Hydra: Underwater Swarm Robot System for Efficient

Deepwater Search

ECE4011 Senior Design Project

Project Hydra Advisor: Dr. Mick West

Team Members

Priyansh Bhatnagar, <u>pbhatnagar6@gatech.edu</u> Trevor Jones, <u>tjones6289@gmail.com</u> Hemanth Koralla, <u>hkoralla3@gatech.edu</u> Aneri Muni, <u>amuni3@gatech.edu</u> Vishnu Perumal, <u>vishnu69@gatech.edu</u> Michael Tatum, <u>michaelptatum@gatech.edu</u>

Submitted

Executive Summary

A flight recorder, or a black box, is an electronic recording device placed in an aircraft to facilitate the investigation of aviation accidents and incidents. The flight recorder is fitted with an underwater locator beacon (ULB) or underwater acoustic beacon (UAB). Once immersed into water, a built-in "water switch" activates the beacon by closing an electric circuit, and the beacon starts emitting its "pings"; the battery power should be sufficient for at least 30 days after the activation.

The disappearance of Malaysia Airlines Flight 370 demonstrated the limits of the contemporary flight recorder technology, namely the need for the ULB's range and battery life to be extended. Current beacons are typically supplied with electrical power by a lithium battery, thus giving them a limited lifespan. This makes recorder retrieval a time sensitive mission. It would be more efficient and, in some cases, safer to send robots instead of divers for such missions.

The goal of our project was to develop a swarm robotics system of autonomous underwater vehicles (AUVs) that would aid in the search and retrieval missions for flight recorders. Each robot in this system contained infrared sensors to communicate with one another and determine the location of the missing black box. The cost for developing a prototype swarm was approximately \$1000.00. The robots demonstrated the ability of decentralized control algorithms to command swarms of AUVs through tests while maintaining a specified formation. These prototype tests were used to present the project's applicability to real-world problems, such as the search and retrieval of a flight recorder.

Table of Contents

Ex	Executive Summary (5 pts) II		
1.	Introduction	n (5 pts)1	
	 1.1 Objec 1.2 Motiv 1.3 Backg 	tive	
2.	Project Des	cription and Goals (15 pts)2	
3.	Technical S	pecification (20 pts)	
4.	Design App	roach and Details (20 pts)5	
	4.1 Design4.2 Codes4.3 Constra	Approach	
5.	Schedule, T	asks, and Milestones (10 pts)13	
6.	Project Den	nonstration (5 pts)	
7.	Marketing a	and Cost Analysis (5 pts)15	
	7.1 Market7.2 Cost A	ing Analysis	
8.	Conclusion	(10 pts)	
9.	References	(5 pts)	
Ар	ppendix A		
Ар	ppendix B		
Ap	ppendix C		

Hydra: Underwater Swarm Robot System for Efficient Deepwater Search

1. Introduction

Hydra is a proof-of-concept of an underwater swarm robotics system that will traverse areas of the ocean that are inconvenient for humans in order to perform tasks such as airplane black box identification and retrieval. The system consists of multiple robots each configured with a microcontroller, thrusters, and various sensors. The team requested \$1,000.00 to fund the prototype of Hydra.

1.1 **Objective**

For the proof-of-concept, it was the objective of the team to design and prototype a system that will allow multiple robots to collaborate with one another in an underwater setting through a combination of infrared signals. Each robot shall communicate with others that are adjacent to it. The robots shall be able to start in any location meet in a common location.

1.2 Motivation

The main motivations for Hydra were removing the need of human presence in dangerous underwater situations, as well as creating the most efficient ocean searching mechanism. Airplane catastrophes are time-sensitive. By having a system of autonomous underwater robots that will efficiently work together to quickly locate a black box, authorities will be able to deal with these situations in a more responsive and effective manner. Products with similar objectives include General Dynamics' Bluefin 12D AUV and OceanServer's Iver3 580. Hydra's aim was to improve on them by focusing on being cost-effective, commercial, and autonomous [1].

1.3 Background

Recently, many efforts and resources are being dedicated towards research on the implementation of swarm robotics systems in underwater settings. Many companies are investing in the research and development of AUVs for commercial products: fishing-drones like PowerRay, photography-focused AUV's such as Trident, and Blueye's underwater drone meant for marine biologists [2]. These different products are slowly beginning to come to market, and are paving the way forward for AUV presence in the commercial sector.

Universities are also focusing on underwater swarm robotics systems. The Autonomous Control Engineering (ACE) Lab at the University of Texas, Austin focuses on creating swarm robotics systems in different settings and how various systems of swarm robots could work together. Their goal is to build a "system of systems," specifically in the underwater realm [3]. Additionally, CoCoRo is a EU funded project with the goal to "efficiently and autonomously search areas of the ocean for specific, hard-to-find targets" [4], an objective similar to that of Hydra.

Existing underwater drones are not economical, and research efforts such as CoCoRo are proprietary and not available to the public. For these reasons, Hydra is focused on cost-minimization and commercialization.

2. Project Description and Goals

The goal of Hydra was to produce a system of aquatic swarm robots that were autonomously able to create and execute an efficient ocean traversal algorithm. The targeted users of this product were the military and authorities responsible for locating airplane wreckages. Three AUVs were designed and prototyped by this group. Each AUV consisted of a waterproof chassis, motors, propellers, sensors, and microcontrollers. The microcontrollers allowed implementation of high and low-level programming, controlling the swarm's performance. Sensors, such as a Raspberry Pi Camera, were used for color detection between neighboring AUVs and IR transceivers were used for communication between neighboring robots.

Each AUV has the following features:

- Ability to transmit and receive data through infrared communication
- Ability to detect and follow a specified color
- Propellers that provide two-dimensional movement capability
- Ability to function for one hour on a single charge

The swarm of AUVs has the following features:

- The robots calibrate to detect their neighboring robot(s)
- Ability to perform the consensus algorithm by bringing all robots together

3. Technical Specifications

This swarm project consists of three robots called HydraBots. Below are all the specifications defining the technical details of an individual robot within the swarm:

Item	Proposed Value	Measured Value	
Minimum dimensions $(l \ge w \ge h)$	22 cm x 22.4 cm x 5.6 cm	8.13in x 8.13in x 4in	
Speed (max)	1/20 cm/s	700 KV (RPM/V)	
Power Interface	Micro-USB	16V, 1.5A LiPo Battery	
Build Material	ABS (plastic)	BPA Free Plastic	

 Table 1. Enclosure Specifications

Item	Proposed Value	Measured Value
Brushless DC Motors	2	14.4V, 11A, 140W, 700 KV
3D Printed Motor Nozzle Housing	N/A	3 in (diameter) x 3.5in, ABS Plastic

 Table 2. Components: Vehicle Actuators

Item	Proposed Value	Measured Value	
Time-of-Flight Sensors	6	N/A	
Three-axis accelerometer	1	N/A	
Gyroscope	1	N/A	
MS5803-14BA Pressure Sensor	1	N/A	
Rain Sensors	5	N/A	
Acoustic modem	1	N/A	
Raspberry Pi Camera Module V2	N/A	Sony IMX219 8-megapixel	

Table 4. Components:	Microprocessor	and C	Communications
----------------------	----------------	-------	----------------

Item	Specification	Measured Value
Processor	Raspberry Pi Zero	Raspberry Pi 3 Model B, 32GB MicroSD Card
Microcontroller	Arduino Pro Mini	Arduino Uno
Visible Light Communication LED Transmitter	1 RGB LED	115.2 kbit/s
Visible Light Communication Receiver	470-527 nm wavelength	38kHz

Table 5. Mission	Requirements
------------------	--------------

Item	Proposed Value	Measured Value
Vehicles	6	3
Inflatable Pool	45" x 10"	6' (diameter) x 20" (depth)
Acoustic Beacon	25-40 kHz	N/A

4. Design Approach and Details

4.1 Design Approach

System Overview

This project consists of two major components: the hardware design and implementation and multi-agent coordination algorithms. Each of the HydraBots contains an Arduino Uno and a Raspberry Pi 3, a Pi Camera, a 16V LiPo battery, propellers, 2 DC brushless motors, Electronic Speed Controllers, and an IR transceiver skirt. Figure 1 shows the overall flow of how all of these elements interact with one another. The software uses Python 3 to implement the concepts from decentralized graph theory to achieve the desired swarm capabilities. OpenCV library is used employed for image processing.



Figure 1. Data flow from sensors and inputs to motors and communication modules.

4.1.1 Hardware Specification

4.1.1.1 Platform



Figure 2. Left: Tupperware HydraBot Right: Final prototype from Mechanical Engineering Team

The chassis for each HydraBot is a watertight tupperware container as seen in Figure 2. The motor nozzles, two on either side, were 3D printed and screwed into the tupperware on either side. The enclosure is made negatively buoyant using dumbbell weights to ensure underwater testing. Epoxy, Marine Goop and Vaseline were used to seal the enclosure and make it waterproof.

4.1.1.2 Sensors

Each HydraBot includes one IR transmitter at the front of the robot and four IR receivers forming a sensor skirts to enable inter-agent communication, as shown in Figure 4. A Raspberry Pi Camera is attached at the front of each HydraBot for color detection.



Figure 3. Infrared communication skirt around HydraBot that shows coverage of its surroundings.

4.1.1.3 Microcontroller/Computer

Each robot uses a Raspberry Pi 3 as an onboard computer along with an Arduino Uno. The ESC motor pairs and IR sensors are controlled using the Uno, whereas the Raspberry Pi is used as the main processing unit for swarming algorithms and image processing and sends motor control commands to the Arduino via serial. The choice of these controllers is due to their small size, that in turn helps limit the size of the overall robot. In addition, both of these processors are economically efficient and provide an adequate number of input/output ports for the necessary number of sensors and motors this project requires.

4.1.2 Software

The Software for this project has three main components: swarming algorithm, communication protocol and simulator design.

4.1.2.1 Coverage Control Algorithms

The main algorithm for multi-agent coordination used for the demonstration was the rendezvous problem accomplished through the consensus equation

$$x_{i}^{'} = \sum_{j \in N_{i}} \frac{||x_{j} - x_{i}|| - d_{ij}}{||x_{j} - x_{i}||} (x_{j} - x_{i})$$

where is x_i ' is the velocity of HydraBot i, x_i is the position of HydraBot i, x_j is the position of HydraBot j, a neighbor of HydraBot i, and N_i is the set of neighbors of HydraBot i. Using this, the HydraBots should always meet at the average of their initial positions. This team's demonstration performing consensus was inspired by this algorithm. In addition, the formation control algorithm and leader-follower algorithm were tested in the Robotarium as possible future implementations on the HydraBots. The formation control algorithm instructs a group of robots to form and maintain a specified shape using the equation:

$$x_{i}^{'} = \sum_{j \in N_{i}} \frac{||x_{j} - x_{i}|| - d_{ij}}{||x_{j} - x_{i}||} (x_{j} - x_{i})$$

where d_{ij} is the distance to be maintained between neighboring HydraBots. Keeping this distance constant will three HydraBots will ensure an equilateral triangle is formed.

The leader-follower algorithm instructs every robot except the leader to follow the leader robot using the equation:

$$x'_{i} = \sum_{j \in N_{i}} (||x_{j} - x_{i}||^{2} - d^{2}) * (x_{j} - x_{i}) * Gain$$

where d is the desired distance between robots. The leader robot is separately instructed to move however the user wants it to. Together, formation control and leader-follower could be combined to create a maximally-spanning configuration that searches a given environment

•

4.1.2.2 Calibration



Figure 5. HydraBots calibrating with Raspberry Pi Camera and OpenCV to maintain configuration

The HydraBots perform calibration by spinning in place until they identify their neighbor. The identification process was done through color detection using the Raspberry Pi Camera and OpenCV. Each HydraBot has a pre-assigned neighbor with the yellow robot's neighbor being the orange robot, the orange robot's neighbor being the red robot, and the red robot's neighbor being the yellow robot. Each HydraBot uses masking through OpenCV to mask out all color in each frame of its video feed besides the desired color. Once enough pixels of the desired color were observed, the HydraBot stops and begins a program to maintain its neighbors color at the center of the frame the Raspberry Pi Camera is receiving. If the color is too far to the right of the frame, the HydraBot moved to the right, and if the color is too far to the left of the frame, the HydraBot moved left. If the color ever left the Raspberry Pi Camera frame, the HydraBot returned to calibration mode, spinning in place, until it rediscovered its neighbor.

4.1.2.3 Communication Protocol

An infrared transmitter and receiver were used to communicate between the various robots within this system. Each HydraBot sent various numbers to indicate which state it was in. By doing so, every member of the swarm was informed of the state of their immediate neighbor. Below is a table that shows the different messages that were sent and their significance.

Message	Significance		
Send nothing	end nothing Raspberry Pi Camera does not see a HydraBot		
0	Raspberry Pi Camera sees another HydraBot but does not receive any IR signal		
1	Raspberry Pi Camera sees another HydraBot and receives an IR signal of 0		
2	Raspberry Pi Camera sees another HydraBot and receives an IR signal of 1		

Table 6. Infrared Messages and Their Significance

The receiving of '2' by all of the HydraBots indicated the completion of the calibration process and the start of consensus, where each HydraBot moves towards its designated neighbor.

4.1.2.4 Simulator Design

The 'UUV_Simulator' is a package, written in Python, containing the implementation of Gazebo plugins and Robotic Operating System (ROS) nodes necessary for the simulation of unmanned underwater vehicles, such as ROVs (remotely operated vehicles) and AUVs (autonomous underwater vehicles). A modified version of this package was used as a basis to demonstrate formation control and leader-follower behavior for underwater robotic swarms. The simulation ends with the swarm approaching a shipwreck at the bottom of the ocean, exhibiting the fulfillment of the proposed goal for this project.

4.2 Codes and Standards

The universal serial bus is used on both the Raspberry Pi 3 and Arduino Uno as the communication channel between the two and also provide the necessary connection with a PC to transport algorithms and code. The IR transmitter encodes the communication message using the SONY protocol, which employs 850 nm wavelength. The message is then decoded by the receivers on neighboring HydraBots.

4.3 Constraints, Alternatives, and Tradeoffs

Constraints for the HydraBot include size, weight, cost, waterproofing, battery power consumption, and sensor range. Each HydraBot is designed with a size and weight constraint to make it aerodynamically ideal for seamless movement through the water, as well as one that results in natural buoyancy. Another crucial aspect of designing these underwater robots is to ensure they are completely waterproof. The battery life for each HydraBot is over three hours when continuously running. The design includes two major trade-offs where different alternatives were debated upon: the use of IR communication vs color detection for neighbor identification, modifying a pre-existing toy vs designing new surrogate vehicle, the use of light versus sound for multi-robot communication, and higher processing microcontroller versus a cheaper alternative. Instead of leveraging pre-existing toys in the market, three surrogate vehicles were designed to demonstrate swarming behavior, because many toys were not big enough and did not provide the capability needed for this project. Communication is a difficult task underwater and underdeveloped technology. Sound is a strong alternative that was avoided because the signal becomes too distorted in short range and infrared light is a simpler solution for short range communication. Furthermore, instead of using IR for neighbor identification during the calibration process, the Raspberry Pi Camera is used for color identification as it yielded more reliable results compared to IR signals.

12

5. Schedule, Tasks, and Milestones

The Hydra team designed and tested robots over the course of several months during Spring 2018. Appendix A contains the Gantt flowchart which shows the tasks that were able to be accomplished concurrently. Appendix B contains the complete Gantt Chart. Each section of the chart was determined by approximating the amount of time needed to accomplish each task. Some tasks comprise of smaller tasks which will be divided among team members. Appendix C contains the Pert Chart which lists the probabilities of completion

6. Project Demonstration

The swarming behavior of the HydraBots were demonstrated during the Capstone Design Expo in a clear water children's swimming pool. This project demonstration contained various components:

- **Live Demo:** Three HydraBots autonomously demonstrated Consensus Algorithm inside the pool, i.e. they were able to meet at a common location irrespective of their starting point.
- **Final Design Prototype:** A skeleton of the final design prototype (given by the Mechanical Engineering Team) was placed on the table and provided for a hands-on experience.
- Swarming Algorithm Simulation: A video was shown which demonstrated a simulation of various swarming algorithms, i.e. leader-follower and formation control, through Gazebo - a ROS simulator.
- Swarming Algorithm Implementation: A video was shown which demonstrated the implementation of various swarming algorithms, i.e. leader-follower and formation control, on robots in the Georgia Tech Robotarium.

- **Poster:** A poster explaining the goal, necessity for, and technical details of Project Hydra was placed at the front of the demonstration;
- Video: A video summarizing the project was shown on a main display. This video explained the problem statement as well as Project Hydra's proposed solution.

Updated Specifications:

The Underwater Learning Robot is no longer being used as the design model for the HydraBots. As a result, the physical specifications of the robot chassis changed - the updated design is a tupperware box. Otherwise, the specifications of the HydraBot remain the same as they are based on the electronics inside such as the motor, microprocessor, and battery.

Prototype Testing:

- Swarming Algorithm: The Robotarium in Van Leer was used to test the implementation of the swarming algorithm. The leader-follower and formation control algorithms were successfully implemented on the Robotarium robots.
- **Surrogate Vehicle:** A single surrogate vehicle was tested in the pool for waterproofing, motion control, battery longevity, camera visibility, and the accuracy of OpenCV.
- **Multiple HydraBots:** Multiple robots were tested in the pool for the responsiveness of the consensus algorithm.

External Documentation:

- Main Video: <u>https://youtu.be/bQWFXUzdTtw</u>
- Swarming Simulation: <u>https://youtu.be/wZkDHHQDu_g</u>

• Swarming Implementation: <u>https://youtu.be/XnniQ1b79gc</u>

7. Marketing and Cost Analysis

7.1 <u>Marketing Analysis</u>

The target customers for AUVs that perform search functionalities are researchers, the military, and disaster relief organizations. For example, the National Oceanic and Atmospheric Administration (NOAA) uses submersibles, such as the Hercules Remotely Operated Vehicle (ROV) to explore the ocean [6]. The military and disaster relief organizations often have common goals for their AUVs, namely search and rescue missions. The U.S. Navy is currently using the torpedo-sized Bluefin 12D AUV and three OceanServer Iver3 580 AUVs to search for the sunken Argentine Armada submarine *San Juan* [7]. General Dynamics created the Bluefin 12D AUV and it was previously used in the search for Malaysia Airlines Flight 370. The Bluefin 12D is 14.2 feet long and 574 pounds on dry land, explaining its use for military operations as opposed to commercially-available swarm operations [8].

While significantly smaller than the Bluefin 12D AUV, the OceanServer Iver3 580 AUV still ranges between 60 and 85 inches with a weight between 59 and 85 pounds, both measurements significantly higher compared to the Hydra AUVs' dimensions. In addition, a single Iver3 AUV costs \$119,250.00 [9]. The main advantages Hydra has over its competitors is its swarming capability, reduced size and weight, and comparably minimal cost. Hydra's swarming capabilities allow it to gain greater coverage of an area in a shorter amount of time. In addition, its decreased size and weight makes it more accessible to all users, as opposed to militaries that can transport AUVs weighing hundreds of pounds into the middle of the ocean. Finally, Hydra's economical cost of \$10,000.00 makes it affordable to researchers and universities interested in swarming AUVs, not just militaries and well-funded disaster relief organizations.

7.2 Cost Analysis

According to Table 5, the parts for developing six AUVs for prototyping the swarming capabilities of Hydra will cost approximately \$800.66. The most expensive part chosen is the inertial measurement unit (IMU) at \$14.95. Currently, this team is planning to build the AUVs with parts similar to those used for the ULR. Furthermore, the cost to create all prototypes remains low because the AUV chassis can be printed at Georgia Tech for free using 3D printers.

Product Description	Quantity	Unit Cost	Total Cost (6 AUVs)
Raspberry Pi Zero [10]	6	\$ 5.00	\$ 30.00
Arduino Pro Mini [11]	6	\$ 9.95	\$ 59.70
Turnigy 1600mAh 2S 20C Lipo Pack			
Battery [12]	6	\$ 7.64	\$ 45.84
Switched Mode Regulator [13]	6	\$ 4.30	\$ 25.80
DC Motor [14]	18	\$ 7.29	\$ 131.22
RGB LED [15]	6	\$ 1.95	\$ 11.70
TOF Sensor (Senior Design Lab)	36	\$ -	\$ -
Chassis (3D Printed)	6	\$ -	\$ -
Propellers [16]	18	\$ -	\$ -
IMU [17]	6	\$ 14.95	\$ 89.70
Rain Sensor [18]	30	\$ 7.89	\$ 236.70
Miscellaneous (Cables, Wiring, Epoxy, Etc.)		\$ 50.00	\$ 50.00
Packaging		\$ 30.00	\$ 30.00
Total Parts	•	•	\$ 710.66

 Table 7. Parts Costs

Table 6 displays the approximate development costs for Swimming Swarm, assuming each engineer is paid \$40 per hour. Group meetings require the most hours of labor because all team members must meet frequently to update each other on their independent progress to maintain a clear understanding of progress and required tasks.

	Total Hours of	
Project Component	Labor	Labor Cost
Chassis Development		
Chassis Design	20	\$ 800.00
Chassis Construction	10	\$ 400.00
Hardware Development		
Sensor Testing	30	\$ 1,200.00
Microcontroller		
Configuration	40	\$ 1,600.00
System Integration	40	\$ 1,600.00
Software Development		
Algorithm Programming	200	\$ 8,000.00
Algorithm Simulation		
Testing	100	\$ 4,000.00
Algorithm Robotarium		
Testing	50	\$ 2,000.00
Algorithm AUV Testing	100	\$ 4,000.00
Demo Preparation	100	\$ 4,000.00
Group Meetings	300	\$ 12,000.00
Total Labor	990	\$ 39,600.00

 Table 8. Development Costs

Table 7 provides the total development cost given that the fringe benefit is 30% of the total

labor cost and overhead is 120% of the cost of parts, labor, and the fringe benefit combined.

1	
Parts	\$ 710.66
Labor	\$ 39,600.00
Fringe Benefits (30% of Labor)	\$ 11,880.00
Subtotal	\$ 52,190.66
Overhead (120% of Parts, Labor, and	
Fringe)	\$ 62,628.79
Total	\$ 114,819.45

 Table 9. Total Development Costs

Considering the team sells 500 units over 5 years, where a unit consists of six AUVs, the price per unit will be \$10,000.00, as shown in Table 8. Since the team will no longer have access to free 3D printing upon graduation from Georgia Tech, a plastic mold will be needed to develop new HydraBots. An outside company like Rex Plastics charges approximately \$12,000.00 for a plastic mold [19]. This does not include the price of manufacturing each chassis. The prices of individual parts, such as sensors and motors, will decrease when bought in bulk. With all of these development cost changes considered, the price to produce a unit will be approximately \$600.00. Technicians will assemble and test each AUV at a rate of \$20.00 per hour. Sales expense, consisting of the price of advertising, will be 3% of the selling price, which is \$10,000.00. The amortized development cost is the total development cost (\$115,017.45) divided by the expected number of units produced (500). This cost is already added to the selling price of each unit. Selling each unit at \$10,000.00 results in an expected revenue of \$5,000,000.00 over five years. With a profit of \$8,278.80 per unit sold, the percent profit is 480.99%.

Selling Price	\$ 10,000.00
Profit	\$ 8,278.80
Subtotal, All Costs	\$ 1,721.20
Amortized Development Costs	\$ 100.00
Sales Expense	\$ 300.00
Subtotal, Input Costs	\$ 1,421.20
Fringe)	\$ 775.20
Overhead (120% of Parts, Labor, and	
Subtotal	\$ 646.00
Fringe Benefits (30% of Labor)	\$ 6.00
Total Labor	\$ 20.00
Testing Labor	\$ 10.00
Assembly Labor	\$ 10.00
Parts Cost	\$ 600.00

Table 10. Selling Price and Profit Per Unit

8. Conclusion

Currently, the Hydra team has three functional HydraBot surrogate vehicles. These robots are able to perform the consensus control algorithm in a pool using color following, where the red robot is programmed to follow the yellow robot, the yellow robot is programmed to follow the orange robot, and the orange robot is programmed to follow the red robot. With more time, this team would perfect the robot's ability to communicate using IR. Each robot is equipped with the transmitters and receivers necessary to communicate using IR, but they proved unreliable in the field due to Arduino programming issues. The goal is to make each robot communicate with its neighbor using color detection as they currently do. Upon finding its neighbor, each robot would stop spinning and send an IR signal saying its ready to move forward to complete the consensus algorithm. Once each robot has become aware that all other robots are calibrated through IR communication, the robots would move together using color-following as they currently do and complete the consensus algorithm.

This team was unable to move forward with IR communication due to the complexity of Arduino-Raspberry Pi serial communication as well as the placement of the sensors on the robot. Due to the configuration of the surrogate HydraBot, communication capability was not as ideal as expected. Additionally, not all members of the team fully understood the Arduino code that made IR possible. When bugs arose, the other team members were unable to help, causing a standstill in progress. In the future, this team would allocate a larger number of its members to addressing IR communication, and even seek outside assistance from other resources. Another issue this team faced was adjusting each surrogate vehicle's motor controls due to the non-uniformity of their designs. This will hopefully be mitigated by using the Mechanical Engineering Team's AUV in the future. If each vehicle is then uniform, the Python and Arduino codes for each robot may need to be modified to all be uniform. To extend this work to the next level, a team will first need to implement IR communication between neighbor robots to get true swarming behavior. Then, they should move all their hardware into the vehicles made by the Mechanical Engineering Team to reduce the number of hardware failures in future testing. Once the consensus algorithms are performing satisfactorily in these vehicles, the team should add distance-measuring capabilities to each robot using the purchased lasers. Using OpenCV, these lasers can determine the distance between themselves and the objects they are pointing at. This can be used within the robots to determine their distances from their neighbors, eventually performing the formation control algorithm where the robots make a desire shape. With three robots, this would be an equilateral triangle.

9. References

- S. Gallagher, "US Navy sends underwater robots to assist in search for Argentine sub
 [Updated]," ars TECHNICA, para. 3, Nov. 20, 2017. [Online], Available:
 https://arstechnica.com/information-technology/2017/11/us-navy-sends-underwater-robots-to-assist- in-search-for-argentine-sub/. [Accessed Nov. 29, 2017].
- [2] Z. Stone, "These Six Luxury Underwater Drones Are Disrupting Fishing and Fun," Forbes, para. 4, June 15, 2017. [Online], Available: https://www.forbes.com/sites/zarastone/2017/06/15/these-six-luxury-underwater-drones-are-disrupting-fishing-and-fun/#6058f511d978. [Accessed Dec. 1, 2017].
- [3] M. A. Joordens and M. Jamshidi, "Underwater Swarm Robotics Consensus Control," Deakin University, Waurn Ponds, Australia, Tech. Report. 978-1-4244-2794-9/09, 6 Aug. 2014.
- [4] T. Schmickl *et al.*, "CoCoRo -- The Self-Aware Underwater Swarm," 2011 Fifth IEEE Conference on Self-Adaptive and Self-Organizing Systems Workshops, Ann Arbor, MI, 2011, pp. 120-126.
- [5] A. Jackson, "Underwater Learning Robot,"ece4012y2017.ece.gatech.edu, para. 2, April 14, 2017. [Online]. Available: http://ece4012y2017.ece.gatech.edu/fall/sd17f25/overview.html.
 [Accessed Dec. 1, 2017].
- [6] J. Newman, "Hercules (ROV) and Friends," Ocean Explorer, Jan. 21, 2014. [Online],
 Available: http://oceanexplorer.noaa.gov/technology/subs/hercules/hercules.html. [Accessed Dec. 1, 2017].

- [7] S. Gallagher, "US Navy sends underwater robots to assist in search for Argentine sub [Updated]," ars TECHNICA, Nov. 20, 2017. [Online], Available:
- [8] Ocean Server, "Bluefin-12D," Bluefin-12D datasheet, March 2016.
- [9] Bluefin Robotics, "Affordable Work Class Autonomous Underwater Vehicle (AUV)," Iver3 datasheet, April 2010.
- [10] Adafruit. Raspberry Pi Zero Version 1.3 [Online]. Available: https://www.adafruit.com/product/2885. [Accessed Nov. 30, 2017].
- [11] Adafruit. Arduino Pro Mini 328 5V/16 MHz [Online]. Available: https://www.adafruit.com/product/2378. [Accessed Nov. 30, 2017].
- [12] HobbyKing. Turnigy 1600mAh 2S 20C Losi Mini SCT Pack (Part LOSB1212) [Online].
 Available: https://hobbyking.com/en_us/turnigy-1600mah-2s-20c-losi-mini-sct-pack-part-losb1212.html. [Accessed Nov. 29, 2017].
- [13] DigiKey. Linear Regulator Replacement DC DC Converter 1 Output 5V 1.5A 7V 36V Input
 [Online]. Available: https://www.digikey.com/product-detail/en/OKI-78SR-5/1.5-W36-C/811-2196-5-ND/2259781. [Accessed Nov. 29, 2017].
- [14] PiBorg. DC Motor, Gearbox, Wheel and Tyre [Online] Available: https://www.sparkfun.com/products/105. [Accessed Nov. 30, 2017].
- [15] Sparkfun. LED RGB Clear Common Cathode [Online]. Available: https://www.adafruit.com/product/2378. [Accessed Nov. 30, 2017].
- [16] "Microheli Plastic 3 Blade Propeller 82mm Tail Blade (ORANGE) BLADE 230S/250CFX,"
 eBay. [Online]. Available: http://www.ebay.com/itm/Microheli-Plastic-3- Blade-Propeller-

82mm-Tail-Blade-ORANGE-BLADE-230S-250CFX-/262928078404. [Accessed Nov. 29, 2017].

- [17] GY-521 MPU-6050 Module 3 Axis gyro 3 Axis Accelerometer Module For Arduino," eBay.
 [Online]. Available: http://www.ebay.com/itm/GY-521-MPU-6050-Module-3-Axis-gyro- 3-Axis-Accelerometer-Module-For-Arduino /231214060723?hash=item35d56e94b3%3Ag%3AFfkAAOxy4YdTWoBQ. [Accessed Nov. 29, 2017].
- [18] Amazon. 5PCS Rain Water Level Sensor module Depth of Detection Liquid Surface Height Arduino [Online]. Available: https://www.amazon.com/dp/B01N058HS6/ref=asc_df_B01N058HS65286820/. [Accessed Nov. 30, 2017].
- [19] Rex Plastics, "How Much do Plastic Injection Molds Cost?," rexplastics.com, para. 4, July 15, 2013. [Online]. Available: https://rexplastics.com/plastic-injection-molds/how-much-do-plastic-injection-molds-cost. [Accessed Nov. 30, 2017].

Appendix A - Gantt Flowchart



Appendix B - Gantt Chart

ID	Task Name	Start	Finish	Duration	Jan 2018 Feb 2018 Mar 2018 Apr 2018 1/7 1/14 1/21 1/28 2/4 2/11 2/18 2/25 3/4 3/11 3/18 3/25 4/1 4/8 4/15 4/22
1	Select and Order Hardware Platform to Use	1/8/2018	1/15/2018	1w 1d	
2	Finalize/modify Chassis Design and Motor Control	1/18/2018	1/25/2018	1w 1d	
3	Test basic hardware platform	1/21/2018	1/28/2018	1w 1d	
4	Finalize Microcontroller to Use	1/8/2018	1/15/2018	1w 1d	
5	Select battery	1/8/2018	1/21/2018	2w	
6	Team review of microcontroller	1/21/2018	1/28/2018	1w 1d	
7	Select and Order Light tx/rx	1/8/2018	1/28/2018	3w	
8	Finalize communication protocol	1/21/2018	1/28/2018	1w 1d	
9	Software	1/8/2018	2/22/2018	6w 4d	$\overline{\nabla}$
10	Determine swarm algorithm to use	1/8/2018	1/28/2018	3w	
11	Simulate swarm algorithm behavior	1/28/2018	2/22/2018	3w 5d	
12	Create Control Flow	2/15/2018	2/28/2018	2w	
13	Integrate sensors on hardware platform	2/15/2018	2/22/2018	1w 1d	
14	Software/Hardware compatibility review	2/22/2018	2/28/2018	1w	
15	Integrate Software and Hardware	2/28/2018	3/4/2018	5d	
16	Design PID controller	2/1/2018	2/16/2018	2w 1d	
17	Identify PID gains	2/1/2018	2/7/2018	1w	
18	Test Single Robot	2/22/2018	3/4/2018	1w 4d	
19	Create Multiple Robots	3/4/2018	4/8/2018	5w 1d	\bigtriangledown
20	Create Second Robot	3/4/2018	3/15/2018	1w 5d	
21	Create Third+ Robots	3/25/2018	4/8/2018	2w 1d	
22	Test Multiple Robots	3/16/2018	4/22/2018	5w 3d	\bigtriangledown
23	Test communication between two robots	3/16/2018	3/25/2018	1w 3d	
24	Product testing on multiple robots	4/8/2018	4/22/2018	2w 1d	
25	Oral Presentation	1/8/2018	1/22/2018	2w 1d	
26	Review of Design Proposal	1/8/2018	1/22/2018	2w 1d	
27	Final Project Demonstration	4/22/2018	5/1/2018	1w 3d	
28	Participation in Design Expo	5/1/2018	5/1/2018	1d	

Appendix C - PERT Chart

Task	Most Likely Duration	Optimistic Duration	Pessimistic Duration	Expected Duration	Standard Deviation	SQR	STDEV OF TOTAL
						11.1111	7.09655
5	53	40	60	52	3.333333333	1111	6285
9	12	8	26	13.66666667	3	9	
						1.77777	
10	5	2	10	5.333333333	1.333333333	7778	
						17.3611	
11	35	20	45	34.16666667	4.166666667	1111	
						11.1111	
12	36	25	45	35.66666667	3.333333333	1111	
13	9	9	9	9	0	0	