Dancing Roomba Swarm

Design Document

Team 02

Client: Tyagi Akhilesh

Adviser: Tyagi Akhilesh

Team Members: Marcella Anderson (Software Engineer/Reports),
Joshua Arment (Software Testing/Meeting Scribe),
Adam Brandt (Software Testing),
Greyson Jones (Hardware Testing),
Noah Kiel (Chief Software Engineer),
Devon Kooker (Task Coordinator/Debugging),
Hunter May (Client Interaction/Hardware Testing)

sdmay22-02@iastate.edu

https://sdmay22-02.sd.ece.iastate.edu/

Executive Summary

Development Standards and Practices Used

- IEEE 802.11 - Wireless Networking
    - Allows easy connection between devices
- IEEE 754 - Floating point arithmetic specifications
    - Floating point allows for more precise measurements
- IEEE 1588 - Precision Time Protocol
    - Synchronize clocks across roombas
- IEEE 1801 - Unified Power Format
    - Track power consumption of the Roomba, to maintain an acceptable charge life.

Summary of Requirements

- Roombas must be able to exhibit swarm-like behavior
- Follower Roombas must follow behind a lead Roomba at a 60 cm specified distance and angle within 10% error
- The follower Roombas should not receive any controls and should rely only on their own sensor data
- The leader Roomba will receive movement directions from a base computer
- Components must be able to be powered by Roomba battery
- Components purchased for the Roomba will cost no more than $500

Applicable Courses from Iowa State University Curriculum

- CprE 288 - Embedded Systems I: Introduction
- CprE 488 - Embedded Systems Design
- ComS 309 - Software Development Practices
- ComS 327 - Advanced Programming for C/C++
- Math 166 - Calculus II

New Skills/Knowledge Acquired that was not Taught in Courses

- Swarm Algorithms and their implementation
- Determining which parts/devices to purchase
- C.A.D/Autodesk design programs
- Digital to physical conversion
- Lidar programming experience

Table Of Contents

1 Team 6

1.1 Team Members 6

1.2 Required Skill Sets for Project 6

1.3 Skill Sets Covered by Team 6

1.4 Project Management Style 6

1.5 Initial Project Management Roles 6

2 Introduction 6

2.1 Problem Statement: 6

2.2 Requirements & Constraints: 6

2.2.1 Functional requirements: 6

2.2.2 Economical Requirements: 6

2.3 Engineering Standards: 7

2.4 Intended Users and Uses: 7

2.4.1 Users: 7

2.4.1 Use Cases: 7

3 Project Plan 7

3.1 Project Management/Tracking Procedures 7

3.2 Task Decomposition 7

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria 8

3.4 Project Timeline/Schedule 9

3.5 Risks And Risk Management/Mitigation 9

3.6 Personnel Effort Requirements 10

3.7 Other Resource Requirements 10

4 Design 10

4.1 Design Context 10

4.1.1 Broader Context 10

4.1.2 User Needs 11

4.1.3 Prior Work/Solutions 11

4.1.4 Technical Complexity 11
4.2 Design Exploration 12
4.2.1 Design Decisions 12
4.2.2 Ideation 12
4.2.3 Decision-Making and Trade-Off 12
4.3 Proposed Design 13
4.3.1 Design Visual and Description 13
4.3.2 Functionality 13
4.3.3 Areas of Concern and Development 14
4.4 Technology Considerations 14
4.5 Design Analysis 14
4.6 Design Plan 14
5 Testing 14
5.1 Unit Testing 14
5.2 Interface Testing 15
5.3 Integration Testing 15
5.4 System Testing 16
5.5 Regression Testing 16
5.6 Acceptance Testing 16
5.7 Results 16
6 Implementation 17
7 Professionalism 17
7.1 Areas of Responsibility 17
7.1.1 IEEE - Code of Ethics 18
7.1.2 NSPE - Code of Ethics 19
7.1.2.1 Rules of Practice 19
7.1.2.2 Professional Obligations 19
7.2 Project Specific Professional Responsibility Areas 20
7.3 Most Applicable Professional Responsibility Area 21

8 Closing Material 21

8.1 Discussion 21

8.2 Conclusion 21

8.3 References 21

8.4 Appendices 21

8.4.1 Team Contract 21

1 Team

1.1 Team Members

Marcella Anderson, Joshua Arment, Adam Brandt, Greyson Jones, Noah Kiel, Devon Kooker, Hunter May

1.2 Required Skill Sets for Project

- C programming experience
- Multiple Sensor Knowledge
- Software/Hardware Experience
- digital-physical conversion skills
- Low-level Networking

1.3 Skill Sets Covered by Team

- C programming experience - Hunter, Joshua, Devon, Noah, Adam, Greyson, Marcella
- Sensor Knowledge - Hunter, Greyson, Adam, Devon, Marcella
- Software/Hardware Experience - Hunter, Joshua, Devon, Noah, Adam Greyson, Marcella

1.4 Project Management Style

We are using the Scrum project management style. It requires a lot of conversation but has proven effective for our team.

1.5 Initial Project Management Roles

Joshua Arment - Meeting Scribe, Testing Lead

Hunter May - Client Interaction, Hardware Tester

Devon Kooker - Sensor Coordinator

Adam Brandt - Digital Conversion Overseer

Greyson Jones - Digital Conversion Overseer

Marcella Anderson - Report Manager

2 Introduction

2.1 Problem Statement:

Implement a design so that a collection of Roombas will follow a lead Roomba based on certain specifications.

2.2 Requirements & Constraints:

2.2.1 Functional requirements:

- Roombas must be able to exhibit swarm-like behavior
- Follower Roombas must follow behind a lead Roomba at a 60 cm
- specified distance and angle within 10% error
- The follower Roombas should not receive any controls and should rely only on their own sensor data
- The leader Roomba will receive movement directions from a base computer
- Components must be able to be powered by Roomba battery

2.2.2 Economical Requirements:

- Components purchased for the Roomba will cost no more than $500

2.3 Engineering Standards:

- IEEE 802.11 - Wireless Networking
    - Allows easy connection between devices
- IEEE 754 - Floating point arithmetic specifications
    - Floating point allows for more precise measurements
- IEEE 1588 - Precision Time Protocol
    - Synchronize clocks across roombas
- IEEE 1801 - Unified Power Format
    - Track power consumption of the Roomba, to maintain an acceptable charge life.

2.4 Intended Users and Uses:

2.4.1 Users:

- Iowa State University Computer Engineering 288 Students and Faculty
- Fire Rescue Teams
- Defense Systems
- Self Driving Cars

2.4.1 Use Cases:

- Create a swarm of n Roombas.
- Control the lead Roomba, and the swarm follows.
- Play music for the lead Roomba and it moves the swarm, making the swarm “dance”.

3 Project Plan

3.1 Project Management/Tracking Procedures

We will manage our project with KanBan which is an agile methodology management system that focuses on continuous small changes. KanBan works by visually organizing tasks in columns according to their stage in the development process. KanBan allows for the backlog of tasks to be constantly changing as well as provides openness about the progress of the project.

A combination of a KanBan board (probably on Trello) and various GitLab features will help track progress and manage tasks.

3.2 Task Decomposition

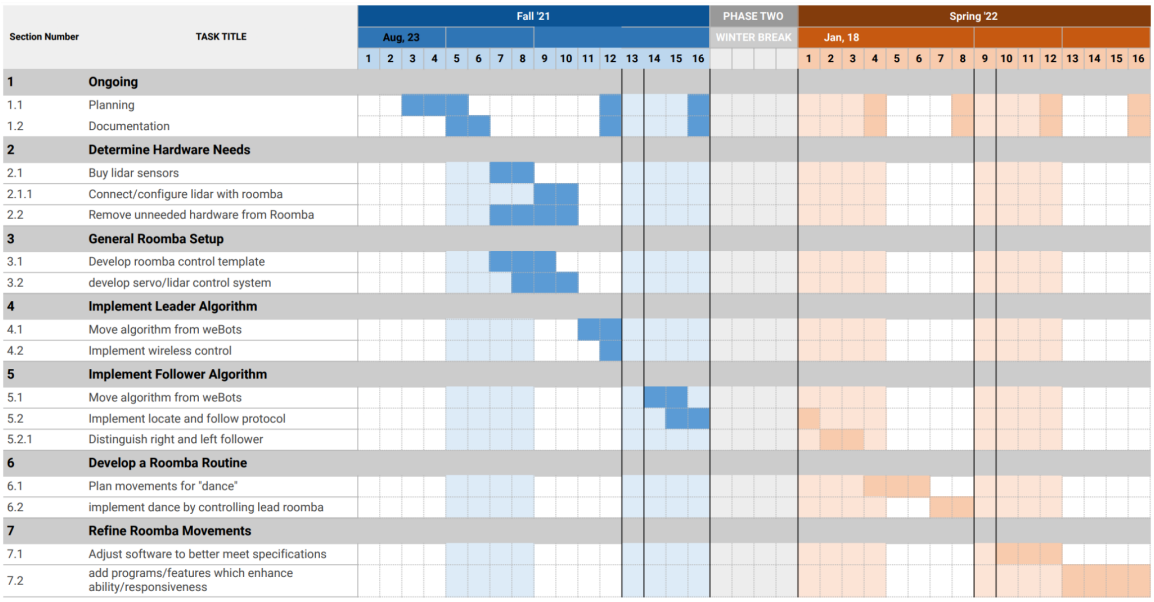
1. Ongoing - Documentation
    a. Weekly team meetings
    b. Project Documentation
2. Determine Hardware Needs
    a. Buy lidar sensors
    b. Connect lidar to Roomba
    c. Remove unneeded hardware from Roomba (IR and Sonic Sensors)
3. General Roomba Setup

    a. Develop template for general roomba control
    b. Develop control systems for lidar and servo
4. Implement leader robot algorithm
    a. Move algorithm from simulated weBots to classroom Roomba
    b. Implement wireless control of leader
5. Implement follower robot algorithm
    a. Move algorithm from simulated weBots to classroom Roomba
    b. Implement locate and follow protocol for roomba
    c. Setup system to distinguish between right and left follower
6. Develop a Routine for Roombas
    a. Plan movements for a “dance” that the Roombas follow
    b. Implement dance by only controlling the lead roomba
7. Refine Roomba Software and Movements
    a. Adjust software to better comply with specifications by client
    b. Add programs which enhance ability and responsiveness of Roombas

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

- Roomba control template is coded and configured with the servo and lidar sensor
    - Roomba can set servo position within 2 degrees of error
    - Lidar sensor can be read by Roomba and is accurate within 1 inch
    - Combo servo and lidar sensor can identify a post representing a roomba
- Lead roomba can be operated wirelessly
    - Lead Roomba will follow the preset dance routine
    - The lead roomba can be controlled by a user
- Followers will follow a leader with less than 15% deviation from the prescribed following distance.
    - Members of the swarm can observe the distance between them and their leader
    - Members of the swarm can observe the angle between them and their leader
    - Members of the swarm can adjust accordingly to follow a leader at an acceptable distance
    - Members of the swarm can adjust accordingly to follow a leader at an acceptable angle
- The swarm can reliably move in formation without falling more than 7 inches out of place.
    - A swarm member can distinguish other members between environment objects
    - A swarm member can observe distances between one another
    - A swarm member can adjust to fit formation based on angle and distance from one another
- Develop Roomba routine that uses sound/song as input to control the swarm
    - The swarm leader will listen/play/know the song and move accordingly
    - The swarm followers will follow the lead roomba without knowledge of the sound

3.4 Project Timeline/Schedule



The Gantt chart starts at the beginning of the 1st semester, but our work is scheduled to start on week 3, that is around the time we got our group assignments and started discussing the project. Determining Hardware Needs starts on week 7 of semester 1, this stage will help us gather the equipment needed as well as remove excess, after this stage we will have a roomba with all the equipment needed to complete the project. Also starting on week 7 of semester 1 is General Roomba Setup. This stage will configure the roombas with base code that will assist the algorithm implementation later on, this stage will also develop a control system for the servo and lidar to scan the area. The Implement Leader Algorithm stage involves moving the previously built algorithm from weBots to the physical roomba and setting up wireless control, this stage begins on week 11 of semester 1 and ends right before fall break. Implement Follower Algorithm starts right after fall break on week 14 and goes until after winter break to the 3rd week of semester 2. This stage will implement the follower algorithm from weBots along with protocols that allow them to join the swarm. After this task we will have a basic swarm built. On week 4 of semester 2 the Develop a Roomba Routine stage begins and will take until spring break to build and test a dance controlled by the lead roomba. The last stage starts after spring break. The Refine Roomba Movements stage works to improve the functionality of the Roomba to have more accurate control, as well as expand beyond that of the original definitions, allowing us to use client input to build outside the original scope.

3.5 Risks And Risk Management/Mitigation

- Determine Hardware Needs
    - Hardware incompatibility
    - The hardware we decided are needed are modeled from the weBots project, and there may not be a real world parallel for the part.
    - There are workarounds that may exist to still make the system work.
    - Risk probability: 0.2
- Implement follower robot algorithm
    - 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:0. 45

3.6 Personnel Effort Requirements

Hours	Task
60	Ongoing - Documentation
15	Determine Hardware Needs
30	General Roomba Setup
30	Implement Leader robot algorithm
40	Implement follower robot algorithm
60	Develop Roomba Routines
Remaining Time	Refine Roomba Software and Movements
235 +	Total:

3.7 Other Resource Requirements

- At least 3 roombas outfitted with the technology used in CPR E 288
- Wifi chips to operate these remotely
- The LiDAR sensors that were used in the simulation

4 Design

4.1 Design Context

4.1.1 Broader Context

Our project can help autonomous entities organize their movement and position to achieve a common goal. Many different areas of society have applicable applications that would benefit from the organization of multiple entities to accelerate or optimize a process.

Area	**Description**	**Examples**
Public health, safety, and welfare	Fire Rescue Drones	Increased ability to locate fire victims Carry/disperse fire retardants Can function if infrastructure is broken

Global, cultural, and social	National Defence Systems.	Could lead to different types of software in defensive drones
Environmental	Drones which can release fire-fighting chemicals	Increasing ability of drones used to tackle and/or prevent forest fires
Economic	Self Driving Cars	Ease of development of self driving cars could lead to lower costs

4.1.2 User Needs

Fire Rescue: Fire rescue needs a swarm which can work autonomously to search for people because internet and inter-device communication can not be counted on in fire rescue.

Defence Systems: Defence systems need swarms which can work autonomously because then communication between devices cannot be intercepted, interrupted, or interfered.

Self Driving Cars: Self driving vehicles need to work autonomously with each other and know each other's locations in order to prevent vehicle collisions.

4.1.3 Prior Work/Solutions

We inherited our project from a senior design group last year, they started this project by building a virtual simulation of the Robot flock. The previous group implemented a design to allow the bots to organize into a flock with each other using only the sensors on the bots. The simulation code contains most of the logic for our flock to work, it lacks the design to be easily modified to our physical application but is structured and documented well which will make it simple to rewrite for our design.

Previous work has been done on flocking in robotics. Prior research has gone as far as fixed wing flocking examples(Hauert, 2021)[1]. Some of these robots can communicate amongst themselves, but they do offer insight issues others have run into before when implementing flocking. There are also examples of algorithms which have been developed for flocking behaviour between autonomous robots(Virágh, 2021) [2]. This previous work has given the field a base of understanding which we can pull from in our project.

[1]Hauert, S. et al, 2021. Reynolds flocking in reality with fixed-wing robots: Communication range vs. maximum turning rate. [online] Ieeexplore.ieee.org. Available at: <https://ieeexplore.ieee.org/document/6095129>

[2]Virágh, C. et al, 2021. Flocking algorithm for autonomous flying robots. [online] Arxiv.org. Available at: <https://arxiv.org/ftp/arxiv/papers/1310/1310.3601.pdf>.

4.1.4 Technical Complexity

Our project includes different subsystems, including: LIDAR controls, Servo Control systems, Cliff and Edge detection systems, Wall/Bump Systems and potentially more, if we, as a group, decide that we need different equipment to complete our task. Implementation of these systems systems will be a challenge of its own, as there is no guarantee that they will interact well with each other.

After that we will test the logic of the previous team on the physical application, and evaluate if it will fulfill our needs. Our current plan is to design a robot template that can be implemented for any type of physical application allowing the algorithms we create to be reused for other platforms. We will start by moving their C code into C++, as an Object Oriented design for the template will create a better structure for our project.

4.2 Design Exploration

4.2.1 Design Decisions

So far we have made two major decisions: Convert the previous year's C code into C++ and Exchange the previous year's Direction LIDAR/servo with an Omnidirectional LIDAR. We are not completely sure on how the Dance implementation will work. Due to that we are holding off on making any large decisions regarding that aspect of the project.

4.2.2 Ideation

In regards to the Omnidirectional LIDAR, we went through a couple of different design options. Including:
1. Keeping the previous project's directional LIDAR

2. Using the on board IR and Sonic Sensors

3. WIFI strength triangulation

4. Cameras and vision processing

We chose the OmniDirectional LIDAR as it seems to provide the best aspects of most of the previous design considerations. As the LIDAR does both the Sonic and IR tasks in one sensor. And eliminates the need for a rotating servo that the directional LIDAR needs. The WIFI option would not be accurate enough for our needs, and the cameras would need more environment setup then the other options.

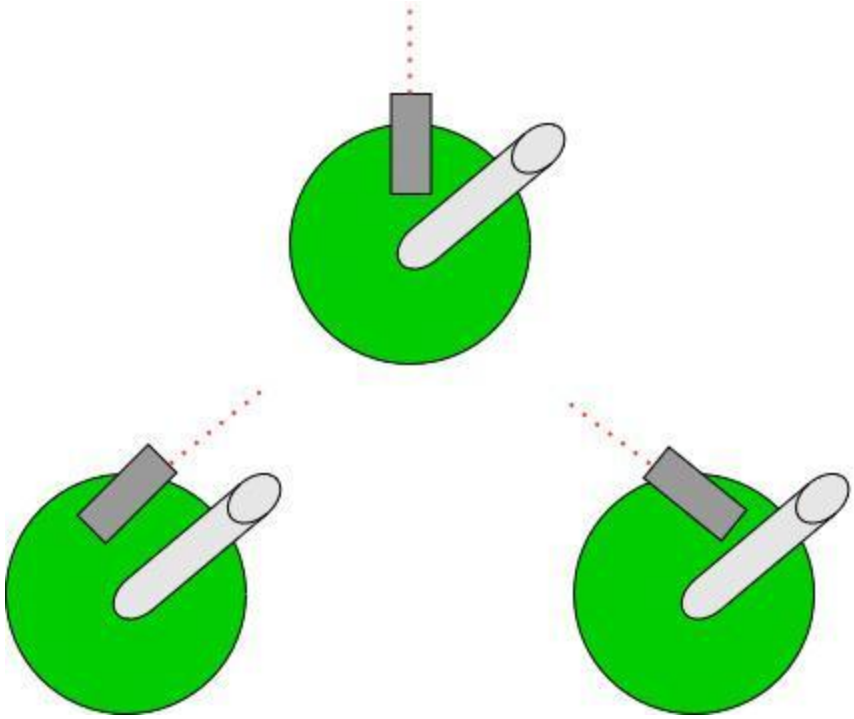
4.2.3 Decision-Making and Trade-Off

The WIFI and Camera with vision processing would not be viable due to the low accuracy those systems would provide. As well as the cameras would need extra environment setup adding to its complexity. Both types of LIDAR provide greater accuracy and scannable area compared to the Sonic and IR sensors which lack the technological advances of LIDAR. Finally we settled on the omni-directional LIDAR due to the scanning area. This will allow the sensor to easily track both the lead bot and obstacles without extra complexity on our part.

	Simplicity	Accuracy	Scan Area	Price	Totals
WEIGHT	1	3	3	2	
directional LIDAR	2	4	3	3	29
IR & Sonic	6	3	2	1	23
WiFi Tri	1	2	4	4	27
Camera & Vision	2	1.8	2.5	3	20.9
Omni-directional LIDAR	3	3.5	6	4	39.5

4.3 Proposed Design

4.3.1 Design Visual and Description



The diagram above shows the organization of our Roomba flock-swarm. The dark grey boxes represent our omni-directional LIDAR. To allow the LIDAR to locate other bots we will use long pvc pipes, represented as light grey cylinders above, as reliable markers for the sensors to identify. Since they will be the same size we can configure the robot to reliably identify that marker and track it for relative position.

4.3.2 Functionality

Determining a follower's next move will be a proportion of the follower's distance to the leader and the relative angle between the follower's heading direction and the leader's detected angular position. If the follower reads the leader as moving outside the desired angular position, the follower will alter its direction to maintain a lock on the leader, constantly modifying the left and right speeds to maintain distance and angle from the leader.

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 maneuver like a turn. Since the follower on the outside edge of the turn must travel a longer distance to maintain relative position it needs to move faster than the leader to do so. This helps ensure that the follower's movement wouldn't require it to exceed the maximum possible speed.

4.3.3 Areas of Concern and Development

The biggest area of concern is getting the LIDAR to accurately identify and measure the distance and direction of the leader. The inherited project used a direction LIDAR on a servo in which they controlled the angle of the sweeping scan following the leader, our proposed solution will use an omni-directional LIDAR which will look around in all directions to both follow the leader and help avoid obstacles. Since the previous project used a simpler LIDAR solution we will have to greatly modify their code to work with our more elaborate system. It is also new technology for any of our group members to use with a Roomba so it may take more time to properly connect. Early testing and a strong design will help us develop a solution that will be accurate and minimize the changes needed to the inherited code.

4.4 Technology Considerations

The main technical consideration for our project is the sensor we use to determine relative position. The provided hardware includes a directional sonar and IR sensor mounted to a servo to determine distance and direction. The advantages of this system are the already existing software support as well as the price, this system is very inexpensive in comparison to its counterparts, the weaknesses however include a limited view, the sensors limit the sensing distance and accuracy compared to more advanced(expensive) systems. The servo motor also limits the arc of sensing the robot can achieve, limiting the system to only sensing what is directly in front. We considered a Lidar sensor attached to a servo motor, this led to an increase in cost but with better accuracy and distance capabilities, but this still led to the disadvantages a servo comes with, a limited arc for sensing. One consideration we had for fixing the limited arc range was using a 360 degree servo but this would lead to more complicated engineering problems that would not be worth the cost to solve. So we found our final option and solution which is an omni-directional lidar sensor. This sensor, although with a high price tag comes the improved sensing distance of a lidar sensor along with allowing that sensing range to be in a complete 360 rather than limited to the arc possible by the servo motor. We decided on this sensor because for the robots to not communicate with each other it is vital they understand their surroundings.

4.5 Design Analysis

Our original design and thought was created by a previous senior design project. Though we were provided with an original design we decided to make a few alterations that will allow greater functionality as well as expected functionality but easier. Though we still have not completed the implementation of our design, we have all of the pieces and planning in place to hit the ground running during the second semester. More specifically we have the lidar sensor which is the piece of technology which required the most consideration.

While we are still getting some of the hardware put together, our designs have been checked by an external expert and will be able to begin the implementation once the second semester begins.

4.6 Design Plan

We plan on modifying the Roombas with a lidar sensor and creating a 3D model in autodesk to mount the lidar sensors on the Roomba. This will allow us to have the Roombas be able to follow the leader Roomba. We then plan on modifying the existing code base from the previous team's project to be able to efficiently have the Roombas follow the lead Roomba. In addition, we want to modify the Roomba to be able to play music for the lead Roomba and it moves the swarm, making the swarm "dance". We will then follow our testing guidelines to ensure that the Roombas work properly.

5 Testing

5.1 Unit Testing

Most of our unit testing will be done by testing the functionality of individual components of the roombas. For example to test the lidar we will have the lidar scan a distance we control and manually analyze its results.

- LIDAR
    - Manual setup of an object at a specific distance and checking if the LIDAR results match the actual measurement.
- Movement
    - Programming a movement routine testing and manually checking if the roomba does the expected routine. The routine will be moving forward and driving in a figure 8. It won't test reverse because the project requirements don't require reverse driving.
- Direction decisions
    - We will program a movement routine and manually check if the roomba does the expected routine.
- Receiving controls
    - We will send a movement routine over the air and manually check if the roomba does the expected routine.
- Following a leader.
    - We will program a movement routine for the leader and manual check in the follower follows.

5.2 Interface Testing

Things which will be tested during interface testing:

- Manual leader control
- Leader tracking with LIDAR
- Obstacle detection
    - With all sensor types
- Obstacle avoidance
    - As follower or leader

We will test interfaces by setting up situations for our roomba where these interfaces are tested. For example we will test object detection by setting up an environment with obstacles and test the robot's various systems for detecting the obstacles. We will test leader tracking by moving the leader manually while a

follower tries to follow directly, rather than the flock pattern. This will test the follower's locating ability without needing to combine multiple interfaces.

5.3 Integration Testing

Things which will be tested with integration testing:

- Integrating LIDAR with the rest of the robot
- LIDARs connection to movement direction (followers)
- Movement connection to controls received (leader)
- Obstacle detections connection to movement

To test interactions we will test multiple functionalities at the same time. For example we may have the roomba run into an obstacle and watch to make sure it stops. This would verify that the obstacle detection would be able to move the Roomba appropriately. Another example would be changing the scene around the roomba in a controlled way to affect the lidar readings and watching to verify the follower roomba still moves accordingly.

5.4 System Testing

Things which need to be included in system testing:

- Leader system testing
    - LIDAR detects obstacles
    - Robot follows desired path
    - Robot avoids obstacles
- follower system testing
    - LIDAR detects leader
    - LIDAR detects obstacles
    - Robot follows leader at the expected distance
    - Left and right robot follow the leader at the correct angle from the leader and each other
    - Robot avoids obstacles

Using the roombas there are two major systems to test the leader system and the follower system. Both systems require the robot's movement, sensor, and LIDAR systems to work as well as detect and avoid obstacles. These interfaces and specific subsystems allow both robot systems to have the basic functionality required of any robot that would belong in this swarm, to move and avoid obstacles. The leader system would require specific testing to verify any pre-programmed paths can be followed. The follower system would require additional tests that allow it to properly follow the leader. This includes the leader detection interface as well as the follow leader interface verifying distance and angle.

5.5 Regression Testing

We will ensure new things don't break old things after any major changes by putting the roomba on the ground and running it through various preconfigured tests to check if it moves and responds as expected. Checking various subsystems frequently will allow us to detect where in our implementation of the leader and follower robots we disrupt our systems if such an event occurs. The system we expect to build is very dependent on a variety of subsystems operating correctly and in unison. Updates to the system must be met with testing to ensure that all the dependent systems are kept in working order so new capabilities don't

disrupt the old or the tasks previously accomplishable. The main focus of our regression testing will be maintaining that the robot subsystems are kept in working order.

5.6 Acceptance Testing

Test against requirements or other client set constraints

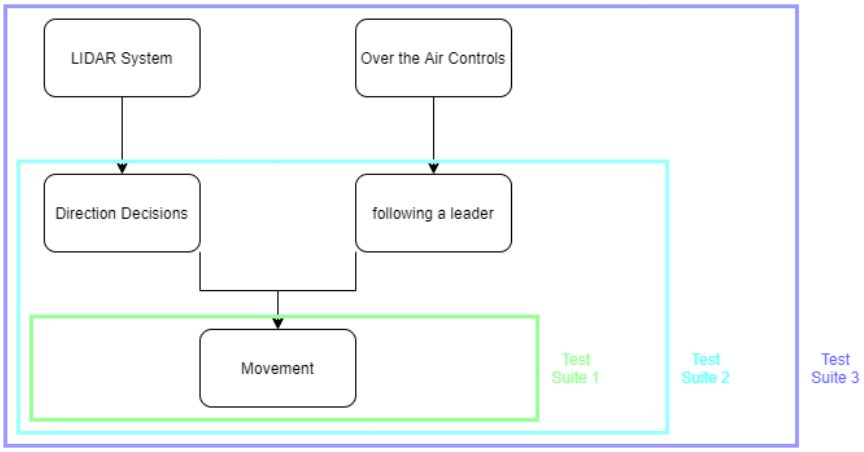
- Build tests to show simple desired flock movement
- Roombas follow the given movement routine
- Build advanced tests showing the flock avoiding obstacles

We will use a video to show the client, as well as to analyze position, angle, etc. We will use this video and the specifications of the movement routine to see if our project meets the criteria above.

5.7 Results

- Results will be in the form of visual verification by the team
- Results of tests will be recorded on video or documented with a written report
- Analyze video for robot meeting positional functional requirements

Our requirements talk about the movement of the roombas in relation to their environment and each other. If the roombas move in the way we expect them to in a testing environment then they are meeting the requirements. Therefore watching the roomba's movements will give us our results.



6 Implementation

This semester, we inherited our project and online code from the previous team. We spent several weeks learning and understanding that code in order to translate it onto the physical machines. From there, we did some in depth research about what specific sensors we wanted to use, and ordered everything. Additionally, we have looked into how the sensors will be mounted and are ready to mount the LiDAR and begin the coding of the movement algorithms at the start of next semester.

Our goal for the beginning of next semester is to finalize our design of the modified Roomba from the iCreate bots used in CprE 288. That being the removal of the Sonic, IR and servo on the Roomba. Then the addition of our 360 degree LIDAR sensor and a Pipe to the leader Roomba. After the Roomba is modified we will continue testing our code, debugging and implementing any changes that should arise from the testing.

For more information about this schedule, visit section 3.4 in this document. The progress that we have made this semester has set us up to continue following our schedule without any conflicts.

7 Professionalism

7.1 Areas of Responsibility

Chosen Code of Ethics: IEEE, with the full list below this table.

Professional Responsibilities	**How IEEE relates**	**Differences between IEEE and NSPE**
Work Competence	Code 2, Code 3, Code 4, Code 5, Code 6, Code 7	Code 6 and NSPE's Rules of Practice 2 both instill Engineers should only perform tasks they are educated and qualified to be competent.
Financial Responsibility	Code 4, Code 9	IEEE code 4 and NSPE's RoP 4 both hold Engineers accountable to act as trustful employees and not accept bribes or any other form of compensation.
Communication Honesty	Code 3, Code 7	IEEE's code 3 requires engineers to be honest about the information they're disclosing. NSPE, on the other hand, requires engineers to limit the information disclosed to only that which is not confidential and consented to by their superiors.
Health, Safety, Well-Being	Code 1, Code 9	Both related IEEE codes as well as the NSPE RoP 1 demand engineers hold safety above anything else.
Property Ownership	Code 1, Code 3, Code 5	The IEEE Code of Ethics focuses primarily on the safety and understanding of the material of their work. This differs from the NSPE RoP and Professional Obligations, as they focus on the materials used by the engineers being the sole property of the client.
Sustainability	Code 1, Code 6, Code 9, Code 10	Many of these IEEE Codes of

		Ethics touch on engineers using their abilities and communications with other engineers to promote sustainability. Parallelly, the NSPE Professional Obligations speak directly on engineers being encouraged to employ sustainability techniques.
Social Responsibility	Code 6, Code 7	The NSPE Professional Obligations #8 discusses that engineers should acknowledge their responsibilities for their actions specifically. This is different from the IEEE Code of Ethics #6 and #7, which speaks on engineers being honest with their abilities and accepting/giving criticisms.

7.1.1 IEEE - Code of Ethics

1. To accept responsibility in making decisions consistent with the safety, health, and welfare of the public, and to promptly disclose factors that might endanger the public or the environment.

2. To avoid real or perceived conflicts of interest whenever possible, and to disclose them to affected parties when they do exist.

3. To be honest and realistic in stating claims or estimates based on available data.

4. To reject bribery in all its forms.

5. To improve the understanding of technology; its appropriate application, and potential consequences.

6. To maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations.

7. To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to properly credit the contributions of others.

8. To treat fairly all persons and to not engage in acts of discrimination based on race, religion, gender, disability, age, national origin, sexual orientation, gender identity, or gender expression.

9. To avoid injuring others, their property, reputation, or employment by false or malicious action.

10. To assist colleagues and co-workers in their professional development and to support them in following this code of ethics.

7.1.2 NSPE - Code of Ethics

7.1.2.1 Rules of Practice

1. Engineers shall hold paramount the safety, health, and welfare of the public.
2. Engineers shall perform services only in the areas of their competence.
3. Engineers shall issue public statements only in an objective and truthful manner.
4. Engineers shall act for each employer or client as faithful agents or trustees.
5. Engineers shall avoid deceptive acts.

7.1.2.2 Professional Obligations

1. Engineers shall be guided in all their relations by the highest standards of honesty and integrity.
2. Engineers shall at all times strive to serve the public interest.
3. Engineers shall avoid all conduct or practice that deceives the public.
4. Engineers shall not disclose, without consent, confidential information concerning the business affairs or technical processes of any present or former client or employer, or public body on which they serve.
5. Engineers shall not be influenced in their professional duties by conflicting interests.
6. Engineers shall not attempt to obtain employment or advancement or professional engagements by untruthfully criticizing other engineers, or by other improper or questionable methods.
7. Engineers shall not attempt to injure, maliciously or falsely, directly or indirectly, the professional reputation, prospects, practice, or employment of other engineers. Engineers who believe others are guilty of unethical or illegal practice shall present such information to the proper authority for action.
8. Engineers shall accept personal responsibility for their professional activities, provided, however, that engineers may seek indemnification for services arising out of their practice for other than gross negligence, where the engineer's interests cannot otherwise be protected.
9. Engineers shall give credit for engineering work to those to whom credit is due, and will recognize the proprietary interests of others

7.2 Project Specific Professional Responsibility Areas

Work Competence

- Work Competence applies to our project, as high levels of precision and consistency with how the Roombas will communicate and move are necessary, as specified by our requirements.
- Our team has put a significant amount of effort into completing the individual sections of this project in a timely manner while maintaining a high level of performance. Additionally, there has been effort already placed into having the completed sections of code operate as consistently and precisely as possible.

Financial Responsibility

- Our group does need acceptable levels of financial responsibility, as we've been given a budget to purchase sensors that will be used in the development of the swarm/dance algorithms.
- As a team, we have a high level of performance in this responsibility area. We have properly researched and budgeted the LiDAR sensors that we'll need for this project.

Communication Honesty

- Open and honest communication between all members of our group, as well as the client, is essential to our project being successful. Without proper communication, there may be confusion between what our client wants and what we implement.
- The communication between our team internally and with the client is at a medium level of performance. While our communications so far have been acceptable, higher quantity and quality of communications may be necessary to avoid any confusion of the stakeholder(s).

Health, Safety, Well-Being

- The health, safety, and well-being of the stakeholder(s) must be accounted for in our project, and we need to avoid any hazardous consequences of pushing the hardware too far.
- A high level of performance has been achieved in this area, as we've properly researched what the roombas can handle and how much of an affect our sensors/attachments will have on them.

Property Ownership

- Property Ownership does apply to this project due to the nature of how our group "rents/borrows" the roombas from Iowa State University. We are responsible for the well being and maintenance of the bots, and are required to return them how we received them. We will also be working with the intellectual property of code we are writing and code which was written by the previous team.
- The performance of the team in this area is at a high level, as we are taking good care of the devices and services provided to us for this project.

Sustainability

- The Sustainability responsibility area applies to our project because this technology could be used in the future to further environmental efforts. Swarms could be used in drones in firefighting efforts for example.
- The rating of our team in this professional responsibility area is low because we have not spent much time considering the impacts of our project on the environment.

Social Responsibility

- This project does have social responsibility requirements, as the end goal of our project is to create a Swarm of robots that could benefit society by improving any industries that use autonomous systems.
- At this point in the project, our team has not had any impact in this area, meaning that the current level of performance is not assessed.

7.3 Most Applicable Professional Responsibility Area

Work Competence

This professional responsibility is incredibly important to our project, and means that we need to ensure a high level of consistency and precision from the algorithms written for the project to succeed. In this project, we've already begun working on our basic movement/leader algorithm with the leader roomba. The optimizations of the previous team's code allows us to begin using these algorithms with a preexisting degree of quality/integrity, and further optimizations from our team will bring these algorithms to their highest efficiency. The current functions/algorithms that we have are able to consistently move the roomba as ordered, but the requirements of precision and consistency are present specifically in the follower roombas.

8 Closing Material

8.1 Discussion

This semester we were not able to meet the requirements we set for ourselves, though we are on track to meet them next semester. There were some setbacks with our lidar sensor(it took longer to order and arrive than expected) but we expect to make up that time early next semester and possibly over break. We were able to spend this semester getting familiar with the Roombas and the provided Iowa State codebase, building a framework for us to use with the physical Roomba system as well as the algorithmic implementations of the previous group. This work will allow us to hit the ground running next semester and not be further delayed.

8.2 Conclusion

Our goal for this project was to create a swarm of roombas which perform a dance routine. To do this we are attaching a lidar to the robots, as well as a pole for detection. We are then attempting to use flocking algorithms to create a swarm with our robots which will perform a dance. During our first semester we were held back in our work by the lidar sensor. This roadblock was overcome and has paved the way for a successful second semester. In future iterations it is most important to have the sensors bought early on in the process.

8.3 References

Hauert, S. et al, 2021. Reynolds flocking in reality with fixed-wing robots: Communication range vs. maximum turning rate. [online] Ieeexplore.ieee.org. Available at: <https://ieeexplore.ieee.org/document/6095129>.

Virágh, C. et al, 2021. Flocking algorithm for autonomous flying robots. [online] Arxiv.org. Available at: <https://arxiv.org/ftp/arxiv/papers/1310/1310.3601.pdf>.

8.4 Appendices

8.4.1 Team Contract

Team Members:

1) _____Devon Kooker_____ 2) _____Joshua Arment_____
3) _____Adam Brandt_____ 4) _____Hunter May_____
5) _____Greyson Jones_____ 6) _____Marcella Anderson_____
7) _____Noah Kiel_____ 8) _____

Team Procedures

1. **Day, time, and location (face-to-face or virtual) for regular team meetings:**
    - Friday 1:00 pm - Meet with Dr. Tyagi via webex
    - Friday 1:30 pm - Meet with Xinyao li via webex
    - Open to scheduling meetings throughout the week via Discord for intergroup meetings.

2. **Preferred method of communication updates, reminders, issues, and scheduling (e.g., e-mail, phone, app, face-to-face):**
    - External communication: e-mail; Internal communication:
        i. A Discord server has been established, and we will regularly update it with reminders, announcements, etc.
    - Emails with our advisor and T.A. seems to be the preferred method due to COVID-19 restrictions.
3. **Decision-making policy (e.g., consensus, majority vote):**
    - Decisions will be a majority vote (with 7 members there should be no ties). Members are expected to "vote" within 48hrs, unless the decision needs to be made sooner.
4. **Procedures for record keeping (i.e., who will keep meeting minutes, how will minutes be shared/archived):**
    - Joshua Arment will write the meeting minutes and share/archive them via a folder in the team's shared Google Drive.

Participation Expectations

1. **Expected individual attendance, punctuality, and participation at all team meetings:**
    - If possible, all meetings should be attended. If anyone is unable to attend, they are to go read the minutes for the meeting in the shared Google Drive and inform the team if they have any input.
2. **Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:**
    - Everyone is expected to complete a fair share of the work. Due to COVID-19 and working remotely, we as a group understand that not everything can be split 7 ways, nor should everything be split 7 ways.
    - It is to be understood that if you are unable to work on something that was your responsibility, that you will communicate that to the team. So that the task can be completed on time.
3. **Expected level of communication with other team members:**
    - Team members are expected to communicate when they need help, and to do so as soon as they feel the need to do so. We are working together on this project, so others might know a solution to a problem that you do not.
    - We will have a weekly update, via discord, on how separate parts of the project are going, and see the direction that the project is going in.
4. **Expected level of commitment to team decisions and tasks:**
    - Team members are expected to approach any task with diligence and enthusiasm, whether or not they voted for that decision.
    - Concerns and considerations should be brought to the attention of the group if you are unsure of said decisions.

Leadership

1. **Leadership roles for each team member (e.g., team organization, client interaction, individual component design, testing, etc.):**
    - Marcella Anderson: Reports Manager, software engineer

- Joshua Arment: Software Testing, Meeting Scribe;
- Adam Brant: Software Testing
- Greyson Jones: Software/hardware testing
- Noah Kiel: Chief Software Engineer
- Devon Kooker: Task board coordinator, software debugging
- Hunter May: Client Interaction, Hardware testing

2. **Strategies for supporting and guiding the work of all team members:**
    - We as a team should be able to rely on each other for assistance, this is not to say that we will pass on work to other team members unnecessarily.
3. **Strategies for recognizing the contributions of all team members:**
    - Contributions can be noted in our weekly reports, and in the final report.
    - As well as documented through git activities including but not limited to commits, pushes, and merges.

Collaboration and Inclusion

1. **Describe the skills, expertise, and unique perspectives each team member brings to the team.**
    - Marcella Anderson - Writing in C, took CPRE 288, proficient in C development
    - Joshua Arment - Writing in C and effective test case development
    - Adam Brant - Took CPRE 288, proficient in C development
    - Greyson Jones - Developing Algorithms with higher level programming languages.
    - Noah Kiel - Worked with robot pathing algorithms before, PID controller/loops and proficient at C/C++
    - Devon Kooker - Took CPRE 288, proficient in C and C++ development
    - Hunter May - Adept with C programming and embedded systems.
2. **Strategies for encouraging and support contributions and ideas from all team members:**
    - Keep a board of all known issues , then discuss solutions to issues on board with all members taking all solutions into account for a final vote.
    - Any potential idea should be discussed and perceived as a better software/hardware implementation.
3. **Procedures for identifying and resolving collaboration or inclusion issues (e.g., how will a team member inform the team that the team environment is obstructing their opportunity or ability to contribute?)**
    - Team members should feel comfortable and confident with sharing with the group any issues that they are having. If any member does not feel so, then that is a problem in of itself and should be addressed.
    - If the group feels that it is necessary to do so, a Discord Bot (Voltaire, seems to be the go to for this) can be added to the server, so that messages can be sent anonymously to the group.

Goal-Setting, Planning, and Execution

1. **Team goals for this semester:**
    - Build a project timeline to mark milestones and help achieve timely goals.
    - Complete work consistently so as to not fall behind.

- Collaborate as much as possible on the project to minimize any potential confusion.

2. **Strategies for planning and assigning individual and team work:**
    - On a bi-monthly basis, we look at what needs to be done and create a 2 week sprint to complete the tasks.
3. **Strategies for keeping on task:**
    - Checking in weekly with team members outside of our regular meetings should help keep each team member focused.

Consequences for Not Adhering to Team Contract

1. **How will you handle infractions of any of the obligations of this team contract?**
    - Infractions will be discussed with the person responsible and solved amongst the team.
2. **What will your team do if the infractions continue?**
    - If the infractions continue, a TA or professor will be contacted.

****

a) *I participated in formulating the standards, roles, and procedures as stated in this contract.*
b) *I understand that I am obligated to abide by these terms and conditions.*
c) *I understand that if I do not abide by these terms and conditions, I will suffer the consequences as stated in this contract.*

1) Devon Kooker DATE 09/19/2021
2) Joshua Arment DATE 09/19/2021
3) Adam Brandt DATE 09/19/2021
4) Hunter May DATE 09/19/2021
5) Greyson Jones DATE 09/19/2021
6) Noah Kiel DATE 09/19/2021
7) Marcella Anderson DATE 09/19/2021
8) _____ DATE _____