Dancing Swarm of Robots

DESIGN DOCUMENT

TEAM 40
CLIENT: DR. AKHILESH TYAGI
ADVISORS: DR. AKHILESH TYAGI, DR. DIANE ROVER, DR. PHILLIP JONES
MEMBERS: ABDALLA ABDELRAHMAN, DANIEL NIKOLIC, BENJAMIN SCHNEIDER,
NOAH THOMPSON, MASON WALLS, COLE WEITZEL
SDMAY21-40@IASTATE.EDU
HTTPS://SDMAY21-40.SD.ECE.IASTATE.EDU/
REVISED: 11/15/2020

Executive Summary

Development Standards & Practices Used

WiFi and Bluetooth wireless connection standards are used for robot communication. Sensor data is transmitted via the I2C interface. Software development is conducted under clean, consistent practices with thorough documentation. Version control is managed by use of Github. This project follows an Agile development structure with the basic plan, develop, test, deliver, and assess steps. Progress organization will be handled via Git issues.

Summary of Requirements

- The robot swarm should model the triangular pattern of bird flocks.
- The swarm will be made up of two follower robots following a single leader.
- Only the lead robot may receive movement directions from a base computer station. There may be no communication between swarm members.
- Both followers must determine their movements from sensor data alone.
- Both followers must maintain a certain separation distance between the other two members within an approximate 10% tolerance.
- The swarm should be able to operate in a closed, controlled environment with enough empty, level floor space for all members to complete their movements unobstructed.
- Swarm members must be able to operate with ambient indoor light and electromagnetic noise.
- The total cost of the project should stay within the $500-$750 allotted budget.
- Approximately 250 to 450 person hours will be required for project completion.

Applicable Courses from Iowa State University Curriculum

Many of the classes that we have taken at ISU will be beneficial in completing our project. The first major course is computer engineering 288. Since everyone in our group is a computer engineering major, we have all taken this class. This class gave us the technical experience with the iCreate roombas that we will be using in our project. It also gave us insight into embedded systems and how to efficiently integrate our software into the existing hardware. The second applicable course that we all have taken is computer science 311. This class is important because it taught us about efficient algorithm design. It also taught us how to use different data structures and how to effectively implement them into an algorithm. The combination of these two classes will help us succeed in completing our project.

New Skills/Knowledge acquired that was not taught in courses

Use of the WeBots software suite was a completely new topic for this project. As this tool has not been taught in any of our courses, every group member came into this project with no prior experience. Additionally, managing a project as large as this with no in-person interaction was an entirely new concept. With the arrival of the COVID-19 pandemic last march, the first semester of this project was the first semester with fully-online course delivery for Senior Design.

Table of Contents

List of figures/tables/symbols/definitions

1 Introduction

1.1 ACKNOWLEDGEMENT

We would like to thank our advisors Dr. Akhilesh Tyagi, Dr. Diane Rover, and Dr. Phillip Jones, as well as Lee Harker for their assistance and guidance throughout this project. Additionally, we would like to thank the Department of Electrical and Computer Engineering for providing funding for this project.

1.2 PROBLEM AND PROJECT STATEMENT

General Problem Statement

Bird swarms are thought to move with a single leader that determines the direction of the flock. All other members simply maintain a certain local position relative to their neighbors. This presents several advantages including increased aerodynamic efficiency, increased scalability for large groups, and simplified navigation for the group. Other animal species such as fish and lobsters have been observed exhibiting similar behavior. This movement heuristic has several applications in the field of robotics including transportation, search-and-rescue operations, and space operations. The purpose of this project is to model the movement behavior of these swarms in two dimensions on the ground with a set of three robots.

General Solution Approach

This project will model a triangular bird swarm with a single leader robot and two followers. Only the lead robot will receive instructions for movement, both followers will have to compute their own movements without any communication between members of the swarm. The followers will determine their movements from sensor data alone, maintaining a certain distance between themselves and the other two swarm members.

To determine their local movements, followers will perform several distance readings over a small angular field of view of approximately 30 degrees. This field of view (FoV) is restricted to only the area around the angular position where the follower last found the leader. Restricting the angular FoV allows full-slew measurements to be performed much faster than a full 180-degree sweep.

Once the leader is found, the follower will then make corrections to its left and right wheel speeds to steer in the appropriate direction needed to maintain local positioning with the leader. By doing so, this project will produce a swarm system consisting of three physical robots that can perform coordinated swarm maneuvers within a controlled operating arena without any direct communication between participants.

1.3 OPERATIONAL ENVIRONMENT

The robot swarm will operate in an indoor space with enough empty floor space for the lead robot to execute its maneuvers and for the followers to properly follow the leader. There should be no physical obstructions within the operating area. The swarm should be able to operate with ambient indoor light and electromagnetic noise. The floor of the operating area should be flat and level with an even terrain. Excessive infrared interference such as motion detectors should be removed if possible.

1.4 REQUIREMENTS

- The robot swarm should model the triangular pattern of bird flocks.
- The swarm will be made up of two follower robots following a single leader.
- Only the lead robot may receive movement directions from a base computer station. There may be no communication between swarm members.
- Both followers must determine their movements from sensor data alone.
- Both followers must maintain a certain separation distance between the other two members within an approximate 10% tolerance.
- The swarm should be able to operate in a closed, controlled environment with enough empty, level floor space for all members to complete their movements unobstructed.
- Swarm members must be able to operate with ambient indoor light and electromagnetic noise.
- The total cost of the project should stay within the $500 allotted budget.
- Approximately 250 to 450 person hours will be required for project completion.

1.5 INTENDED USERS AND USES

The intended users of our final product are Iowa State University's Cpr E 288 professors. The intended use for this product is to help model and understand swarm robotic behavior and may be used to consider a redesign of the Cpr E 288 course. The results of this project would be incorporated into the course's lab in some capacity, either as a demonstration or student lab project. Once this design has been proven with a generalized algorithm, it can be more easily reproduced in a similar controlled environment such as the CprE 288 lab.

Considerable research has been done in the field of swarm robotics and its possible applications [1][2]. By modelling the group movement behaviors observed in birds, insects, fish, and even humans, groups of robots can coordinate together to perform complex coordinated tasks. Tasks done as a swarm tend to be much more efficient, scalable, and fault-tolerant than when done with a coordinating body governing the actions of individuals. Rather than being concerned with the group's movement as a whole, individual swarm members only have to maintain their local actions relative to those of their neighbors. Similar to these applications, this project will demonstrate group movement by individuals maintaining a local position.

1.6 ASSUMPTIONS AND LIMITATIONS

Assumptions

- The swarm will operate without any external EM interference.
- The operating arena will be free of physical obstructions that may impede the swarm.
- The swarm movement will be started with all robots already in formation and fully charged.
- The swarm will consist of three robots.
- All robots that need to be detected by other robots will have adequate reflective material to be properly sensed.

Limitations

- Like a bird swarm, the leader cannot stop instantaneously or turn in place.
- No external IR light sources may be present in the operating arena.

1.7 EXPECTED END PRODUCT AND DELIVERABLES

Follower Robot

The design for the follower robot consists of an iCreate Roomba vacuum, fitted with a Tiva C-Series microcontroller, a servo motor, and a mini LiDAR sensor. This robot also has an embedded program that is designed to follow the bot that is in front of it. There are two versions of the program, one for a bot to the left of its lead, and one for a bot to the right of its lead. The follower robots hold a scalable embedded program, so any number of bots can be added to the swarm.

Lead Robot

The design for the lead robot consists of an iCreate Roomba vacuum, fitted with a Tiva C-Series microcontroller, and Wi-Fi card. This robot also has an embedded program that is designed to allow a user to drive this robot in any direction, using Wi-Fi communication. Because the follower bots are scalable, only one lead bot is necessary for a swarm.

These products will be delivered by May 1st, 2021.

2 Project Plan

2.1 TASK DECOMPOSITION

1. Planning
    a. Platform Evaluation - A motion platform must be chosen to serve as the motion base and main logical hub for each of the three robots. The iRobot Create platforms from CprE 288 may be used for this purpose.
    b. Sensor Evaluation - A shortlist of sensor setups for the two follower robots must be compiled for later testing. This sensor setup will be the follower's only communication with the outside world.
2. Simulated Design Testing - The entire design will first be prototyped in a simulated environment with the WeBots software suite.
    a. Algorithm Design - A generalized algorithm heuristic must be devised for the two followers to execute to maintain their distance and the shape of the swarm.
    b. Simulated Platform Movement - The base platform of a swarm member will be modelled in the simulated environment and basic movement controls will be implemented through a WeBots node controller. A platform shortlist must be started before this task can be completed.
    c. Simulated LiDAR Testing - A LiDAR sensor node will be implemented and tested with a controller to enable distance data to be read from the simulation node. A sensor shortlist is required to complete this task.
    d. Integrated Object Tracking - The LiDAR sensor will be integrated with the platform node and a basic tracking algorithm will be implemented to make the simulated bot follow a moving object. Both the simulated platform and LiDAR sensor must be complete to start this task.
3. Simulated Prototyping - Two robots from the simulated design will be integrated together with a simulated leader to model a virtual robot swarm.
    a. Movement Algorithm Testing - The previously designed algorithm will be implemented with a single robot following a moving target at a fixed distance as it changes direction.
    b. Combined Interaction Testing - Two LiDAR sensors will be modelled in play at once to test the effects of possible IR interference between the robots. This requires the capability to model the sensors alone.
    c. 3-Participant Movement Testing - The entire swarm will be modelled with two follower robots implementing the tested algorithm following a target leader. All previous tasks must be finished before this task can be completed.
4. Physical Design Porting - The simulated design will be replicated on physical hardware. Taskset 2 must be completed before this taskset can begin.
    a. Physical Platform Construction - The sensors of each follower will be mounted on their according movement platforms and any needed IR reflectors will be mounted.
    b. Basic Platform Movement Integration - The simulated node controller from task 2.b. will be replicated on the physical platform and tuned as needed.
    c. Sensor Software Integration - The follower robot sensors will be integrated with the platform such that each follower is able to read distance data from the mounted sensor.
5. Physical Testing - The constructed swarm members will be integrated into a cohesive group, the swarm behavior will be replicated on hardware, and requirements qualification will be completed.

a. Basic Movement Testing - Basic movement and sensor integration will be implemented on single robots. The lead robot will be made to follow movement instructions sent from a main controller. Task 4.b. must be completed before starting this task.
b. Object Tracking Testing - A follower will be made to track a moving target via the mounted LiDAR sensor. Task 3.a. and all of taskset 4 must be completed before this point.
c. Movement Algorithm Testing - The moving target from task 5.b. will be followed at a fixed distance as the target changes direction, following the designed movement algorithm.
d. 3-Participant Formation Movement Testing - All three participants will be tested together to perform the swarm movement behavior with both follower robots following the leader. Tasks 3.c., 5.a., and 5.b. must be completed before this point.
e. Final Qualification - The full system will be formally qualified to ensure that all requirements are met. All previous tasks for the project must be finished to complete this task.

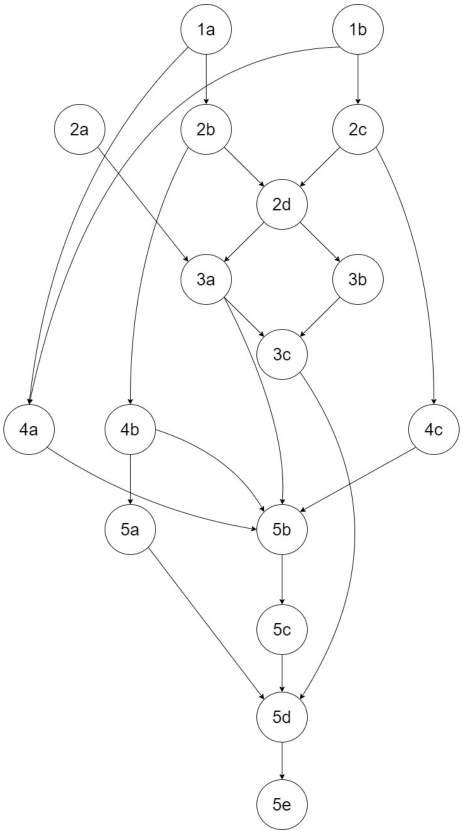

Figure 1 - Subtask Dependency Graph

2.2 RISKS AND RISK MANAGEMENT/MITIGATION

- LiDAR Mini Sensor
    - The sensor requires a 5V power rail, Tx and Rx connections, and a ground connection.
    - The CyBot may not be able to power the 5V power rail.
    - Risk probability: .2

- Algorithm Efficiency
    - An algorithm is required to process sensor data, and decide how to move the bots.
    - The algorithm may not be efficient enough to calculate a follower bot''s next move, causing it to get further and further away from the leader.
    - Risk probability: .4

- Time Constraint
    - With the accelerated Fall and Spring Semesters, the team will have less time to develop the project.
    - The team may not be able address every issue that we encounter, and may have to prioritize some errors over others.
    - Risk Probability:.3

2.3 PROJECT PROPOSED MILESTONES, METRICS, AND EVALUATION CRITERIA

- The simulated prototype can follow a moving target with a following distance within 10% of the goal.
    - The prototype can identify a moving target
    - The prototype can track a targets motion using sensors and servos
    - The prototype can follow the moving target within an acceptable distance
- The simulated swarm can operate with members drifting less than 20% off of their designated separation distances.
    - Members of the swarm can observe distances between one another
    - Members of swarm can identify when distances between one another fall into certain ranges and act accordingly
    - Members of the swarm can move away from one another when drifting too close
- The physical prototype will follow a moving target with less than 15% deviation from the prescribed following distance.
    - Members of the swarm can observe distances between follower and leader
    - Members of swarm can identify when distances between follower and leader fall into certain ranges and act accordingly
    - Members of the swarm can adjust speeds accordingly to follow a leader at an acceptable distance
    -
- The physical swarm can reliably move in formation without falling more than 7 inches out of place.
    - The swarm members can distinguish other members between environment objects
    - The swarm members can observe distances between one another
    - The swarm members can adjust to fit formation based on angle and distance from one another

2.4 PROJECT TIMELINE/SCHEDULE

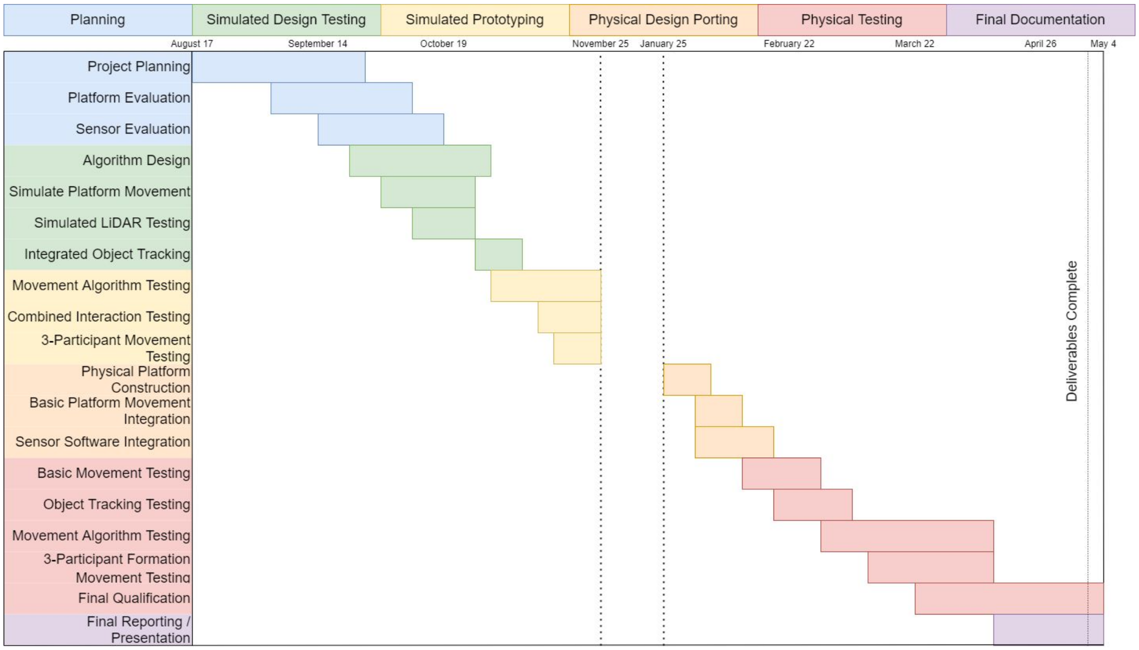

Figure 2 - Project Timeline

Major project planning was conducted from the beginning of the semester through the 29th of September. Primary simulation designs started September 21st, continuing through the 26th of October. During this time, we will be evaluating different platforms to build from, multiple sensor schemes, and our preliminary movement algorithm. After this stage, we will have determined what hardware will need to be ordered for the final project and the general design of the follower robots. Following primary design testing will be a simulated prototyping stage running from October 27th to the end of the semester on November 25th. This stage will finalize the simulated prototype design to the point where it can be replicated and implemented in the real world. The Spring 2021 semester will see this simulated design realized in hardware with primary design porting, baseline coding, and sim replication running through February 15th. Physical testing of the design, hardware, and movement algorithm will begin after this to produce the final deliverable. This is expected to run through around the first week of April. After that, the remaining three weeks will be spent finalizing the deliverable hardware, documentation, and reporting. All deliverables will be completed and delivered together by May 1st.

With this 6-phased incremental development roadmap, the project should safely be able to be completed in two semesters. By verifying the design in a simulated model before physical design, a large amount of potential issues can be headed-off early. Additionally, many tasks can be conducted simultaneously, allowing us to leverage our 6-person workforce.

2.5 PROJECT TRACKING PROCEDURES

Version control throughout the project will be managed through use of a GitHub repository. Project progress will be directly tracked through use of Git issues. For effective direct communication, a Discord server will be used.

2.6 PERSONNEL EFFORT REQUIREMENTS

Task	Personal Effort Time (Person-Hours)
Platform Evaluation	6 Hours
Sensor Evaluation	15 Hours
Algorithm Design	60 Hours
Simulated Platform Movement	30 Hours
Simulated LiDAR Testing	75 Hours
Integrated Object Tracking	150 Hours
Movement Algorithm Testing	15 Hours
Combined Interaction Testing	12 Hours
3-Participant Movement Testing	12 Hours
Physical Platform Construction	3 Hour
Basic Platform Movement Integration	3 Hour
Sensor Software Integration	15 Hours
Basic Movement Testing	6 Hours
Object Tracking Testing	15 Hours
Movement Algorithm Testing	15 Hours
3-Participant Formation Movement Testing	30 Hours
Total	462 Hours

Figure 3 - Personal Effort Requirements Timetable

The planning portions of this project are expected to take a total of 21 hours cumulatively. Beginning shortly before the conclusion of planning, initial simulated design work will start for a total of 165 person hours. Making up the bulk of the project, simulated prototyping will begin concurrently with simulated design as components are finished. This will take approximately 189 person hours. After the end of winter break, physical porting of the simulated prototype will start, requiring about 21 projected hours. As component groups are finished, final physical design testing will be conducted, taking up a final 66 hours. The remaining semester time will be spent on final documentation.

2.7 Other Resource Requirements

Three iRobot Create motion platforms will be required for each of the three swarm participants. Additionally, a suitable testing arena must be selected for use once physical testing begins. For simulated modelling, the WeBots software suite will be utilized.

2.8 Financial Requirements

A $500-$750 budget has been allocated to purchase the additional hardware required for this project such as sensors, additional controllers, or third-party motion platforms.

3 Design

3.1 PREVIOUS WORK AND LITERATURE

The most applicable course to this project is CprE 288 as the entire project is centered around the CyBot platform from that course. The embedded systems principles of this course are the backbone of this project's implementation. The work we have done already in CprE 288 will be used mostly in the implementation of the lead bot. As for the follower bots, we will need to implement new designs to allow the bots to communicate to each other using only the sensors on the bots. The combination of three cybots into a working design is something that was not done in CprE 288, and will require more work and testing to achieve the desired swarm behavior.

There has been considerable research conducted in the area of swarm robotics, but not in this project's application[1][2]. Most swarm experiments are centered around groups of flying drones such as quadcopters. These swarms usually navigate an area or perform flying maneuvers as a group, sometimes with a designated leader guiding the swarm's moves. Similarly to our project, though, followers simply move to maintain their own local separation between swarm members. The limitation to ground movement alone and to simple sensors is somewhat unique to this project.

3.2 DESIGN THINKING

The design for our robot swarm stems from the already implemented Cybot created by the CprE department. This design has been proven to work in single-bot designs, utilizing all of the equipped sensors. We chose this platform because we are familiar with how it works, and can easily modify it to fit our needs. Before making this decision, we researched other potential platforms such as the Sphero Rovr. We stayed with the Cybot platform because it already has working microcontrollers, custom mounting plates, and proven performance. All of this would have been missing on the Sphero Rovr, requiring much more time spent finding compatible microcontrollers, servors, and sensors. Apart from the platform. We have also decided to swap the attached infrared sensor from the cybot, and replace it with a LiDAR mini sensor. This new sensor allows for faster scanning speeds, and is more accurate than its counterpart. We considered a few different versions of a LiDAR sensor, including an iteration that would allow for 360° scanning. We decided to go with the LiDAR mini, as it would easily connect to the given Tiva microcontroller, and work easily with the pre-attached servo on the CyBot.

3.3 PROPOSED DESIGN

Our design centers around the iRobot Create platform for two main reasons. Firstly, ISU has several of the platforms available for the project and a considerable amount of documentation available concerning their use and expansion. This allows us to more easily fit within our required budget without limiting the movement or sensor capabilities of the participants. Secondly, our simulation software, WeBots, has a prebuilt node for the iRobot Create. This saves a considerable amount of time in modelling and improves the realism of the simulation, reducing the amount of restructuring needed in hardware reconstruction.

To allow follower robots to detect the leader's position, a Parallax Standard Servo is mounted on the CyBot with a sensor head attached to the servo arm assembly. This sensor head mounts a LiDAR distance sensor capable of detecting the distance between the sensor and the nearest reflecting object within the sensor's line of sight. Dimensions of the servo arm and mounting assemblies can be found in design schematics available from the ETG.

By sweeping the sensor through a portion of its 180-degree field of view (FoV), a picture of the robot's surroundings can be constructed. Using this setup, the follower robots are able to lock onto the leader, figure out the leader's position relative to themselves, and maneuver to maintain a certain separation between the moving leader and itself. This results in a swarm-like movement with followers simply focused on maintaining their local position as described in the figure below.

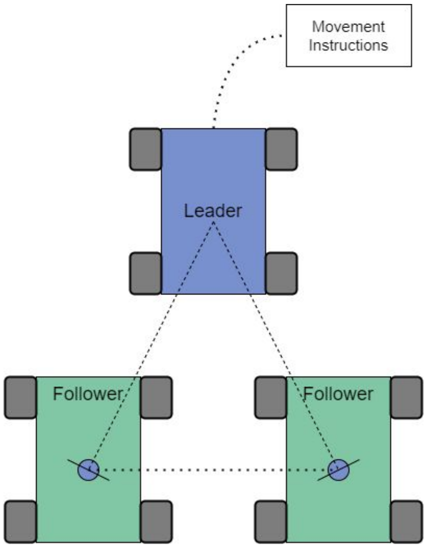

Figure 4 - Proposed Design Diagram

The determination of a follower's next move is a product of the robot's distance to the leader read by the LiDAR sensor and the relative angle between the follower's nose and the leader's detected position. If the follower reads the leader as moving too far left or right from the center of the follower's FoV, the FoV window will be slewed left or right to maintain a lock on the leader. The follower will then scale the left or right motor's speed along a linear scale relative to this angular rotation of the FoV window. Doing so will realign the follower's movement vector with that of the leader.

If the follower closes in on or falls behind the leader as determined by the LiDAR sensor's distance reading, the follower will scale both wheels' speeds relative to the difference between the desired separation length and the measured distance to the leader. This will speed up or slow down the follower until it is properly positioned with the leader.

The combination of the sensor head's differential steering and the distance sensor's speed control should allow a follower to maintain local positioning with the leader as it performs movement patterns. To allow followers adequate headroom to catch up to a maneuvering leader, the leader's maximum allowed speed should be around half to three-quarters that of the iRobot Create's top speed. This will ensure that followers are able to catch up to the leader if it begins a complex maneuver like a turn. Since the follower on the outside edge of the turn must move faster than the leader in order to maintain its local position, it must be ensured that the follower's movement wouldn't require a follower to exceed its maximum speed.

We elected to use WeBots for our modelling due to its high fidelity of sensor and physics modelling, expansion, and wide availability of documentation. The prebuilt iRobot Create node is used as a base for our simulated robot. From there, additional child nodes are added to model the CyBot's added acrylic plates, circuit boards, and extra hardware mounted on top of the base platform. Dimensional schematics of these added components are, again, available from the ETG.

Though we have had some difficulties with modelling all geometries correctly and ensuring that child nodes interact with their surroundings properly, WeBots is still the best tool for our uses. It has the best balance of expandability, simplicity of use, realism, and portability to hardware. When a refinement of the design has been completed in WeBots, reconstruction of this design in actual hardware will begin shortly after.

3.4 TECHNOLOGY CONSIDERATIONS

Our simulation software, WeBots, has some considerable simplifications in terms of environmental interactions with the modelled robots. Not all aspects of the operating arena such as signal noise, EMI interference, and hardware errors are modelled. These potential issues must be dealt with when the system is reproduced on actual hardware, as there isn't a way to effectively reproduce them in the simulator. Additionally, the controller code in the simulation is not identical to the code needed to implement the actual sensor hardware we plan to use, so this will need to be restructured in the next project phase as well. Despite these shortcomings, though, WeBots is still the most effective simulation tool that we can make use of, given our budget and time constraints.

Our other major tradeoff came in the selection of our LiDAR sensor and platform. The CyBot platform that we chose to use is a relatively basic platform with only some simple sensor options. However, it is equipped with a servo and sensor head mount that is adequate for our uses. It is also owned by ISU, so using it would result in large savings on our budget. For these reasons, we did not choose to purchase a more advanced robot platform. We also opted for a simpler LiDAR sensor rather than a higher-performance 360-degree module for similar reasons.

3.5 DESIGN ANALYSIS

Thus far, our proposed design seems to be effective for our purposes. One major limitation with the design having only a single distance sensor is the fact that follower robots cannot tell if or when the lead robot rotates in place. Additionally, if the leader abruptly changes its course at high speed, it is possible for the follower to lose its lock and become lost. However, both of these movement cases do not reflect the swarm movement of birds that we are trying to model with this project. After all, most bird swarms aren't capable of hovering and turning in place or quickly changing course without collapsing on themselves. Therefore, we chose to limit our scope to following leader movements that are more gradual in nature. This both alleviates some potential issues with the design and reflects a more realistic model.

3.6 DEVELOPMENT PROCESS

Our project has followed an adapted Agile development process. As the method has been proven in other projects and has been emphasized through the course lectures, we decided it would be the best fit. Incremental features are implemented in a sprint-like structure with time frames shifted for the pace of the project. As the sproatic required assignments for this course don't fit very well into the locked-down nature of an Agile sprint, some adjustments have been made along the way as needed.

3.7 DESIGN PLAN

We will initially start designing our project using various simulation softwares such as WeBots. We will attempt to incorporate the algorithms into the simulation in order to ensure that the followers will be able to follow the leader without receiving instructions. Within our design we will ensure that it meets the requirements that were highlighted in Section 1.4. Outside of the simulation environment, we will have to ensure that all sensors that we acquire are compatible with the iRoomba bots and that the iRoomba bots have no defects. We will do this by developing unit tests to ensure that the movements of the bots are that as planned. We will then move into implementing our algorithms into the actual bots rather than on the simulation software.

Next semester, we plan on using the labs in Coover to test and build our project design. This will be a lot more efficient rather than working remotely. This semester, working remotely has been acceptable because most of our work consists of writing documents and recording presentations, however during actual implementation it will be easier to have in person communication.

4 Testing

4.1 UNIT TESTING

For our project we will have two different controllers. The first controller will be for the leader robot, and the second controller will be for the two follower robots. Each of these controllers will be tested individually with unit testing to ensure that they are working as expected and controlling the robots correctly. We will also have a separate class that will be used to read the sensor data and another for the path planning and movement of the follower robots. We will be testing the basic movement of the iRobots to ensure that the robots are able to independently move forward/backward, turn left/right, detect an obstacle/cliff and maneuver the environment based off of the information collected from the sensors.

The leader robot controller will need testing for the movement and to make sure that it is following the specified path given by the user. This can be tested by providing a specific output, running the simulation, and observing that the leader robot moved to the correct location.

The follower robot controller will need unit testing done for the sensor data analysis and movement of the robot. We will also need to test the ability of the follower robots to move solely based off of sensor data. This can be tested by running the simulation, moving the robot, and observing the sensor data output and determining that it is detecting objects locations.

4.2 INTERFACE TESTING

WeBots allows for a 3D simulated environment where all the robots can be controlled at the same time. By using this 3D environment, we are able to have all three robots in formation and are able to test that the algorithms are working correctly. We will be able to give the leader robot a specific path or destination, and let the follower robots use the sensor data to move and follow the leader.

For objects within a single robot, WeBots renders the servo and distance sensor as child nodes of the main robot node hosting the controller. To test the interface between the robot controller and the robot's child nodes, unit tests have been conducted with each component. For the servo, test cases setting the motor position to specific positions have been made. We then verified that the motor completed these motions correctly. To test the distance sensor, objects were rendered at fixed distances away from the sensor head and readings were taken from the LiDAR model to verify its correct functionality.

4.3 ACCEPTANCE TESTING

During the testing phase, we will be able to video record our simulation while it is running. In order to ensure that the functional requirements have been met, we will ask the client for a specific test case or an example of the expected output, and then record the simulation and submit it to the client for approval. We will also create our own test cases that reflect the design requirements and present those to the client as well. In terms of the non-functional requirements, the WeBots simulation tool allows us to be able to 3D model the environment and control the robots through the user interface. Although, when/if we are able to port our project over to physical 288 CyBots, then if we have time we will create a user interface that will be able to control the lead robot in order to improve user experience.

4.4 RESULTS

- Functional WeBots Tests
    - Our functional tests of our WeBots simulated model have found several issues with our design's node structure that have since been fixed. It was discovered that all components that a controller needs to interact with must be created as children of the robot running the controller. Sibling or parent nodes will not be visible to the controller.
    - Tests
        - The robot's servo controls were tested by assigning specific angle positions to the servo motor lookup table, waiting for the robot to complete the operation, and verifying that the resulting angular position was correct. An angle table was then stepped through to ensure accurate transition between positions. Edge case assignments were also attempted to verify that the model did not exceed the physical servo's limited 180-degree Field of View (FoV).
        - The LiDAR sensor node configuration and controller functionalities were verified by keeping the robot stationary, placing objects directly in front of the robot at known distances, and verifying the correctness of the sensor readings. Through this process, we were able to correctly tune the lookup table steps of the servo to create a reliable function for performing distance measurements.
        - Combined functionality of the servo controls and LiDAR sensor was tested by building an obstacle course of boxes around the robot similar to the final project for the CprE 288 lab. The CyBot model was then set to navigate the course while sweeping the servo through its full FoV and taking measurements while in motion. If an object was detected, the robot would stop and turn in place. With some tuning of the max speed and servo slew rate, the robot successfully avoided large objects. This verified correct functionality of the modelled platform, servo, LiDAR sensor, and combined controller.

5 Implementation

Over the course of the fall semester, we have started implementing our project using the WeBots simulation. We took advantage of the pre-made roomba model that came in the WeBots platform, but we had to add some parts so that it would resemble the CPRE 288 roombas. We added the acrylic plate, the servo, and a lidar sensor. We needed to add these components so that when we start working with the physical hardware next semester, it will be easier to port the code to the physical platform.

We finished 3D modeling the robot toward the end of the fall semester. We also already started working on movement control of the lead robot. We have it set up that the arrow keys or the "wasd" keys will move the lead robot. We have basic forward and backward movement working, along with gradual turning and turning the robot in place.

Our plan for winter break is to get 3 robots modeled in the same environment. We also plan to start working on algorithm design. Over the course of the design period, several of us have tried to implement an initial version of the follower movement algorithm. However, due to obstacles that were encountered while developing the simulation, the algorithm was not working. While we were initially discouraged with this setback, we now have a good base on how to start implementing the algorithm correctly. Our goal is to get an initial version of the follower movement algorithm working during the winter break and the first few weeks of the spring semester. Next spring semester we can split up our team members and have half of us work on the physical hardware and have the other half continue working on the movement of the follower robots. At the beginning of next semester, depending on if we can use the physical robots or if we will continue to use the simulation, we will focus on algorithm development and getting the followers robots to be able to follow the leader.

For a more detailed list of our plan, visit section 2.4 of this document. Our plan for next semester is feasible because we already have the simulation set up and working. We are already working on an algorithm and we also can utilize our skills that we learned from CPRE 288 when we start working with the physical robots.

6 Closing Material

6.1 CONCLUSION

During the fall semester, we completed our preliminary modelling of the CyBot platform in WeBots. We selected the WeBots software suite to conduct our simulated prototyping because it offers the greatest level of realism without an excessive setup overhead cost. Additionally, the software is free with plenty of documentation available. Other similar simulation software platforms are either too simplified to be useful for our testing, have an excessive amount of setup required to model different designs, or don't allow for realistic controller development. Though the controllers used in WeBots are relatively simplified with sensors and actuators being modelled as lookup tables rather than actual memory-mapped components, it still allows us to test the system at a higher level. By use of this controller scheme, we are still able to design and test the underlying movement algorithm, albeit with a slightly more abstracted implementation.

Our simulated model includes the iRobot Create base, added acrylic plates, circuit boards, servo, and sensor head assembly. We chose to use ISU's Cybot design as a platform due to its simplicity and cost-effectiveness. Since the university already has hardware platforms available, using them would eliminate the cost of purchasing a new robot platform for each of the three individual swarm members. Additionally, the CyBot already has all of the three required components for a follower: a controllable movement platform, a distance sensor to measure separation from the leader, and a rotating sensor head assembly mounted on a programmable servo. The only required addition would be an optional improved LiDAR sensor for the two followers to more accurately and quickly perform distance measurements.

The simulated robot's LiDAR sensor and servo is controllable by the robot's controller code. This controller is written to resemble the final hardware code that will ultimately be generated. Currently, our controller is capable of sweeping the sensor head through the full 180-degree field of view (FoV), taking periodic distance measurements from the sensor assembly, and stopping the robot's movement when an object is closer than 25 cm. Over the upcoming winter break and into the beginning of the next semester, this rudimentary controller will be modified to perform scans within a limited but movable FoV window for faster scanning, adjust the robot's speed to maintain a set following distance from a moving target, use the FoV window's position to keep a lock on the target, and steer the robot's movement to keep the FoV window around a certain angular position.

Once simulated prototyping is complete with a fully-functional three-member swarm, the hardware implementation of this design will begin. The controller code from WeBots will be adapted to utilize the CyBot's memory-mapped servo and our added LiDAR sensor. With our prototyped algorithm, follower robots will be able to acquire and maintain a sensor lock on the leader and use the LiDAR sensor data and servo position to maneuver with the moving leader and maintain a relative local position with it. This would produce the swarm-like behavior that we are seeking to achieve with a physical robot system.

6.2 REFERENCES

[1] Tan, Ying, and Zhong-Yang Zheng. “Research Advance in Swarm Robotics.” Defence Technology, vol. 9, no. 1, 2013, pp. 18–39., doi:10.1016/j.dt.2013.03.001.

[2] Schranz, Melanie, et al. “Swarm Robotic Behaviors and Current Applications.” Frontiers in Robotics and AI, vol. 7, 2020, doi:10.3389/frobt.2020.00036.

6.3 APPENDICES