BRADLEY
UNIVERSITY

Cooperative Control of Heterogeneous Mobile Robots Network

Gregory A. Bock, Brittany J. Dhall, Ryan T. Hendrickson, & Jared A. Lamkin
Dr. Jing Wang & Dr. In Soo Ahn
Department of Electrical and Computer Engineering

October 15th, 2015

EXECUTIVE SUMMARY

The goal of the cooperative control of heterogeneous mobile robots network project is to design, develop, and test control algorithms. With increasing interest in the study of autonomous vehicles, the Air Force Research Lab has proposed a project to research cooperative control of heterogeneous mobile robots network. Given the following proposal "Multiagent task coordination using a distributed optimization approach", a senior project at Bradley University was created. The network of different robots must complete a task autonomously using only local information. The various control algorithms that will be developed and tested are self-organization, formation stabilization, and self-steering. The solution must overcome limited communications and sensing, as well as system model uncertainties.

This project will aid in the development of cooperative based control solutions using mobile robots network. All robots within the network must communicate with other robots in the network by using local communication only. The limitations that robotic platforms must overcome are limited communication capabilities, limited sensing capabilities, and system uncertainties.

The defined areas that will be focused on during the project include object detection, object avoidance, self-organization, and self-steering. Areas that are out of scope for the proposed project include network security and negative emergent behaviors.

The objectives that the proposed project must consider are that the robots network should be cooperative, self-organizing, self-steering, reactive, and adaptive. The software used for the proposed project must be portable. This means that the software must be easily transferable to other robot platforms.

The project is funded by a grant from the Air Force Research Lab (agreement number: FA8780-13-0109), which is part of a larger project. The estimated cost of the cooperative control of heterogeneous mobile robots network is $23,610.00. The total estimated cost includes the robotic test platforms, software, chargers, and an additional 20% for unanticipated costs.

The proposal will introduce the constraints of the project, the scope of the project, the design approach, the method of solution, financial analysis, the timeline of the project, how the project will be divided, as well as this project could potentially impact the society and environment. The cooperative control of heterogeneous mobile robots network will introduce a new understanding of emergent behaviors in robotic networks.

ABSTRACT

The applications of cooperative control strategies for heterogonous mobile robots network are significant and far reaching. This technology would most likely be used by the United States Military for search and rescue, surveillance and reconnaissance, and drone strikes. The project, funded by the Air Force Research Lab, seeks to develop and design cooperative control methods that control a group of robots to perform different tasks autonomously. In order to achieve this goal, various control strategies will be investigated and tested using MATLAB and Simulink. Using Kilobots, E-pucks, and QBot 2s as robotic test platforms, algorithms will be designed on each individual robot. When the robot platforms exhibit their desired behaviors, they will be integrated together to create a heterogeneous network. The desired behaviors include self-organization, formation stabilization, and self-steering. Behaviors are based on the consensus of the robot group through local information exchange. Testing will be performed through an iterative cycle of mathematical validation, simulation, and hardware implementation. Major obstacles to the solution include communication and sensing limitations as well as system model uncertainties. Once developed, the technology should enhance military operations, improve efficiency, and save lives.

TABLE OF CONTENTS

I. INTRODUCTION

A. *Problem Background*

With the rapid development of embedded computing, communication, and sensing technology, recent years have seen ever increasing research interests in the study of unmanned vehicles. These vehicles often require a human to pilot them at another location. This situation makes it difficult for a single person to control multiple unmanned vehicles at once. Research in distributed cyber physical systems can provide information to further the use if unmanned vehicles. The two main topics (analysis of information flow and cooperative control) can be addressed to solve this problem.

B. *Problem Statement*

The development of cooperative based control solutions allows for the improvement of mobile robots network. Cooperative control strategies should permit for the best use of available information to produce positive emergent behaviors. The desired behaviors from the control methods are self-organization and self-steering. Implementation of cooperative control strategies should be performed on a group of heterogeneous mobile robots. Each robot agent must have the ability to communicate with any other robot agent within the network, but the highest level of communication must be performed locally, not globally. Individual robots will have communication, sensor, and movement limitations, which will produce uncertainties in the network. These uncertainties must be overcome for the network to exhibit optimal performance. To summarize, the goals of the project are as follows: the mobile robots network should exhibit self-organization and self-steering behaviors, the ability to detect objects and communicate among different robot platforms. Not only must the robots achieve these behaviors, they must also have algorithms that are easily portable to different robot platforms.

C. *Constraints*

A list of constraints for the mobile robot is shown in Table I. These constraints are the result of the different robotic test platforms being used to implement the cooperative control strategies. In order for the project to be successful, these constraints must be overcome.

TABLE I. A LIST OF CONSTRAINTS

Constraints
An agent of the network must have limited communication capabilities
An agent of the network must have limited sensing capabilities
The mobile robots network must overcome system uncertainties

When an agent has limited communication capabilities, all agents should communicate locally within the network (neighbor-to-neighbor). Each of these constraints means that information can be distributed throughout the network with minimum transfer of data. Due to the fact communication is done locally; the behaviors of the overall system emerge from agent-to-agent interactions. In order for the agents to perform the required tasks, limited sensor information should be distributed throughout the network to achieve consensus. The minimization of uncertainties is the aspect problem solving that will be addressed in the method of solution portion of this paper.

D. *Scope*

The project scope of the cooperative control of heterogeneous mobile robots network is to design and test distributed control algorithms using multiple robots. The test platforms that the control algorithms will mainly be programed on are the Kilobots and QBot 2s. During the project, sensing and communication sharing challenges among the robots will have to be addressed, as well as studying distributed control algorithms and collision avoidance strategies. There will have to be testing scenarios designed, such as formation control behavior and self-steering behaviors to ensure the success of the control algorithms.

TABLE II. SCOPE OF THE PROJECT

Scope of Project		
	In Scope	**Out of Scope**
Object Detection	x	
Object Avoidance	x	
Self-organization	x	
Self-steering	x	
Network Security		x
Negative Emergent Behaviors		x

As shown in Table II, the defined areas which will be focused on during this project are object detection, object avoidance, self-organization and self-steering. Object detection is the detection of objects by the robots network. Object avoidance is the avoidance of objects by robots network. Self-organization is the robotic agents autonomously organizing into a formation. Self-steering is the robotic agents autonomously moving towards a location while in formation. The areas that will not be focused on during the project are network security and negative emergent behaviors. Network security would include communication security/data encryption. Negative emergent behaviors include behaviors that are undesirable due to cooperative control.

II. STATEMENT OF WORK

A. *System Description*

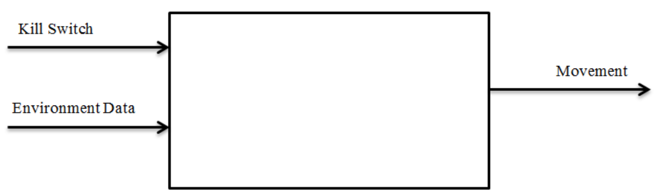

Fig. 1. Robots Network System Block Diagram

The system is considered to be the entire network of robotic agents, not a single agent. The system block diagram is shown in Fig. 1. For situations that cause unforeseen complications, there will be a kill switch for the network. The environment data will be considered any data gathered by an individual agent that is not a result of another agent. The output of the entire system into the environment is movement of the network. Movement can be the entire network of robotic agents, or sections of the network. Meaning the entire robot network does not have to move simultaneously.

There are two different state diagrams that are necessary for this project. The first state diagram models self-organization for the robots network. Appendix F Fig. 4 shows a state diagram for the complete system of robots with a self-organization behavior. The second state diagram models the self-steering behavior for the robots network. The self-steering behavior state diagram can be seen in Appendix F Fig. 5. Fig. 5 shows how the complete system of robots progress through each state to work as a group to get to a desired location.

TABLE III. A LIST OF OBJECTIVES

Objectives
The robots network should be cooperative
The robots network should be self-organizing
The robots network should be self-steering
The software should be portable
The robots network should be reactive
The robots network should be adaptive

The objectives shown in Table III can be described as the following:

- To be cooperative, the individual agents within the robots network must work together to achieve a consensus on how to complete a task.
- To be self-organizing, agents within the robots network autonomously organize into a formation.
- To be self-steering, the robots network must maintain formation while autonomously moving towards a desired location.
- To be portable, the programmed control strategy must be easily transferable to other robotic platforms.
- To be reactive, the robots network must respond (change its behavior) when foreign objects are detected.
- To be adaptive, the robots network must assimilate, or adapt, new robotic agents being added to the network.

The self-steering objective can be achieved by completing three functions, agent-to-agent repulsion, agent-to-agent orientation, and goal attraction, to achieve the desired results. The first function, agent-to-agent repulsion, ensures that the individual agents of the network maintain a constant distance from each other to avoid collisions. This may be accomplished by having the agents being aware of the distance between to their local neighbors, comparing those values to a fixed value, and moving away as necessary. Agent-to-agent orientation uses an agent's neighbors to determine its location and make any adjustments to get into its proper position. By using the neighbor's data, such as *x* coordinates, *y* coordinates, and angle, an agent can determine its own coordinates, using methods such as trilateration. The final function, goal attraction, means that an agent should be aware of its distance and orientation compared to the end goal. The individual agents should actively move towards the goal, by adjusting their position and compare it to the end point.

The self-organizing objective can be accomplished by completing three functions, agent-to-agent localization, agent-to-agent attraction, and position calibration. Agent-to-agent localization has the individual agents using information from local agents to determine where the agent is within the network. Agent-to-agent attraction ensures that individual agents do not move to far away from other agents by measuring the distance to its nearest neighbors and making sure that the distance does not exceed a predetermined value. Position calibration has the agents move to where they need to be in the formation and has the agents adjust themselves to maintain their proper position.

The reactive objective is achieved by a combination of two functions, object detection and object avoidance. Object avoidance ensures that the network is aware of the environment and can react to any changes in it. Individual agents should detect obstacles, relay that information to its neighbors, and adjust its movement accordingly to avoid the obstacle and still avoid collisions with neighboring agents.

In order to achieve the adaptive objective, the system needs a function that detects new agents. This function should identify the addition of any new agents into the overall system and assimilate the new agent into it. A list of functions is shown in Table IV

TABLE IV. A LIST OF FUNCTIONS

Functions
The robots network shall have agent-to-agent repulsion
The robots network shall have agent-to-agent orientation adjustment
The robots network shall have goal attraction
The robots network shall have agent-to-agent localization
The robots network shall have agent-to-agent attraction
The robots network shall calibrate position
The robots network shall detect object(s)
The robots network shall avoid object(s)
The robots network shall respond to new agents

The functions for the cooperative control of heterogeneous mobile robots network have the following specifications:

1. Function: Agent-to-agent repulsion

Procedural Specification: Individual agents will repulse other agents to avoid collisions

Performance Specification: Agents are at least 4 centimeters apart, or within a scale of the QBot 2

2. Function: Agent-to-agent orientation adjustment

Procedural Specification: Using data from neighboring agents, an agent will adjust its orientation

3. Function: Goal Attraction

Procedural Specification: agents will move to a desired location

Performance Specification: At least 75% of agents must move to within 50 centimeters of the range of the desired goal

4. Function: Agent-to-agent Localization

Procedural Specification: Agents will determine their location using data from local neighbors.

5. Function: Agent-to-agent Attraction

Procedural Specification: Using local data, individual agents will stay within range of other agents.

6. Function: Position Calibration

Procedural Specification: Agents shall move to their correct position in a given formation

Performance Specification: At least 75% of agents must move to within 50 centimeters of the agent's correct position

7. Function: Object Detection

Procedural Specification: Individual agents will be able to detect any objects that are within range

Performance Specification: Agent detects object at least 0.5 meter away

8. Function: Object Avoidance

Procedural Specification: Agents shall adjust headings to avoid obstacles, while still avoiding each other.

Performance Specification: At least 75% of agents must avoid objects

9. Function: Adaptive Response

Procedural Specification: If a new agent is added to the network it shall be integrated into the existing network.

B. *Design Approach & Method of Solution*

The project is a proposed control method to produce emergent behaviors (self-organization and self-steering). The cooperative control of heterogeneous mobile robots network consists of two main points:

- Mathematical model:
  1. The model for the robotic platforms, which will be utilized for the design and simulation of control strategies.
- Communication:
  1. Communication is performed by neighboring agents.
  2. Ideally, all communication is achieved through infrared transceiver and receiver pairs.

The mathematical model for an individual agent is necessary to design the cooperative control strategies. Ideally, the model for a single agent would be linear. The other approach is to create a non-linear robot model. A linear model is preferred because of its simplicity; fewer variables will decrease the complexity of the control algorithm. To develop a linear model, a non-linear, unicycle model (differential drive model) must be transformed. The equations describing a unicycle model are the following:

$$X' = V\cos(\theta) \quad (1)$$

$$Y' = V\sin(\theta) \quad (2)$$

$$\theta' = \omega \quad (3)$$

The aforementioned equations decompose the velocity vector of a two-dimensional model into its corresponding horizontal, or *X*, velocity (1) and vertical, or *Y* velocity (2). Equation 3 is the angular velocity of the model, or the rate at which the orientation *θ* changes. The unicycle model can be seen in Fig. 2.

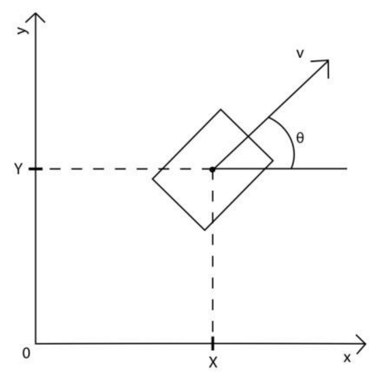

Fig. 2: Non-linear Model of a Robot Agent

In the non-linear, unicycle model, each equation uses the center of mass as the reference point. To linearize the unicycle model, the reference point is moved from the center of mass to the front of the robot agent. This reference point change can be seen in Fig 3.

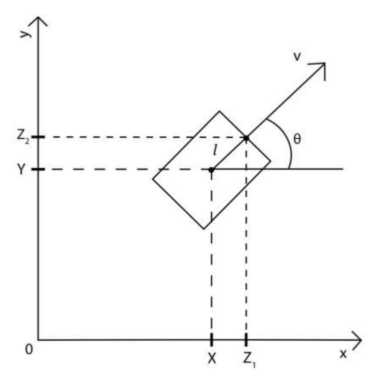

Fig. 3: Linear Model of a Robot Agent

The change in reference produces a new coordinates (Z_1, Z_2). The new coordinates can be derived from the following equations:

$$Z_1 = X + l\cos(\theta), \qquad (4)$$

$$Z_2 = Y + l\sin(\theta), \qquad (5)$$

l is the length from the center of mass to the front of the robot agent. In other words, Z_1 is the distance, X, plus the horizontal distance to the front of the agent and Z_2 is the distance, Y, plus the vertical distance to the front of the agent. Using the new coordinates, the horizontal and vertical velocities of the model can be determined by taking the derivative of equations 4 and 5. The derivative results in the following:

$$Z_1{}' = X' - l\sin(\theta)\theta' \qquad (6)$$

$$Z_2{}' = Y' + l\cos(\theta)\theta' \qquad (7)$$

Substituting equations (1), (2), and (3) into equations (6) and (7) yields:

$$Z_1{}' = V\cos(\theta) - l\sin(\theta)\omega \qquad (8)$$

$$Z_2{}' = V\sin(\theta) + l\cos(\theta)\omega \qquad (9)$$

Transforming equations (8) and (9) into matrix form:

$$\begin{bmatrix} Z_1{}' \\ Z_2{}' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -l\sin(\theta) \\ \sin(\theta) & l\cos(\theta) \end{bmatrix} \begin{bmatrix} V \\ \omega \end{bmatrix} \qquad (10)$$

Equation (10) is set up as a standard matrix equation ($b = Ax$). If matrix A is singular, then the matrix is not invertible. To check if a matrix is singular, the determinant must be determined. If the determinant is 0, then the matrix is singular. In this instance, the matrix of importance, A, is nonsingular or invertible. This result means that the overall model is linear.

Communication is an important topic that must be addressed in order for this project to be successful. Communication must be performed locally, or by neighboring agents. To achieve local communication, a threshold range can be determined by using signal strength. Another possible solution is to use distance measurements to determine which agents are local to one another.

Not only must communication be done locally, but it must also be possible for different robot platforms to communicate with other robot platforms. Local communication can be achieved by using a common communication protocol in every platform. These messages should be understood by using infrared transceivers and receivers in each robot test platform.

There is an alternative solution, if infrared only communication is not possible. The alternative solution is to create a hierarchy based communication design. This would be based on the size of the test platform as well as the adaptability to additional sensors. Additional sensors would be added, such as an ultrasonic sensor, to provide an agent with more information. Then the agent will give this new information to an intermediate agent. This intermediate agent will determine what information to send to the least adaptable test platform.

To test the overall system, the software and hardware need to be tested individually before being integrated as a complete system. Algorithms need to be validated by using mathematical representation, then computer simulation, and finally implementation on platforms. The hardware needs to be calibrated to ensure accuracy of each agent. Once all software and hardware has been validated then it can be integrated into the overall system. The overall testing for each function is explained in Appendix J.

C. *Economic Analysis*

The cooperative control of heterogeneous mobile robots network project is part of a larger project commissioned by the United States Air Force Research Lab. The Air Force Research Lab will be funding the project through a grant given to Bradley University. The grant agreement number provided for this project is FA8780-13-0109. The cost of the project is projected to be $23,610.00; an additional 20% was added to account for unanticipated costs. For a detailed description of the components and their cost see Table VI in the Appendix B.

D. *Project Timeline*

There are six major tasks that need to be completed, in order to have a successful project by March 2016. A Gantt chart (Table VII, Appendix C) detailing these tasks can be found in the Appendix. The critical path necessary to complete this project is as follows: individual behaviors (deadline-November 15), individual communication (deadline-December 15), integrated communication (deadline-December 15), control algorithm design (deadline-December 15), integrated behavior (deadline-January 16), and testing (deadline-March 16). Testing will last throughout the design process, as it is an iterative approach. A detailed Gantt chart for the deliverables during the project can be seen in Table VIII, Appendix D.

E. *Division of Labor*

Labor for the heterogeneous mobile robots network has been subdivided into groups based on the robotic test platforms. The groups will perform studies on individual behaviors and communication for each specified test platform. Using the information obtained from the individual studies, the test platforms will then be integrated together to exhibit the desired behaviors. Test platform integration will be implemented by the team as a whole. For a detailed division of labor see Appendix A, Table V.

F. Societal and Environmental Impacts

Due to the fact that this project was commissioned by the Air Force Research Lab, the main focus of the societal and environmental impact of a heterogeneous mobile robots network has been geared toward how it would be utilized by the military. The knowledge gained from the proposed project, once completed, tested, and implemented, could enhance the efficiency of search and rescue missions, surveillance and reconnaissance missions, as well as drone strikes a great deal.

Using the proposed cooperative control strategy, a network would be able to be deployed more swiftly. The agents would be able to search for one person, for instance someone who has been shot down in a fighter jet, or for a larger object such as a commercial airline (Malaysia 370[1] or El Faro [7]). The outcome should be a faster and more efficient method of detection that could ultimately save lives.

Surveillance and reconnaissance missions are another closely related task, which would entail the use of a network to survey landscapes and gather valuable information. This function is desirable and usable as an effective means for military espionage. The information that it would provide would be beneficial to America's interests. The same function implementation can be used to collect data on natural disasters such as earthquakes, floods, and forest fires. Tracking the progress, or damage, of a natural disaster in real time, could help civil authorities respond to and assess the situation in a more comprehensive manner.

The cooperative control of heterogeneous mobile robots network could be used to facilitate more accurate drone strikes. Using this technology, a network could have the capability of conducting multiple drone strikes simultaneously. Drone strikes are used sparingly by the United States military, and are generally authorized by the president, in order to save the lives of military personal as well as civilians. Overall, the cooperative control of heterogeneous mobile robots network could saves lives as well as keeps the nation's weapons technology on the cutting edge.

III. SUMMARY AND CONCLUSIONS

This proposal details the cooperative control of heterogeneous mobile robots network project. The project goal is to create an autonomous system of robots that exhibit self-organizing and self-steering behaviors. The completion of this project would allow interested parties, such as the United States Air Force, to benefit from a greater understanding of emergent behaviors in robotic networks.

IV. REFERENCES

[1] Cbsnews.com, 'Malaysia Airlines Flight 370 - News, Pictures, & Videos - CBS News', 2015. [Online]. Available: http://www.cbsnews.com/malaysia-airlines-flight-370/. [Accessed: 10- Oct- 2015].

[2] K-team.com, 'Kilobot – Software', 2015. [Online]. Available: http://www.k-team.com/mobile-robotics-products/kilobot/software. [Accessed: 27- Apr- 2015].

[3] Quanser.com, 'Quanser - QBot 2 for QUARC', 2015. [Online]. Available: http://www.quanser.com/Products/QBot 2. [Accessed: 21- Apr- 2015].

[4] Roadnarrows-store.com, 'Kilobot - Micro Robots - Robots - All Products', 2015. [Online]. Available: http://www.roadnarrows-store.com/products/robots/micro-robots/kilobot.html. [Accessed: 27- Apr- 2015].

[5] Roadnarrows-store.com, 'Kilobot Starter Pack', 2015. [Online]. Available: http://www.roadnarrows-store.com/kilobot-starter-pack.html. [Accessed: 27- Apr- 2015].

[6] Roadnarrows-store.com, 'e-puck with charger', 2015. [Online]. Available: http://www.roadnarrows-store.com/e-puck-with-charger.html. [Accessed: 27- Apr- 2015].

[7] Washington Post, 'Cargo ship likely sank during hurricane, Coast Guard says', 2015. [Online]. Available: http://www.washingtonpost.com/business/economy/cargo-ship-likely-sank-during-hurricane-coast-guard-says/2015/10/05/2bab59ae-6b79-11e5-aa5b-f78a98956699_story.html. [Accessed: 10- Oct- 2015].

V. APPENDIX

A. *Division of Labor*

TABLE V. A DIVISON OF LABOR OUTLINE

Division of Labor Outline		
Task	**Subtasks**	**Team Members**
Individual Behavior	Kilobot	Jared/Brittany
	QBot 2	Ryan/Greg
	E-puck	Jared/Brittany
Individual Communication	Kilobot - Kilobot	Jared/Brittany
	E-puck - E-puck	Jared/Brittany
	QBot 2 – QBot 2	Ryan/Greg
Integrated Communication	Kilobot - E-puck	Jared/Brittany
	Kilobot – QBot 2	All
	E-puck – QBot 2	All
Algorithm Design	Linear Model	All
Integrated Behavior	Self-organizing	All
	Self-steering	All
Testing	Software Implementation	All
	Hardware Implementation	All

B. *Budget*

TABLE VI. COSTS

Project Cost Analysis		
Item	**Quantity**	**Total Price**
QBot 2	3	$9,999.00
Kilobot	20	$4,583.00
E-puck	3	$5,093.00
Kilobot IDE	1	$ 0.00
E-puck IDE	1	$ 0.00
MATLAB Courseware	1	$ 0.00

C. *Gantt Chart – Tasks*

TABLE VII. A GANTT CHART OF TASKS

Task Name	Group Member	Finish by Date	Sep-15	Oct-15	Nov-15	Dec-15	Jan-16	Feb-16	Mar-16	Apr-16
Individual Behavior										
Kilobot Sensors	Jared/Brittany	9/28/2015								
Kilobot Communication Protocol	Jared/Brittany	10/12/2015								
QBot 2 Image Processing	Ryan/Greg	10/5/2015								
QBot 2 Sensors	Ryan/Greg	10/28/2015								
QBot 2 Communication Protocol	Ryan/Greg	10/19/2015								
E-puck Sensors	Jared/Brittany	10/26/2015								
E-puck Communication Protocol	Jared/Brittany									
Individual Communication										
Kilobot – Kilobot	Jared/Brittany	10/19/2015								
E-puck – E-puck	Jared/Brittany	12/14/2015								
QBot 2 – QBot 2	Ryan/Greg	11/2/2015								
Integrated Communication										
Kilobot – E-puck	Jared/Brittany	12/14/2015								
Kilobot – QBot 2	All	11/16/2015								
E-puck – QBot 2	All	12/14/2015								
Algorithm Design										
Design Linear Based Model	All	12/14/2015								
Integrated Behavior										
Self-organization	All	1/9/2016								
Self-steering	All	1/16/2016								
Testing										
Software Implementation	All	3/7/2016								
Hardware Implementation	All	3/7/2016								

D. *Gantt Chart – Deliverables*

TABLE VIII. A GANTT CHART OF DELIVERABLES

Task Name	**Group Member**	**Finish by Date**	**Sep-15**	**Oct-15**	**Nov-15**	**Dec-15**	**Jan-16**	**Feb-16**	**Mar-16**	**Apr-16**
Deliverables										
Project Proposal - Oral Presentation	All	10/6/2015								
Project Proposal - Document	All	10/15/2015								
Webpage Release	All	10/28/2015								
Fall Progress Presentation	All	11/19/2015								
Fall Performance Evaluation	All	11/19/2015								
Fall Performance Review	All	12/3/2015								
Spring Progress Presentation	All	2/18/2016								
Student Expo Abstract	All	3/18/2016								
Project Demonstration	All	3/24/2016								
Final Presentation	All	4/7/2016								
Student Expo Poster Printing Deadline	All	4/11/2016								
Student Expo Poster Setup	All	4/12/2016								
Student Expo	All	4/14/2016								
Final Report (Draft)	All	4/14/2016								
Final Report	All	4/28/2016								
Final Web Page	All	4/28/2016								
Advisory Board Poster Printing Deadline	All	4/28/2016								
Advisory Board Poster Presentation	All	4/29/2016								

E. *Self-steering and Self-organization State Diagrams*

There are two different state diagrams that are necessary for this project. The first state diagram models self-organization for the robots network. Fig. 4 shows a state diagram for the complete system of robots with a self-organization behavior. The second state diagram models the self-steering behavior for the robots network. The self-steering behavior state diagram can be seen in Fig. 5. Fig. 5 shows how the complete system of robots progress through each state to work as a group in a self-steering behavior to get to a desired location. The state diagrams are similar because the diagrams include the initialize, acquire sensor data, adjust position or orientation, and check position and orientation states. The initialize state includes all initialization process the robots must go through. The acquire sensor data state is where the agents will acquire the local data. From the local sensor data the agent will adjust its position and orientation depending on the data that is received, and the algorithm that it is executing. The next state is when the agent will check the position and orientation and determine if it is in the right position and orientation, if it is not it will go back to the adjust position and orientation state. The self-organization state diagram would move into the next state if the agents are in the correct position and orientation. The agents will idle until all agents are in position and orientation. Once all agents are in the correct position and orientation the formation has been achieved. The self-steering state diagram would follow the same states as the self-organization until the move in formation state. The agents would move into the next state, where the agents are checking if in the desired location, if not then the agent's state moves back into the move in formation state. It will then check again, and repeat this process until the agents are in the desired location. Once in the desired location the agent's will move to the end state.

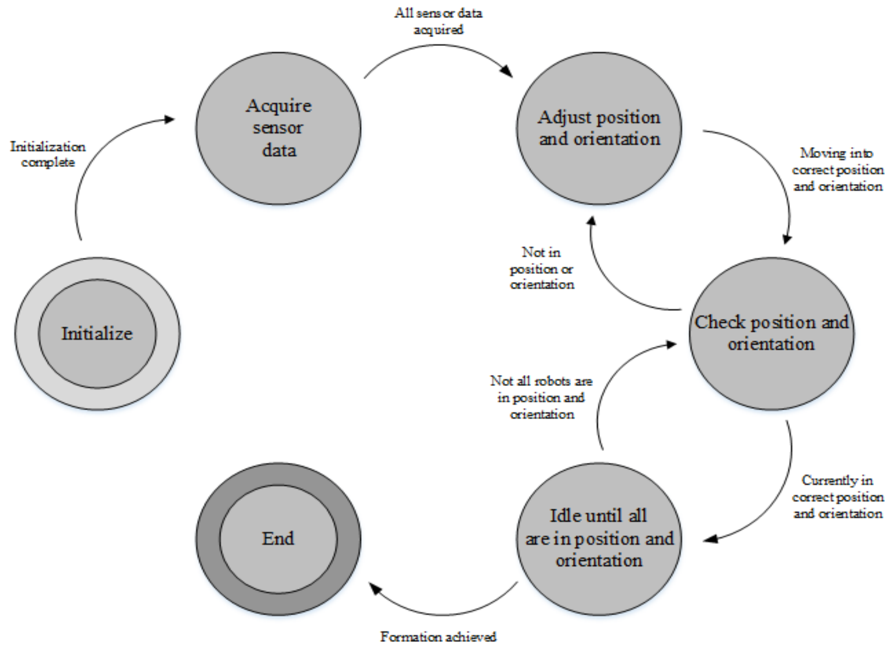

Fig. 4. Self-organization State Diagram

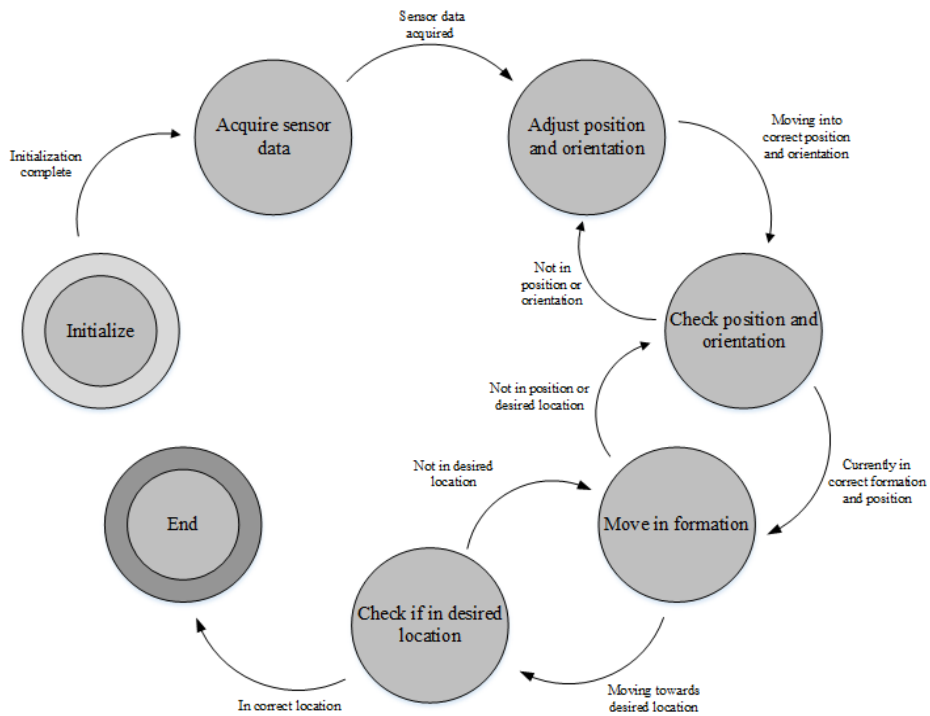

Fig. 5. Self-steering State Diagram

F. *Testing*

- **Static Testing -** Code is not run
    - Pair programming – One person programs code, while other reviews currently written line (After so much time, partners switch roles).
    - Walkthrough – code is explained to team. Team provides questions about possible errors, standards violations, and other problems.
    - Inspection – code read in detail by other team member.
- **Dynamic Testing -** Code is run
    - Load testing – stress code with different inputs
    - Fault injection – introduce faults to determine error handling of code

Tests should be done in a bottom up method. Bottom up method means code should be tested at the function level, then tested with the integration of functions, and finally at the black box level.
The black box level testing method, for this project, is mostly visual testing. The system should be tested multiple times and the software should be rated by averages, not single outcomes. Each outcome should be recorded accordingly.

G. *Test Platform Description - Kilobot*

Platform: Kilobot

Developer: Harvard University Self Organizing Systems Research Group

Microcontroller: Atmega328p

- 8 Bit processor
- 8 Megahertz central processing unit,
- 32 Kilobytes of flash
- 1 Kilobyte of EEPROM

Features:

- 2 Independently controlled vibration motors for movement
- 1 Infrared light sensor
- 1 Light intensity sensor
- 1 Red/green/blue light emitting diode

Fig.6. Test Platform – Kilobot [2]

H. *Test Platform Description – E-puck*

Platform: E-puck

Developer: École polytechnique fédérale de Lausanne

Microcontroller: dsPIC30F6014A

- 16 Bit processor
- Digital signal processing unit
- 64 Megahertz central processing unit
- 8 kilobytes of random access memory
- 144 kilobytes of flash

Features:

- 2 Independently controlled stepper motors for movement
- 8 Red light emitting diodes
- 8 Infrared proximity sensors
- 1 3D accelerometer
- 3 Microphones
- 1 Color complementary metal oxide semiconductor camera with 640 x 480 resolution
- Bluetooth radio link
- 1 Infrared light receiver

Fig. 7. Test Platform – E-puck [6]

I. *Test Platform Description – QBot 2*

<u>Platform:</u> QBot 2

<u>Developer:</u> Quanser

<u>Microcontroller:</u> Gumstix DuoVero Zephyr

- 1 Gigabyte of DDR SDRAM
- 32 Megabytes of flash
- 1 Gigahertz central processing unit

<u>Features:</u>

- 2 Independent motors for movement
- 1 Kinect sensor
- 1 Three-axis gyroscope
- 2 Digital wheel drop sensors
- 3 Digital bump sensors
- 3 Cliff sensors
- 2 Light emitting diodes

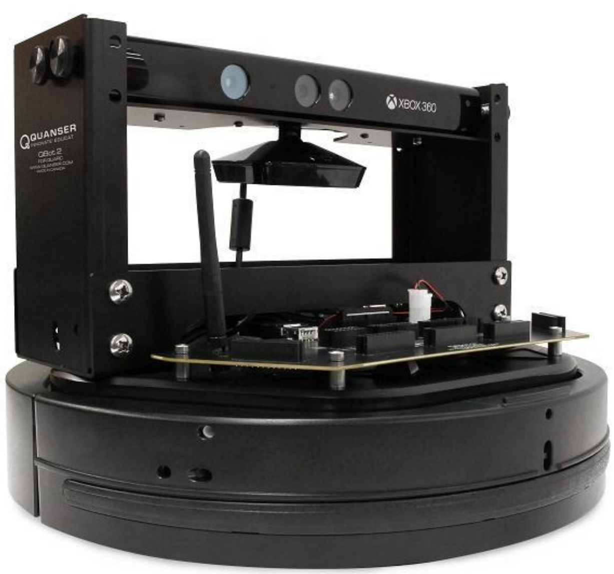

Fig. 8. Test Platform – QBot 2 [3]

J. *Function Testing Procedures*

The procedures that will be used to test each function are the following:

1. **Function: Agent-to-agent repulsion**
   To test the function of Agent-to-agent repulsion, a pair of agents will be put within a touching distance, and then the agent's must disperse to at least 4 centimeters apart.

2. **Function: Agent-to-agent orientation adjustment**
   To test the function of Agent-to-agent orientation adjustment, a pair of agents will be put at least 4 centimeters apart, and will have to determine its orientation and depending on the ending goal adjust the orientation of the agent to the desired orientation.

3. **Function: Goal Attraction**
   To test the function of goal attraction, a group of agents will be placed at a determined location, and will have to move to a desired location. 75% of the agents must move to within 50 centimeters of the desired location for this function to be deemed successful.

4. **Function: Agent-to-agent Localization**
   To test the function of agent-to-agent localization, a group of agents will be placed next to each other. Using LEDs the agents will display their location.

5. **Function: Agent-to-agent Attraction**
   To test the function of agent-to-agent attraction, a pair of agents will be placed within 4 centimeters of each other. The agents must move while staying within a 4 centimeter range of the other agent.

6. **Function: Position Calibration**
   To test the function position calibration, a group of agents will be placed at a determined location, and will move their correct position in a given formation. 75% of the agents must move within 50 centimeters of the agent's correct position.

7. **Function: Object Detection**
   To test the object detection function, each agent will be placed one meter away from an object. It will then drive forward until the object is detected, where the agent must stop at least 0.5 meters away from the object.

8. **Function: Object Avoidance**
   To test the object avoidance function, each agent will placed one meter away from an object. It will then drive forward until the object is detected, when it detects the object the agent must turn right at least 0.5 meters away and proceed forward, while avoiding the object.

9. **Function: Adaptive Response**
   To test the adaptive response function, a pair of agents will be executing an algorithm, and a third agent will be introduced. The third agent must execute the programmed algorithm correctly.