to a continuous video stream such as that provided by the quadrotor. Some initial human interaction could possibly be applied to all incoming images based on color to prevent the need for continuous operator feedback. The GrabCut algorithm also lacks any built-in metric for when the iterative process should be stopped. When applied to some images, consecutive iterations eventually converge towards a constant area, such as in Figure 3-8 where 13 iterations were used, but little change occurred after 6. In other cases the algorithm will shrink the foreground selection to nothing with too many iterations and it is up to the user to stop the process.

3.3.3 Future Work

A great deal of future work can still be done to seek a viable image processing solution. Building a mosaic of the covered area is one of the most useful potential steps. It could help image processing methods and humans alike better identify and determine the extent of potential algal blooms. Mosaicking is still a challenging feature to implement though; images would have to be aligned based purely on internal sensor data since most images will not contain any static features that could otherwise be used. Images would also have to be selected from the continuous video stream, taking into account intermittent transmission interference, glare, and viewing angle.

If image processing were to be used as the primary detection method instead of human interaction, then a camera facing directly downwards instead of at a forward angle could reduce the issues associated with shallow viewing angles seen previously. Currently human interaction provides the most reliable method for identifying and reacting to potential bloom sightings. If two or more vehicles were deployed simultaneously, or a single vehicle with much greater endurance, then image processing could become much more important in reducing the workload for an operator.

4 Proposed Thermal Plume Operations

Being a more recent project, the quadrotor system has not yet been used in field experiments for the study of thermal plumes. As discussed previously though, testing with the thermal camera has confirmed its ability to acquire the required data, albeit at very limited frame rates of approximately one frame per 20 seconds. Further controlled experiments, either in the lab or with in-situ measurements for reference, will be needed to confirm the accuracy of the temperature measurements and any drift that will require calibration by in-situ measurements in the field.

4.1 Operational Methods

The methods for studying thermal plumes begin similarly to those for algal blooms with the definition of a search area. For the thermal plumes visited so far, the areas are well defined in advance; the location of the outflow pipe is known and the flow itself is strong enough to be visible on the surface. The area of the plume is used to define an initial coverage area for the quadrotor over which a lawnmower-style mission can be planned. Individual waypoints are automatically generated to cover the space. Overlap between consecutive snapshots is not critical because the data within each image cannot be used for mosaicking.

The final temperature field will be reconstructed based on the quadrotor's GPS coordinates and orientation as measured by the onboard flight controller. To achieve an accurate position fix and allow time to download individual snapshots from the camera, the quadrotor will need to pause for several seconds at each waypoint before acquiring an image. Ultimately the rate at which the area can be covered is limited by the slow speed at which images are downloaded from the thermal camera into the Gumstix's memory. Complete coverage may require multiple deployments due to the vehicle's reduced endurance when carrying the heavier thermal camera.

More advanced behaviors may be implemented after initial testing using standard lawnmower patterns. Unlike the images acquired for algal blooms, the thermal camera's snapshots do not require any difficult image processing before they can be fed to a model of the thermal plume. Even a simple model, running either on the vehicle or on shore, could provide a variance measure or similar to identify areas where the temperature field is least well known. Since the vehicle's speed is fast compared to the rate at which data can be collected, implementing even a simple next-best view behavior could significantly increase the speed at which the temperature field can be reconstructed. Such an approach could also help direct the quadrotor towards areas where the field is most dynamic. These areas are likely to also have strong currents, making in-situ measurement by surface or underwater vehicles difficult.

The vehicle has not been used in a thermal plume study yet, but it has been successfully deployed over water with the thermal camera equipped to gather several images. Two such images were shown in section 2.1.2 and one is repeated here in Figure 4-1.

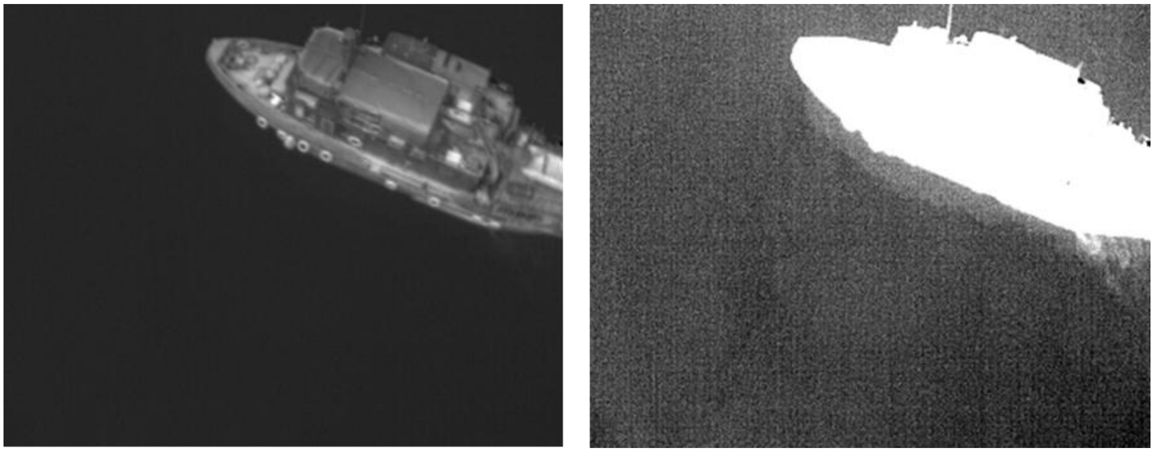

Figure 4-1: Thermal image taken by the quadrotor while hovering above a ship at sea. The image has been post-processed to show the full range of temperatures at left and a much more limited range at right.

4.2 Cross-platform Integration

Like the algal bloom missions, the study of thermal plumes requires coordination with surface and/or underwater vehicles, but in a much more organized manner. When searching for algal blooms, the quadrotor's mission is largely independent, simply forwarding coordinates to other vehicles or human operators when a potential algal bloom is found. On the other hand, the thermal camera used to collect data on thermal plumes is providing one of the same data types collected by surface and underwater vehicles and must be used in the same model of the thermal plume's temperature and current fields.

The sensors used on the different platforms each have their own strengths and weaknesses. The in-situ measurements performed on surface and underwater vehicles are more accurate, but only collect data in the narrow lines traversed by the vehicle. Snapshots from the thermal camera may not be as accurate, but cover a larger area in each image and are better suited to studying fast, dynamic events. The measurements from the camera must also be calibrated occasionally, requiring the quadrotor and surface vehicles to perform measurements in the same location simultaneously. The different sensor characteristics, vehicle capabilities, and operational constraints pose a difficult mission planning problem.

Complete integration between the surface, underwater, and aerial vehicles as well as a model running in real-time remains the subject for a future project. Current work focuses on real-time modeling in conjunction with the surface vehicles carrying CTDs for temperature sensors and pan-tilt mounted ADCPs for measuring currents throughout the thermal plume.

5 Conclusions

We have demonstrated the design and successful operation of a new robotic system for remote observations in marine environmental sensing, built around a small unmanned aerial vehicle. While the capabilities of underwater and surface vehicles traditionally used for marine sensing continue to improve, they still face significant challenges when measuring spatially and/or temporally dynamic phenomena. Harmful algal blooms and industry-related thermal plumes are two such phenomena present not just in Singapore, but in other coastal environments around the world. An aerial vehicle is well suited in these applications because remote sensing with a video or thermal camera is both feasible and useful. The high speed and aerial viewpoint of such a vehicle gives it an advantage compared to slower surface or underwater vehicles.

A complete system was built using relatively inexpensive off-the-shelf hardware and designed to facilitate operator interaction in real time and in coordination with other vehicles. A quadrotor-style vehicle provides a reliable and stable platform on which either a video or thermal camera can be mounted. Data is provided in real time, allowing an operator to react and modify the mission or direct the deployment of surface and underwater vehicles carrying in-situ sensors. The MOOS/MOOS-IvP software framework was leveraged for inter-process communication, user interfacing, and logging functionality while also easing future communication with surface vehicles.

Field trials in Singapore have proven the effectiveness of the quadrotor system for the detection of algal blooms on two separate dates. Surface vehicles subsequently deployed to the sites were able to measure key parameters with in-situ sensors, confirming the presence of an algal bloom. The initial reconnaissance performed by the quadrotor was essential to finding the blooms and giving operators a direct measurement of their location and extent.

Initial tests with the thermal camera have demonstrated its basic capabilities for capturing full 14-bit precision data onboard the quadrotor. Further work is required to characterize the calibration of the thermal camera and design missions accordingly. Full integration with surface vehicles and thermal plume models will enable adaptive sampling with minimal human-in-the-loop interaction.

Many fields stand to benefit from the current proliferation of commercially available and inexpensive aerial vehicles. The system designed here is oriented towards two specific sensing tasks in a marine environment, but the same techniques could be used to develop similar systems for urban, jungle, or other environments. Small but powerful embedded computing devices combined with existing software packages such as MOOS/MOOS-IvP can be used to quickly integrate an autonomous vehicle, aerial or otherwise, into an existing sensor network. Some of the methods discussed here involving embedded hardware, software, and communications have already been applied to ongoing work targeting surface vehicles and underwater nodes in support of acoustic communication research.

Bibliography

[1] Achtelik, M. (2009). Autonomous Navigation and Exploration of a Quadrotor Helicopter. *AUVSI First Symposium on Indoor Flight Issues.* Mayagüez.

[2] AOML. (2012, July 31). *The Global Drifter Program*. Retrieved from The Global Drifter Program: http://www.aoml.noaa.gov/phod/dac/index.php

[3] Ascending Technologies GmbH. (n.d.). (Ascending Technologies GmbH, Konrad-Zuse-Bogen 4, 82152 Krailling, Germany) Retrieved from http://www.asctec.de

[4] Bluefin Robitics. (2012, July 13). *Bluefin Robotics*. Retrieved from www.bluefinrobotics.com: http://www.bluefinrobotics.com/

[5] Boykov, Y. (2001). Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images. *IEEE International Conference on Computer Vision*, (pp. 105-112). Vancouver.

[6] Bushaw-Newton, K. a. (1999). *State of the Coast Report: Harmful Algal Blooms.* Silver Spring, MD: National Oceanic and Atmospheric Administration.

[7] censam. (n.d.). *center for environmental sensng and modelng*. Retrieved from censam.mit.edu: http://censam.mit.edu/

[8] Center for Disease Control. (2012, January 13). *Harmful Algal Blooms (HABs).* Retrieved from www.cdc.gov: http://www.cdc.gov/nceh/hsb/hab/default.htm

[9] Curcio J., L. J. (2005). SCOUT - A Low Cost Autonomous Surface Craft for Research in Cooperative Autonomy. *MTS/IEEE Oceans 2005*, *1*, pp. 725-729.

[10] Digi International Inc. (n.d.). Retrieved from http://www.digi.com/

[11] FLIR. (2012). *Tau Uncooled Cores*. Retrieved from www.flir.com: http://www.flir.com/cvs/cores/view/?id=51374

[12] Fugro OCEANOR. (2012). *Buoys and Sensors*. Retrieved from www.oceanor.com: http://www.oceanor.com/systems/seawatch/buoys-and-sensor/

[13] Hong Kong Agriculture, Fisheries and Conservation Department. (2012, May 8). *Hong Kong Red Tide Information Network.* Retrieved from www.afcd.gov.hk: http://www.afcd.gov.hk/english/fisheries/hkredtide/redtide.html

[14] ImmersionRC Ltd. (n.d.). Retrieved from http://www.immersionrc.com/

[15] Lucieer, A. (2011). Unmanned Aerial Vehicle Remote Sensing for Hyperspatial Terrain Mapping of Antarctic Moss Beds based on Structure from Motion point clouds. *International Society fo Photogrammetry and Remote Sensing.* Sydney.

[16] Massachusetts Department of Publc Health. (2012). *Red Tide Fact Sheet.* Retrieved from www.mass.gov: http://www.mass.gov/eohhs/broken/red-tide-fact-sheet.html

[17] MBARI. (2004, April 29). *OASIS: Ocean Acquisition System for Interdisciplinary Science*. Retrieved from www.mbari.org: http://www.mbari.org/bog/Projects/MOOS/methods.htm

[18] MBARI. (2010, August 17). *Upper-water-column vehicle*. Retrieved from www.mbari.org: http://www.mbari.org/auv/upperwater.htm

[19] McGillivary, P., Fall, K., Maffei, A., Schwehr, K., Twiggs, B., Kitts, C., et al. (2006). elf-Positioning Smart Buoys, The "Un-Buoy" Solution: Logistic Considerations using Autonomous Surface Craft Technology and Improved Communications Infrastructure. *OCEANS*, (pp. 1-5). Boston.

[20] Mellinger, D. (2012). Trajectory generation and control for precise aggresive maneuvers with quadrotors. *The International Journal of Robotics Research*, 665-674.

[21] OceanServer. (2012). *OceanServer IVER2 Autonomous Underwater Vehicle*. Retrieved from www.iver-auv.com: http://www.iver-auv.com/index.html

[22] ONR. (n.d.). *Liberdade XRay Advanced Underwater Glider.* Retrieved from www.onr.navy.mil: http://www.onr.navy.mil/~/media/Files/Fact%20Sheets/advanced_underwater_glider.ashx

[23] Quek. C., a. L. (2010, January 1). 200,000 fish in farms off Pasir Ris dead. *The Straits Times*.

[24] Richardson, L. (1996, July-August). Remote Sensing of Algal Bloom Dynamics. *BioScience, 46*(7), 492-501.

[25] Rother, C. (2004). "GrabCut" - Interactive Foreground Extraction using Iterated Graph Cuts. *Special Interest Group on Computer Graphics and Interactive Techniques (SIGGRAPH).* Los Angeles: Association for Computing MAchinery.

[26] Shen, S. (2011). Autonomous Multi-Floor Indoor Navigation with a Computationally Constrained MAV. *IEEE International Conference on Robotics and Automation.* Shanghai.

[27] Shepard, A. N. (2011). Cost-effective Monitoring and Vanguard Technologies for Ocean Management. *Ocean News & Technology, 17*(4).

[28] Wendel, A. (2011). Natural Landmark-based Monocular Localization for MAVs. *IEEE International Conference on Robotics and Automation.* Shanghai.

[29] WHOI. (2012, 8 1). *Spray Glider*. Retrieved from www.whoi.edu.

[30] WHOI. (n.d.). *Conductivity, Temperature, Depth (CTD) Sensors*. Retrieved from www.whoi.edu: http://www.whoi.edu/instruments/viewInstrument.do?id=1003

[31] Zheng, H. O. (2009). Design of Field Experiments for Adaptive Sampling of the Ocean with Autonomous Vehicles. *12th International Conference on the Enhancement and Promotion of Computational Methods in Engineering and Science.* Macao.