TEAM PANTHER SWARM
FIU MIAMI

NASA SWARMATHON 2016: FIU PANTHER SWARM TEAM TECHNICAL REPORT

FIU Panther Swarm Team*
Scott Jagolinzer, Jonathan Hipps, Jorge Larrarte,
Laura Reyes, Nicolas Polignan, Tomasso Benigni, Rodolfo Guerrera,
Rudnei Moran, Julian Sellan, Sebastian Maciel, Frank De La O

Faculty Advisor: Dr. Sabri Tosunoglu

Florida International University
10555 West Flagler Street
Miami, Florida 33174

April 2016

***Complete FIU team is listed in the Appendix.**

Development of FIU Panther Swarm Algorithm for NASA's Swarmathon Competition

Scott Jagolinzer, Jonathan Hipps, Jorge Larrarte, Nicolas Polignan,
Sebastian Maciel, Frank De La O, Rodolfo Guerrero, Sabri Tosunoglu

Florida International University
Department of Mechanical and Materials Engineering
Miami, Florida 33174

***Abstract* – This paper presents the research, architecture and program development of swarm robot behavior for the prospective exploration of Mars. It also details the steps from initial algorithm brainstorming, to experimentation, to final algorithm development and testing. The strengths and weaknesses of the hardware and how it affects the development of the Swarmathon algorithm are also discussed. The final algorithm was developed to limit the error of the rover pathing and reliably return to the nest after each search loop.**

Keywords - Multi-robot system, swarm robots, UGV, swarm controllers, architecture, algorithm

I. INTRODUCTION

The "swarm" term has been used to identify different types of system in engineering, computer science and nature itself. This type of robotic science has increasingly in the rise lately as one of the most prominent ways to explore and recollect samples of unexplored planets or asteroids. The advantages of this systems are robust, adaptable, scalable, inexpensive and efficient. The Swarmathon is a competition organized by the National Aeronautics and Space Administration (NASA) for multiple universities across the United States to develop a unique search algorithm for a set of swarm rovers to seek, collect, and return tags around an arena. This is meant to help drive the ingenuity in development of swarm algorithms for use in space exploration and collection. Although the code being developed is intended for rovers scanning tags in a parking lot, the concepts of the algorithm can be adapted to a rover or swarm system on unexplored planets or asteroids to search for water, rocks or other materials that could be tagged, collected, or mined.

As one of the participating universities, Florida International University (FIU) established the goal of working together to develop a unique search algorithm that could pick up and return as many tags possible in the quickest possible manner while also limiting the possibility of collisions or getting stuck. Different algorithm propositions were discussed and tested through simulation prior to come out with the final search path. This process was iterated upon team's inspiration from nature behavior such as bee colonies or previous approaches like a spiral method. Through real robot testing, the team was able to run and iterate upon the multiple search algorithms until a satisfactory one was obtained. This final algorithm is the one that works more reliable collecting tags and avoiding the obstacles such as walls and others swarm robots.

Members of the team in representation of the FIU Robotics Lab will continue to research and modify the algorithm for future editions of the Swarmathon competition.

II. LITERATURE SURVEY

Swarm robotics is a field of study that focuses on the coordination of robots to create a system of constituent robots that work together towards a collective effort. Since the 1980s, swarm robots have been reviewed for their forthcoming uses in the foreseen future. Some of the advantages found in the research of multi-robot systems over a single robot system are:

1. Undertaking complex tasks
2. Designing several unpretentious robots is simpler than building a single dominant robot
3. Multiple robots can gather a greater overall view of the problem than a single robot
4. Introducing multiple robots allows for greater robustness through redundancy

Considering that robots can be tailored to be aerial, terrestrial or underwater vehicles and can be developed on a microscale or macroscale, there is a plethora of applications for swarm intelligence. Such technologies can be especially useful for applications such as mining, foraging, and search-and-rescue missions. The idea of swarm intelligence originated from observing biological phenomena and communication between organisms. By fostering the "mindset" of these organisms, such as the stigmergic trails of ants or the flight paths of bird swarms, there are several advantages that are brought about such as the implementation of a decentralized and autonomous system and the flexibility that is intrinsic to such a system.

In order to garner various ideas of multiple approaches, a variety of research papers on swarm robot algorithms were sought. Multiple research papers based on algorithms for search and rescue, ant inspired obstacle avoidance and relative localization methods were carefully analyzed and explained. The research was conducted in order to implement new ideas in the proposed algorithm to achieve the barcode search to be employed in NASA competition.

A. Swarm-Based Path Creation in Dynamic Environments for Search and Rescue [9]

The approach described in this paper is suitable to later revisions of the NASA Swarmathon in which there are obstacles. The current challenge in the NASA Swarmathon is to locate the codes; there is a similar task described in this paper. However, in this situation, the robots are also tasked with identifying obstacles and finding the best path [9], as well requirements based on different zones in which the area is divided into. This course is also dynamic, in that the conditions in these aforementioned zones are continuously changing. This information will be very useful for later revisions of the competition. There is still useful information for the current competition in this paper. Although this paper does not get into the specifics of the algorithm, it does describe the methodology in which the tasks are separated amongst the different robots. These task designations are conditional based upon the situational conditions.

B. A Hybrid Search Algorithm for Swarm Robots Searching in an Unknown Environment [11]

The purpose of this paper, as in others, is for the search and rescue of humans [11]. While these task requirements exceed that of the NASA Swarmathon, there are many aspects that can be observed from this pervious work and perhaps implemented. The algorithm implemented here consists of a hybrid setup; that is, it relies on two algorithms, rather than just one. The specific situation parameters at hand will dictate which algorithm to implement. One of the algorithms is similar to the preloaded software on the NASA Swarmathon project, in that it is an algorithm that implements random paths. However, in this case, the robot implements the random algorithm if it has not found its target for a preset amount of time. The other algorithm is what is referred to a dynamic particle swarm optimization (DPSO). This algorithm is dynamic, in that it continuously communicates with the other robots, feeding it information and making decisions based on this information. The inspiration for the implementation of two algorithms is based on a study of animal preying habits. When an area of interest is identified, a concentration factor will be applied and the random algorithm will be abandoned momentarily while the swarm concentrates on the specific area. A characteristic of this algorithm that may be implemented in the NASA Swarmathon competition is the variation of speed based on its area. The concentration factor includes many characteristics that will improve its chances of success, one of which is altering the speed. However, when the program is implementing the random algorithm, it will maintain a constant speed. Great results were achieved with the algorithm; however, it was only tested in ideal situations in computer simulations. Testing with more realistic situations will be a valuable indicator of its success.

C. The Ant and the Trap: Evolution of Ant-Inspired Obstacle Avoidance in a Multi-Agent Robotic System [12]

This paper, composed as a thesis for the pursuance of a Master of Science and co-authored by Dr. Melanie Moses, who is one of the organizers of the Swarmathon competition, focuses on the iAnt, a modular and affordable swarm robot.

The author delves into the technologies that comprise the fourth generation of the iAnt [12] and the modifications that were made to it for the purposes of the project. A governing topic of discussion in this paper is also the methods the author chose to implement as search algorithms. The algorithms of interest are based on the research that has been garnered regarding stigmergy and other biological phenomena that relate to the coordination between animals; particularly ants. From modeling the behavior of ants and parametrizing said models, the iAnt swarm system could, theoretically, resemble the well-coordinated behavior of ants. The algorithms at hand are the central place foraging algorithm (CPFA), which affords the robots the ability to perform stochastic movements, ant-inspired obstacle avoidance and simple communication between them, and the CPFA-Trails, which affords the extra ability to communicate using stigmergic trails. A neural network, referred to by the author as a genetic algorithm, allows for real-time data to serve as training data that the iAnts can learn from and improves their "learning curve." This genetic algorithm defines how the swarm robots will respond to unanticipated obstacles and is ultimately dictated by a series of parameters. The parameters that make up the decision model are the following: informed search decay rate, pheromone decay rate, pheromone laying rate, search give-up probability, pheromone following rate, travel give-up probability, and uninformed search variation. The iAnt "decides" its path forward using these parameters.

Effectiveness of the algorithms, complemented by the genetic algorithm, were evaluated using simulation software. The final results show that between the CPFA and the CPFAT algorithms, superiority is not as clear cut as anticipated. CPFAT, while expected to consistently be a more effective search algorithm, resulted in minor improvements and, in some cases, worse decision-making. Ultimately, both algorithms demonstrated the capacity for the iAnt swarm robot to serve as an affordable vehicle for swarm robotics research and development.

D. *Relative Localization Method of Multiple Micro Robots Based on Simple Sensors [8]*

The algorithm portrayed in this paper represents a relative positioning approach [8]. This approach is based on an infrared sensor and a compass. The algorithm helps generate relative positions amongst a group of robots allowing them to be aware of their neighboring robots. By attaching a compass to every robot, there will be a common reference direction amongst them all. That direction being the arrow that always points north on a compass. The compass gives the robot self-orientation while the infrared sensor allows the robots to localize each other. By combining both, the robots will know their neighboring robots' locations with respect to themselves. Awareness amongst a group is essential in order for the members to prevent collisions and work effectively.

E. *Distributed Colony-Level Algorithm Switching for Robot Swarm Foraging [5]*

Another addition to the literature survey includes a paper composed by three Harvard students who explored various foraging algorithms at a colony-wide level [5]. In lieu of limiting the mobility of the robots to one algorithm, the authors' purpose was to incorporate a decision model for the swarm that is contingent upon its environment. In total, there are two distributed foraging algorithms and a final one that allows the swarm to choose between the two previous ones.

The first algorithm is addressed in this paper as the gradient algorithm. This ingenious algorithm allows the robots to efficiently survey and navigate between the base and the food source. Each robot can switch between two states: a state of seeking and a fixed, beacon state. The survey is generated by using two gradients: one leading to the nest and one leading to the food source. Once the food source is found, the robots that are mid-path will switch into beacon mode and make the gradient information available to "point-out" the path to and fro the nest. This algorithm operates in a short range but is fast.

The second distributed foraging algorithm is addressed as the sweeper algorithm. The idea behind this one is create a front of robots that essentially sweep an area to encounter a food source. The main method of communication between the robots per this algorithm is a virtual force that acts a "spring force". The robots would be arranged in a line that extends outward from the nest to an arbitrary boundary and would sweep through like the hand of a clock. Once a food source is found, some robots become stationary beacons that allow for navigation of the swarm similar to the first algorithm. This algorithm has a much longer range than that of the gradient algorithm but is slower.

The last algorithm is an adaptive algorithm that allows the swarm to choose which of the two aforementioned algorithms it wants to be dictated by depending on its environment and its success rate. The following flowchart (Figure 3) delineates the decision model that the swarm will base its actions upon. Ultimately, the adaptive algorithm was the most successful in terms of efficiency and success rate because it was able to capitalize on the efficacious methodology behind the sweeper algorithm but at a faster pace.

III. SENSORS & COMMUNICATION

The swarm rovers come with an array of onboard sensors. They include GPS, a compass, wheel encoders, visual camera, and ultrasound sensors. The rovers are also able to communicate with the host system via Wi-Fi, which coordinates the movements of all the connected rovers.

The rovers are run by a code written in C++ that runs within ROS (Robot Operating System). This system allows for the rovers to communicate over Wi-Fi by publishing and subscribing to messages that are broadcasted via topics. This allows for the rovers to pass along their positional data along with information about the tags that they come across. By breaking down the transfer of information to publishers and subscribers via topics allows for certain parts of the program or certain rovers to access only the information that is relevant to them instead of bombarding each robot with a stream of all the real time data that is being passed from the individual rovers to the host machine.

IV. ALGORITHM DEVELOPMENT (METHODS & EXPERIMENTS)

Initially the senior level robotics class worked in small groups to research and develop their own algorithm for the Swarmathon competition with some of the algorithms described below. Each team performed research into the field of swarm robotics and purposed a searching method to the class before moving on to formal coding of the algorithms.

Spiral Surge algorithm has the rover moving to a random location predetermined by the code developer then performing a spiral movement using the formulas described below until a tag is found. At that point, the swarm robot would return the tag to the nest and repeat the process. Figure 1 shows the simulation results for this algorithm and the path traced by the swarm robots during the simulation.

```
goalLocation.theta = currentLocation.theta + 0.25;
```

goalLocation.x = currentLocation.x + (0.375 * cos(goalLocation.theta));

goalLocation.y = currentLocation.y + (0.375 * sin(goalLocation.theta));

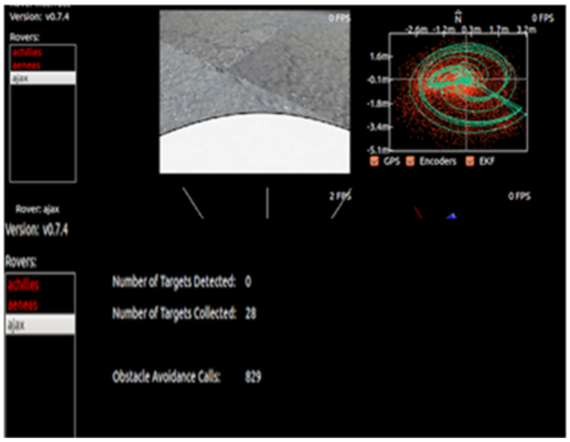

Figure 1: Simulation Results for Spiral Surge Algorithm

Box search algorithm is similar in functionality of the spiral search algorithm but uses a box pattern. The box search pattern matches the shape of the arena so a collision with the walls would not be expected.

Out and back algorithm has each of the rovers starting at the nest and picking a random compass heading. After that, the swarm robot goes in a straight line until the wall is detected and it returns to the nest. If a tag is found, it performs the same line path again until no further tags are found along that path and then selects a new random heading to search.

Rake search algorithm has the rovers performing sweeps across the arena in the same direction. The rovers pick their rake path at random, repeating a path that has yielded a tag until no further tags are found along that path. The rovers use the x-axis of the arena to return tags and reset their rake position to a new one.

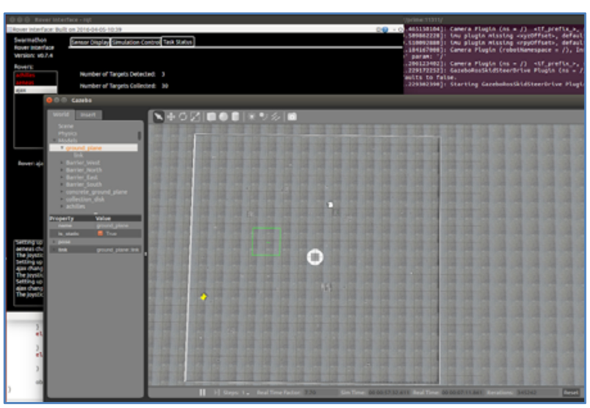

Figure 3: Simulation Results for Rake Search Algorithm

The team compared and ran all algorithms in the simulation code to determine the strengths and weaknesses of each algorithm. From there, some codes were reworked to draw upon strong suits of other team's codes and again simulations were rerun on promising codes.

Figure 2: Scale Model Testing of Algorithms in Class

Only the most efficient algorithms were run on the physical robots. At this point, the team found out issues with the navigation of the physical rovers. With most of the algorithms being highly dependent on knowing their precise location in the arena, the codes needed to be worked on to account the lack of accuracy in the positional data.

The testing performed in simulation and the physical rovers were vital in being able to develop the final algorithm. The simulation was a great tool in verifying that the algorithm is performing what the code developer intended. However, since the simulation is an idealized version of the rovers and the desired arena, all the obtained data is of a much higher accuracy than the real rovers report. Algorithms ran successfully through simulation allowed the teams to determine what data was reliable (compass and sonar) and unreliable (wheel encoders and GPS). FIU will eventually alter the code to work in a way that used the more reliable data while only using the more unreliable data at some instances where no other alternative is found.

One main take away from the experiments with the physical rovers was that it was way easier to find a tag than to return a tag. In the competition, there are so many tags to collect that at least initially there will be no trouble finding a tag regardless of what search algorithm is used. The trouble is accurately and quickly returning to the nest to drop off the tag. This is a much larger issue since the nest is in a distinct location we have to be able to keep track of the rover's position with respect to the nest. However, with the positional data accruing large amounts of error rather quickly, after finding a tag or two an algorithm relying on the positional data can lose track of the nest. The final algorithm and code developed is heavily influenced by the need to keep track of the nest position without using the positional data.

V. FINAL ALGORITHM (RESULTS)

The FIU Panther Swarm team's swarm robots operate using a search algorithm that does not rely too heavily on the positional data from the rover. Over the course of testing the rovers the main issue that arose was a large amount of drift that would build up in the rover's wheel encoders. After only running for a minute, the rover could think the origin point is over a meter away from where it started. The problem, if left unchecked, would just allow for this error to continue to build, quickly causing issues with returning back to the nest to drop off collected tags.

The solution to this issue is to heavily rely on the data from the rover. The FIU algorithm relies heavily on the compass and ultrasound data. Through testing, they were the most accurate and reliable. The team also use the fact that the Swarmathon competition has walls encompassing the testing field. The four walls and the nest tags were used as landmarks that allow the rovers to switch between one of the six modes previously programmed. There are four main modes that allow the rovers to navigate the field and collect and return tags, an initialization mode, and a recalibration mode that allows the rover to reorient itself within the arena if it is unable to find the nest or encounters and unexpected collision.

The initialization mode (yellow) simply moves the rovers from the center into a position where it can start the main tag seeking modes. This initialization also lets the rovers know what round they are in so they can search the proper amount of area. The four main modes can be broken down as runway (green), position (blue), sweep (orange), and return (red). The runway mode is an east to west path along the x-axis that passes through the nest. The idea for this is like a runway at an airport. The team have the rovers passing though the nest all in the same direction such that they will limit the possibility of coming into close proximity with one another. The position mode assigns a y-coordinate value to the rover and has the rover either moving north or south to get to the desired y-coordinate. The sweep mode is a west to east path that has the rover looking for tags as it moves back across the arena. The return mode has the rover moving north or south to return back to the x-axis so that it can go back to the runway mode.

At the end of the runway mode the program performs a check to see if it has passed by the nest. If it has it continues as planned, but if it missed the nest that means the rover is now where the program thinks it is and it sends it to the recalibration mode. The recalibration mode has the rover, depending on that side of the arena it is on, go to the north or south wall then go to the east wall. By using the ultrasound sensors and the known arena configuration we are able to move the rover into an easily found known position (either the northeast or southeast corner of the arena). With the rover back in a known location we return the rover back to its normal operation.

The algorithm takes a simple principle of having the rovers rake across the arena in search of tags and returning them in a way that limits the possibility of the

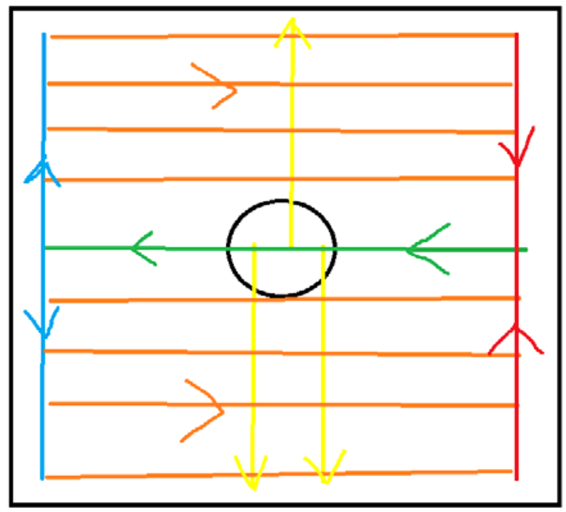

Figure 4: Diagram of Final Search Algorithm

rovers approaching each other head on or in any other way that would interfere with their pathing. This also allows for the rovers to work around the fact that their positional data accumulated large amounts of error. As the only point where the algorithm directly references the positional data is when having it position in the y-direction. But instead of assuming the y-value is constant, we always take the current location value and have it move north or south a relative amount corresponding the desired y-coordinate. This limits the error to a single movement that is at most seven meters in the preliminary round, and should there be an error that is significant enough to have the rover be in a position that is appreciably far away from where the algorithm believes it to be we have the recalibration step. There are algorithms that could find and return the tags in a much faster manner if the positional data was accurate. But this is an algorithm that should work reliably and be able to overcome any tracking errors assuming the compass and sonar data is reliable.

VI. CONCLUSION

To conclude, this paper covers potential ideas and system designs for various opportunities in swarm robotics. FIU team also presented some search algorithms that were explored and the final algorithm developed as a result of the limitations of the rovers discovered though physical testing.

There is a potential to develop an algorithm that is faster and more efficient if the rovers had more accurate positional data. This can be accomplished through better encoders to track movement through the wheels or a more accurate GPS that can provide data accurate enough for the scale that this competition is being performed on. As a result the algorithm submitted has to have a more conservative searching algorithm that can attempt to keep track of its position for as long as possible with a build in recalibration step for if the location of the nest is lost. The result of our conservative pathing and tracking results in the rover being required the full length of the arena twice per search loop, and at maximum performing a path that is half the value of the arena perimeter. This is because the team use the sonar reading in conjunction with the wall to track and reset the positional data of the rover. If the walls were not required to be used as positional landmarks, another algorithm could be developed to cut down on the required distance traveled by each rover. The proposed algorithm can be used and improved for next editions of this Swarmathon competition.

VII. REFERENCES

1. Christian Blum and Xiaodong Li, “Swarm Intelligence in Optimization,” ALBCOM Research Group, Universitat Polit`ecnica de Catalunya, Barcelona, Spain, School of Computer Science and Information Technology RMIT University, Melbourne, Australia.

2. Christopher M. Cianci, Xavier Raemy, Jim Pugh, and Alcherio Martinoli, “Communication in a Swarm of Miniature Robots: The e-Puck as an Educational Tool for Swarm Robotics,” Swarm-Intelligent Systems Group, ÅLEcole Polytechnique FÅLedÅLerale de Lausanne, CH-1015 Lausanne, Switzerland.

3. Erol Sahin, Sertan Girgin, Levent Bayındır, and Ali Emre Turgut; “Swarm Robotics,” KOVAN Research Laboratory Middle East Technical University, Ankara, Turkey; Team SequeL INRIA Futurs Lille, Villeneuve d’Ascq, France.

4. Farshad Arvin, Khairulmizam Samsudin, and Abdul Rahman Ramli; “Development of a Miniature Robot for Swarm Robotic Application,” International Journal of Computer and Electrical Engineering, Vol. 1, No. 4, October, 1793-8163, 2009.

5. Nicholas Hoff, Robert Wood, and Radhika Nagpal. “Distributed colony-level algorithm switching for robot swarm foraging.” Distributed Autonomous Robotic Systems, pp. 417-430, 2013.

6. Joshua P. Hecker, and Melanie E. Moses, Beyond Pheromones: Evolving Error-Tolerant, Flexible, and Scalable Ant-Inspired Robot Swarms, 2015.

7. Lynne E. Parker, “Multiple Mobile Robot System,” Springer-Verlag Berlin Heidelberg, Springer Berlin, Heidelberg, 2008.

8. Mao, Ling, Jiapin Chen, Zhenbo Li, and Dawei Zhang. “Relative Localization Method of Multiple Micro Robots Based on Simple Sensors,” International Journal of Advanced Robotic Systems, 2013, 1. Web. 13 Feb. 2016.

9. Richard, William K., and Stephen M. Majercik. “Swarm-Based Path Creation in Dynamic Environments for Search and Rescue,” Proc. of Genetic and Evolutionary Computation Conference, Philadelphia, PA, USA. N.p., 11 July 2012, Web. 12 Feb. 2016.

10. Shaneyfelt, Ted, Joordens, Matthew A., Nagothu, Kranthimanoj and Jamshidi, Mo, RF Communication between Surface and Underwater Robotic Swarms, WAC'08: Proceedings of the World Automation Congress '08, IEEE, Piscataway, N.J., pp. 1-6, 2008.

11. Shoutao, Li, Lina, Li, Lee, Gordon and Zhang, Hao. “A Hybrid Search Algorithm for Swarm Robots Searching in an Unknown Environment,” 11 Nov. 2014. Web. Feb. 2016.

12. Stolleis, Karl. “The Ant and the Trap: Evolution of Ant-Inspired Obstacle Avoidance in a Multi-Agent Robotic System,” 26 Jun. 2015.

13. Ted Shaneyfelt, Mathew Joordens, Kranthimanoj Nagothu, John Prevost, Anjan Kumar, Mirsaeid Ghazi, Mo Jamshidi, “Control and Simulation of Robotic Swarms in Heterogeneous Environments,” in SMC 2008: Proceedings of 2008 IEEE International Conference on Systems, Man and Cybernetics, IEEE, Piscataway, N.J., pp. 1314-1319, 2008.

Appendix: Team Members

Name	Team	Email
Guerra, Christopher David	1	cguer077@fiu.edu
Mehta, Iti	1	imeht002@fiu.edu
Murton, Brian Anthony	1	bmurt004@fiu.edu
Grahl, Giuliano	2	ggrah004@fiu.edu
Moran, Rudnei	2	rmora112@fiu.edu
Sellan, Julian Christian	2	jsell012@fiu.edu
Billini, William	3	wbill002@fiu.edu
De La O, Frank	3	fdela031@fiu.edu
Mojica, Christopher Josue	3	cmoji002@fiu.edu
Gonzalez, Janesler Louis	4	jgonz725@fiu.edu
Guerrero, Rodolfo	4	rguer054@fiu.edu
Sheffield, Ryan Allen	4	rshef001@fiu.edu
Guerrero, Crystal Emily	5	cguer071@fiu.edu
Sanchez, Kaira Victoria	5	ksanc030@fiu.edu
Walkin, Tomika Ronique Dania	5	twalk011@fiu.edu
Bouzegza, Lyes	6	lbouz001@fiu.edu
Freixas, Abel	6	afrei004@fiu.edu
Moya, Deanna Mariel	6	dmoya003@fiu.edu
Ates, Leandria	7	lates001@fiu.edu
Hernandez, Joseph David	7	jhern110@fiu.edu
Ravelo, Elisa	7	erave001@fiu.edu
Feigelman, Yarden Mario	8	yfeig001@fiu.edu
Rey Brooks, Javier	8	jreyb001@fiu.edu
Reyes, Laura Ines	8	lreye018@fiu.edu
Alonso, Frank	9	falon016@fiu.edu
Maciel, Sebastian	9	smaci009@fiu.edu
Polignan, Nicolas Ariel	9	npoli006@fiu.edu
Porsenna, Carlio	9	cpors001@fiu.edu
Allen Jr, Noel	10	nalle019@fiu.edu
Castellanos Jr, Daniel	10	dcast139@fiu.edu
Solano, Kellby Alexander	10	ksola008@fiu.edu
Benigni, Tomasso	11	tbeni005@fiu.edu
Boga, William	11	wboga001@fiu.edu
Chaviano, Pedro Gabriel	11	pchav017@fiu.edu
Taylor, James Ira	11	jtayl025@fiu.edu
EML 6805		
Larrarte Torres, Jorge Luis	12	jlarr008@fiu.edu
Mugruza, Daniel Cesar	12	dmugr002@fiu.edu
Ramireddy, Arun Gopal Reddy	12	arami217@fiu.edu
Bangalore Somashekar, Suprith	13	sbs001@fiu.edu
Crawford, David	13	dcraw001@fiu.edu
Marchant, William Alexander	14	wmarc002@fiu.edu
Ng, Johnson	14	jng004@fiu.edu
BS/MS STUDENTS		
Hipps, Jonathan	MS	jhipp005@fiu.edu
Jagolinzer, Scott	MS	sjago001@fiu.edu
Wechsler, Adam	UGRAD	awech002@fiu.edu
Zock-Obregon, Hassan	UGRAD	hzock001@fiu.edu

MS student Scott Jagolinzer worked as TA for the Robot Design class, and coordinated the teams. MS student Jonathan Hipps concentrated on swarm robot controllers.

Each of the 14 teams in EML 4840 and EML 6805 has worked on the following team projects independently:

(1) Survey of swarm controllers and algorithms

(2) Modification of mobilityStateMachine, obstacleHandler, and sonarHandler functions

(3) Development of final algorithms

(4) Final code development and software simulations

(5) Testing of code on physical robots

Additionally, each team prepared a PowerPoint presentation and presented their final work on swarm controller survey and algorithms.

The final team algorithm and code was based on the integration of most successful elements of various team codes, which was decided collectively in the class.

Adam Wechsler and Hassan Zock-Obregon independently supported the project in the lab as needed.

Several teams helped to give guided tours in the Robotics Lab as part of the outreach effort.

Laura Reyes and Carlio Porsenna surveyed and prepared documents on NetLogo platform.

Introduction to NetLogo video clip is posted on YouTube: https://youtu.be/a_szxQ4ZW-w

FIU Panther Swarm team experience is summarized in a YouTube video: https://youtu.be/gq2Nkrp3MQc

Scott Jagolinzer, Jonathan Hipps, Jorge Larrarte and Rodolfo Guerrero prepared a technical paper to be submitted to FCRAR 2016 for presentation on May 12-13, 2016.

Jonathan Hipps and Scott Jagolinzer submitted a paper on swarm controllers to be presented at FCRAR 2016 on May 12-13, 2016.