



## Design of a Swarm Search Algorithm: DustySWARM SpiralEpicycloidal Wave (SEW) Code for NASA Swarmathon

Tariq H. Tashtoush, Oscar Gutierrez, Esteban Herrera, Juan Medina, Abraham Peña, Edgar Varela, Roger Hernandez

Texas A&M International University, Laredo, Texas, USA

**\*Corresponding Author:** Tariq H. Tashtoush, Texas A&M International University, Laredo, Texas, USA

**Abstract:** This paper focuses on the design and implementation of a search algorithm that the DustySWARM team will use in the NASA Swarmathon 2017 physical competition. The rovers, or “swarmies”, provided by NASA must be able to communicate with each other while searching autonomously for resources in a simulated environment. Some of the methods implemented during this project were the square spiral path, spiral path, and the Epicycloidal wave path. The results indicated that the most efficient method was the Epicycloidal wave path as it collected more resources within the allowed time limit. This paper will summarize the design and code development process.

**Keywords:** Swarm Robotics, Searching Algorithm, Autonomous, Robot Swarm, Robot Operating System (ROS), NASA Space Exploration, Mars Mining, Simulation, Autonomous Robot Swarm; Collaborative robots

**Public Interest Statement:** Robots had been integrated into almost every field including industrial automation, medical and space exploration. NASA had been working hard to explore the space and discover new resources on the Moon and Mars. One of the going-on projects with NASA is to send a swarm of small robots to Mars, where these robots will explore and collect resources for analysis, this option is affordable and safer. Our team works on developing algorithms that will be implemented on these rovers, to guide them autonomously to search effectively and efficiently while avoiding obstacles and map their environment. This will assist in advancing NASA space exploration technologies.

### 1. INTRODUCTION

The moment the human race landed on the moon, people from different nations have tried to expand the frontiers of where it is possible to go. This is the reason why the first robot or Mars Pathfinder's Sojourner Rover as it was known, rolled for the first time in 1997 with the expectations of discovering more about Mars. Additionally, many organizations are pursuing the creation of improved rover versions following the idea to explore mars. However, this does not mean that any rover will be perfect for the task of exploring Mars as seen in previous versions.

The Spirit rover got stuck and could not take any more pictures and information from the surface of Mars in 2009. The rover had to be shutdown, as its transmission would contain one location from that point forward while its other twin unit would keep working on the other side of Mars. This would lead to this next question, why are we working on smaller versions, and what is expected to be the outcome of this project?

The answer is quite simple, the purpose behind the use of these rovers, or swarmies as they are called, is to find a program that could make them work more efficiently. The main idea behind this project is to improve the process in which the rovers would provide good information and pictures to a central station on Earth about Mars and how should it be approached if the human race decides to travel farther away than it has before.

The National Aeronautics and Space Administration (NASA) is a governmental agency dedicated to space exploration; they started the Swarmathon Competition in 2016 where they invited different universities from across the country to work on creating a robotics system that mimics ant's behavior. Each university in the physical competition will be provided with three small robots called “swarmies” and they should search, find, and gather resources from a sounding simulated environment while communicating among each other and working in autonomous configuration, similar to ants.

TAMU DustySWARM team is participating in the second year of this competition. In 2016, DustySWARM 1.0 team placed the third in the virtual competition. For the 2017 NASA Swarmathon, DustySWARM was invited to participate in physical competitions. The new DustySWARM members are in charge of developing a new code that will be used to control the swarmies and achieve the goal of collecting the most targets within the time limit. DustySWARM team 2.0 members developed the Spiral Epicycloidal Wave (SEW) path and it is illustrated in this paper. The methods and designs followed to reach such a search algorithm are described in detail in this paper. The results demonstrated the efficiency of the selected method and its score toward the final goal.

The winner of this competition will be selected according to the number of resources gathered. NASA will consider all universities participating in this event to gather ideas to implement when sending robots to the Martian planet. Overall, the NASA Swarmathon Competition is a great opportunity to demonstrate teamwork, outreach, planning, and coding skills.

The paper is organized as follows: Section 2 is a background and literature review, Section 3 describes the model and algorithm development, Section 4 deals with the analysis and development of the experimental simulation runs; Section 5 shows the results of the proposed code compared to the original and alternatives, and Section 6 concludes the paper and describe the team future plan.

## **2. LITERATURE REVIEW**

As mentioned before, the attempts made by NASA in order to explore the surface of Mars have been innovative and continue to improve to collect data more accurately. The range of problems that this exploration faces start with the fact that the scenarios have to be simulated. Since humanity is still pending to be physically present on Mars, there is limited information about the conditions in which the rovers have to work with; therefore, creating rovers that are suited for the environment of the planet is an existing challenge. With this information in mind, how does one go about designing rovers suited for such unknown conditions? In this section of the paper, it is discussed the different aspects that are implemented into a rover to conduct research on Mars. Some of the past rovers sent to Mars will be mentioned and how their work improves the parameters needed to design optimized rovers for a future to come.

One of the main functions of the rovers is to send information back about Mars and how its surface is composed. The way NASA communicates with the previous rovers was through radio signals. The problem with communicating with the rovers using this method is the delay from the starting point and the receiving point [1]. Because the rovers did not have an autonomous program, the rovers were not capable of communicating between them or carry-on with the exploration task without the need for human intervention. This is a strong problem because there is a limited amount of periods to communicate with the rovers due to the rotation of the planets and the possibility of the Earth or moon blocking the signal completely. The time it takes for a signal to get back could be better utilized if the rovers had a system in which they could independently explore and continue further research. NASA has done several designs in order to send a new, more optimized version that would condone a more efficient job.

The Spirit and Opportunity rovers are the well know rovers that landed on Mars for data collection. There are limited ways to determine the rover's initial position once it lands and this is an important step for the initialization of the exploration. As Guan mentioned in the article "An innovative high accuracy autonomous navigation method for the Mars rovers", some of the initial position localization methods are radio-ranging technique, landmark-based autonomous position, and attitude determination method. Guan explains, "The main techniques for the rover's autonomous navigation include inertial navigation, vision navigation and celestial navigation" [1]. These methods increase the amount of time spent on exploration and also the amount of information gathered about the surface of the red planet yet, none of these methods (some that are implemented by the Spirit and Opportunity rovers) completely qualify as for autonomous systems due to their individual drawbacks.

A Strap-down Inertial Navigation System (SINS) was implemented in order to increase the efficiency of the autonomous system in the rovers. The results are shown in Figure (1) which indicate the improvements in different areas of the rovers while using the SINS method as it can be seen from the

tables, the error margins from each scenario are significantly improved by the autonomous navigation system.

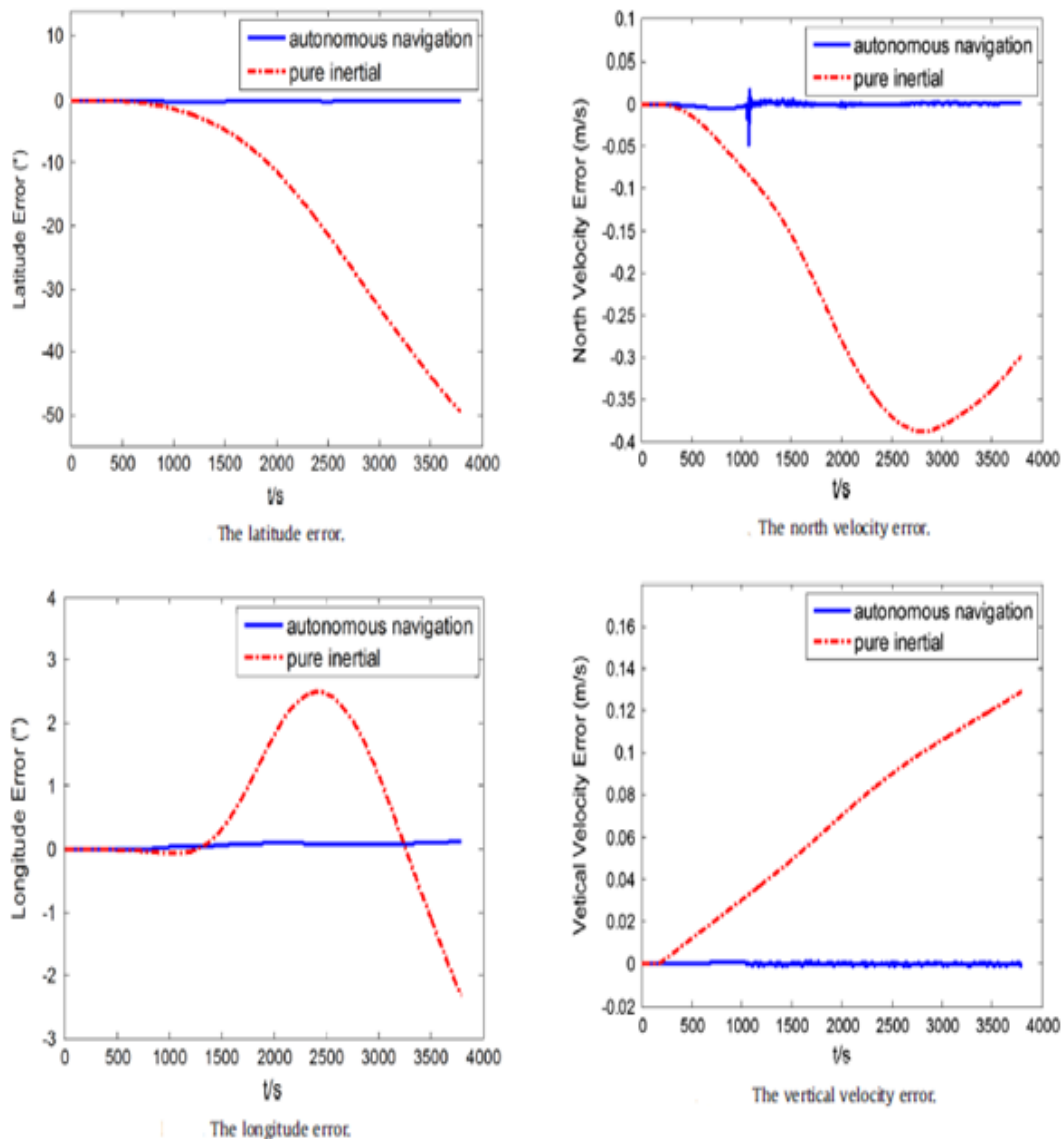


Figure 1. Improvements in rover navigation system [1]

### 2.1. Original iAnt Code and Return Logic

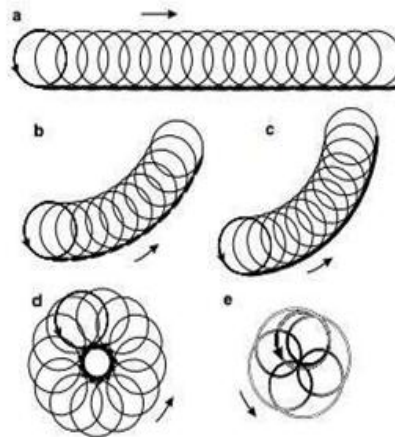
The original iAnt Code developed by the University of New Mexico (UNM) will be used as the base for the rovers search and collect approach. DustySWARM 1.0 modified the original iAnt Code by adding a “return logic”, that allowed the robots to return to the position where they had found a resource after dropping it off into the collection zone. By implementing the “return logic” code, the team noticed an increase in the number of collected tags [2].

Based on DustySWARM 1.0 code changes, our current team decided to modify different sections of the code. Some changes include SearchController.cpp and random generator factors, which resulted in collecting more resources.

### 2.2. Dynamics of Rigidly Rotating Spirals Under Periodic Modulation of Excitability

DustySWARM 2.0 builds their code based on the design of the Epicycloidal wave seen in Figure 2 (d). This wave had been designed and mathematically modeled based on the Belousov-Zhabotinsky (BZ) reactor. Kantrasiriet.al. conducted two experiments: experimental observation and simulation. Experimental observations consisted of exposing the BZ reactor to various light intensities and pulsatory modulations (short light pulses), while the simulation was performed using

numerous equations within the Oregonator model to output the desired results [3]. The equations involved within the simulation aided the DustySWARM team in the development of their Spiral Epicycloidal Wave (SEW) Code.



**Figure2.** (a) – (d) are the results of the experimental observations using pulsatory modulations. (e) is the result of using sinusoidal modulations within the simulation experiment. The spiral waves resemble hypocycloidal and Epicycloidal figures. [2]

### 3. METHODOLOGY & ALGORITHM DEVELOPMENT

DustySWARM 2.0 team spent the time to be familiarized with both the UNM code and DustySWARM 1.0 code. Because the objective of the swarmies is to collect as many resources as possible in an efficient manner, the team researched multiple possible patterns that would satisfy that goal. By the end of the research, the team decided to experiment in two possible paths, namely a square spiral and circular spiral, which eventually led to a new path pattern that is currently being used and modified

#### 3.1. Square Spiral Path

The first path algorithm was developed based on the idea of increasing the search area of the swarmies exponentially would increase the number of objectives recovered at the end of the simulation. The decision of making a square approach of the spiral was made by the belief of making a simpler code and, at the beginning of the project; dealing with cosine and sine functions (covered in a traditional spiral) seemed too complicated. To implement this path, the team needed a dependent variable ‘X’ that would increase each time a task was finished. The task was separated into two sub-task that consisted on a displacement of ‘X’ on the x-axis followed by a 90 degrees turn, and a displacement of ‘X’ on the y-axis ‘followed by a 90 degrees turn. After a task was finished, the team had a variable ‘P’ (with an initial value of -1) that would change its sign in order to determine the polarity of the displacement of both axes as shown in Figure (3).



**Figure3.** Represents the previous example with a negative displacement in red and the positive displacements in blue. Eventually implementing these changes into the code provided by NASA was harder than it seemed and the square spiral path was dropped

Example: the team wants a spiral that has an initial negative displacement on both axes of 10cm each and after its first turns it should follow a uniform path, this means that after two turns, the ‘X’ variable

should increase its value by 10cm and change its initial polarity to positive, and so on following the same pattern. Technically making a generalized function:

- $X = X + 0.1m$  (after each task)
- $P = P * (-1)$  (after each task)

A task is represented as followed:

- Displacement in x-axis =  $\text{current\_position.x} + X * (P)$  with Turn = 1.57 radians
- Displacement in y-axis =  $\text{current\_position.y} + X * (P)$  with Turn = 1.57 radians

### 3.2. Spiral Path

After further analyzing the code, and playing with different values, the team managed to understand the necessary steps needed to create a spiral pattern. Changing the goal objective functions in the Searchcontroller.cpp allowed the team to aim toward the completion of the spiral path in a different approach. How the spiral was meant to be achieved, was by imitating its behavior rather than replicating it. This means that instead of making a perfect spiral across the field, the swarmies would replicate a spiral based on translations and angle changes between each goal objective. This would allow a small translation of 'X' followed by a change in the angle of 'θ' to represent the Spiral Path gradually as shown in Figure (4).



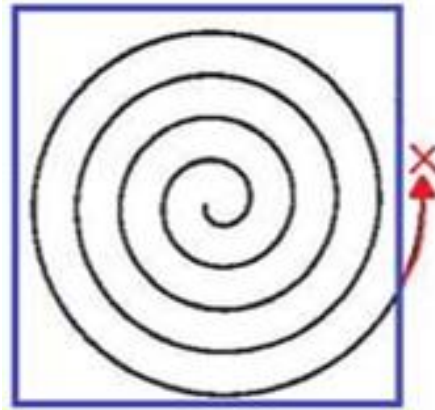
**Figure 4.** Illustration of the square spiral path as the search

Nonetheless, by continuing to analyze the spiral method, the team concluded that eventually one of the goal locations would lay outside the perimeter of the field as seen in Figure (5) making the rovers get inside a loop of following an unreachable goal blocked by an obstacle (in this case the walls).

While looking for a solution to this problem, one of the changes inside the values of the SearchController.cpp completely changed how the pattern of the spiral behaved, and to the team, this new pattern worked for the best of the project.

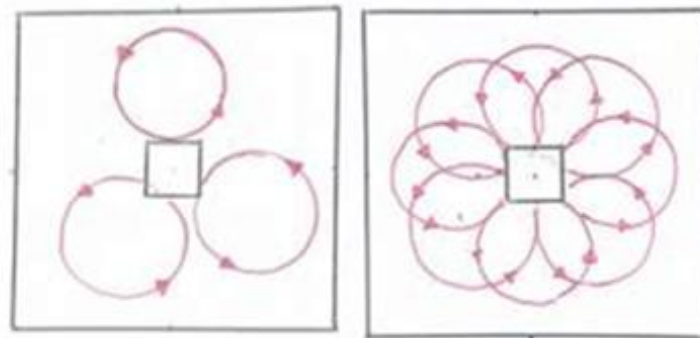
### 3.3. Epicycloidal/Spring Wave Path:

As mentioned in the previous section, this path was achieved by updating the SearchController.cpp and changing the goalLocation.theta to be equal to the currentLocation.theta plus a random Gaussian distribution of 9 to 12 degrees. By utilizing the theta into the provided values of goalLocation.x and goalLocation.y, the team managed to develop a loop path that originates from the center of the map towards the boundaries.



**Figure5.** *Limits of the spiral path*

One of the positive aspects of this path is that after a swarmie is presented with an obstacle or an objective, the center of that loop will shift its center allowing the swarmie to cover more ground in the field. Figure (6) shows the origin loop for the three swarmies, followed by a representation of a loop with a shifting center.



**Figure6.** *Illustration of the Epicycloidal/Spring Wave path as the search algorithm for DustySWARM 2.0*

The Spiral Epicycloidal/Spring Wave Path allows an uninterrupted search as none of the goal locations will end outside the field, but with the sacrifice of leaving possible objectives left behind in the corners. The size of the loops can be altered by modifying the angle and distance traveled after each goal location is achieved. After weighing all pros and cons, this path was chosen by the team, as it will provide the safest path while collecting the most resources.

#### 4. EXPERIMENT SIMULATION RUNS

The experiments were conducted using the simulation provided by NASA. The team decided if the Spiral Epicycloidal Wave path was going to be used for the competition, several simulation runs must be made to determine its functionality. The first initial runs of the simulation with the Spiral Epicycloidal Wave path demonstrated fewer simulation crashes and malfunction of the swarmies compared to other paths, making the decision of taking this path as our main development easier. While conducting the simulations, the team stumbled upon several problems that caused variations in the collected data. These problems included inaccessible goal locations, rovers going inside the center, drop-offs made outside the center, objectives being stuck within the swarmie's claw, and other issues. All simulation runs had at least one or two of the mentioned problems preventing the affected swarmie from collecting targets, Figure (7) illustrates one of the problems mentioned before.



**Figure7.** Issues happened while running the developed code in the simulation system

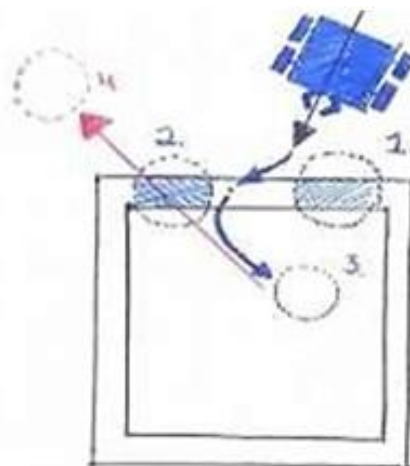
Nonetheless, the team ran the simulations with the disabled swarmies and still managed to collect a total of 19-30 objects of 100 within the 30 minutes. The purpose of running simulations of 100 objects was to let the swarmies collect objectives and still let the team view if the path taken was functioning the way it should.

Once the decision of taking the Spiral Epicycloidal Wave path was set in stone, the team worked on fixing the issues that prevented the rovers from functioning the way they should. These changes were mainly addressed by changing DropOffController.cpp, PickupController.cpp and Mobility.cpp files. Most of the problems occurred when a swarmie is going to drop an object at the center of the map (home position), or trying to collect the objects. Each change was analyzed in the simulation to determine whether it benefited the code or not. This trial and error approach allowed the team to be more familiarized with the code and make further changes.

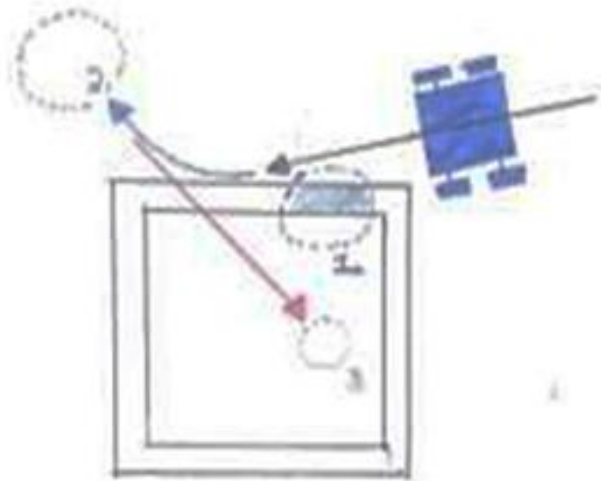
## 5. RESULTS SUMMARY

The first change to solve an issue with the rovers missing the targets when attempting a pickup. By increasing the value of the result.cmdVel from 0.18m to 0.2m when a target was still not locked, the rovers managed to get the extra push needed to successfully pickup target 90% of the time. The team also deleted the negative movement of backing up after completing a successful pickup. Although the previous change benefited the code, they did not solve the problem entirely.

In Figure (8 and 9), the team illustrated two cases representing two possibilities of a swarmie coming into the center attempting a drop-off. It is important to take into consideration how the code behaves in order to detect the center, as this part of the code is critical for the swarmie's functionality.



**Figure8.** Successful target drop-off



**Figure 9.** *Error when target dropping-off*

Three statements can affect how the swarmies move, and they are right, left, and `seenEnoughCenterTags`. There is also an extra variable called `TurnDirection` that is dependent on `seenEnoughCenterTags` being true. If that is the case, the value will change from a positive one to a negative one. If `seenEnoughCenterTags` is false, variable right will execute a negative displacement of 10cm and an angle turning of 0.2 radians allowing the swarmie to position itself towards the center.

Similar behavior goes for the variable left but with a negative angle. If the statement of `seenEnoughCenterTags` is true, the negative 1 will affect both the direction and the angle of the right and left variables, changing right to a positive displacement turning left, and vice versa.

It is also worth mentioning that `seenEnoughCenterTags` will allow the program to determine if there are too many center tags on the left or right. If the value of right minus five is greater than left, the value of left will be false making allowing a right turn, and the reverse scenario happens if there are too many in the right. Figure (8) illustrates a successful drop-off once it sees enough tags in the left (1) then enough tags on the right (2) followed by going inside to drop-off (3) and back up (4). Yet if the angle in which the swarmie is coming in is too acute, the rover will execute only one of the turns making a drop-off outside the center and eventually getting inside the center due to the reversing; leaving the swarmie inside and pushing all collected targets outside the center as seen in Figure (9).

The further modification included changes to the `DropOffController.cpp` by adding 0.1 radians more to the angle turned when seeing a tag in the left or right. A variable by the name of `anglediv` was created in order to decrease the value of the previous turn by 0.2 radians. At the beginning of the Epicycloidal/Spiral wave pattern, the team has a constant increase of 10 degrees in theta per each goal location completed. The following figures are the result of how the swarmies behave when there are no targets, coalition; with other swarmies existence, the center of the loop will be shifted which will make the swarmies cover more terrain.

Figure (11) represents the simulation run for 30 min with targets in the field, the team can observe that the path changes drastically as the center of the loop is shifted for each objective and obstacle encountered. At the end of the simulation, the rovers collected 24 targets and decided that adding the Gaussian distribution of 9-12 degrees in theta will prevent the rovers from staying in a perfect circle in case of no targets encountered.



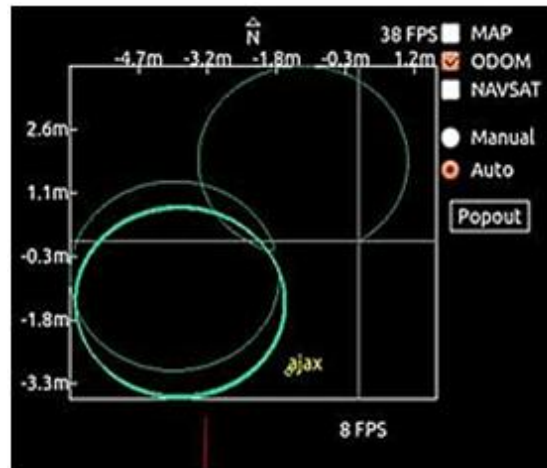


Figure10. Terrain explored by swarmies when obstacle avoidance

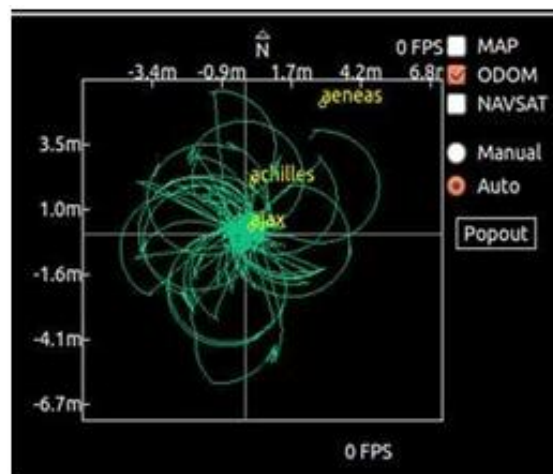


Figure11. Terrain explored by swarmies during simulation

## 6. CONCLUSION

In conclusion, the development of a searching algorithm allows the team to understand the range of possibilities in which the swarmies can behave. Because there is still a significant amount of randomness within the simulation as of now, the team is still working on optimizing the code in order to increase efficiency. When it comes to understanding the code, there has not been a more reliable method than simple trial and error, as it takes independent changes from independent members to understand how that modification altered the code. Eventually, if a change is noticeable, the team learns about it and can refine it in the future if needed.

DustySWARM team 2.0 with their Spiral Epicycloidal Wave (SEW) search algorithm placed the Sixth in the physical competition and third place in the Technical Report competition among nineteen (19) participating teams from all over the USA.

## ACKNOWLEDGMENT

Thanks to NASA and the University of New Mexico (UNM) teams for all the help and the opportunity to participate in such great competition. Thanks to all our sponsors from Texas A&M International University (TAMIU) and Laredo.

## REFERENCES

- [1] Guan, X., Wang, X., Fang, J., & Feng, S. (2014). An innovative high accuracy autonomous navigation method for the Mars rovers. *Acta Astronautica*, 104(1), 266-275.
- [2] Hernandez, R., Yanez, R., Gonzalez, J., Moreno, H., Escobar, V., & Tashtoush, T., "Design of a Swarm Search Algorithm: DustySWARM Reverse-Twister Code for NASA Swarmathon." Texas A&M International University, School of Engineering (2016).

- [3] Kantrasiri, Supichai, PramoteJirakanjana, and On-Uma Kheowan. "Dynamics of rigidly rotating spirals under periodic modulation of excitability." *Chemical physics letters* 416.4 (2005): 364-369.
- [4] W. E. Shots, "The Linux Command Line", 2nd ed., San Francisco, CA: Creative Commons, 2013.
- [5] J. M. O’Kane, "A Gentle Introduction to ROS", Columbia, SC: University of South Carolina, 2014.
- [6] <http://nasaswarmathon.com/>

#### **AUTHOR'S BIOGRAPHY**



**Dr. T. Tashtoushis** is an Assistant Professor at the School of Engineering in Texas A&M International University (TAMIU), Laredo, TX. He got his Ph.D. and M.S. degrees in Systems and Industrial Engineering from State University of New York (SUNY) at Binghamton on 2013 and 2009, respectively and his B.S. in Mechatronics - Mechanical Engineering from Jordan University of Science and Technology (JUST), Irbid, Jordan on 2005. He is a multi-discipline engineer, who has industrial and academic experience in the field of Systems Simulation and Design, Production Quality and Management, Lean Manufacturing, Six-Sigma Processes, Industrial Robotics and Automation, Exploration and Autonomous Robotics, Artificial Intelligent (AI), Computer Integrated Manufacturing, 3D Printing Processes, Engineering Statistical Analysis, Project Management, Optimization, Instruments and Electrical Devices, Reliability, Healthcare Systems, Nano-Technology and Energy Harvesting, and Human Factors/Egomaniacs Studies.

**Citation:** *Tariq H. Tashtoushet.al., (2020). "Design of a Swarm Search Algorithm: DustySWARM SpiralEpicycloidal Wave (SEW) Code for NASA Swarmathon", International Journal of Research Studies in Computer Science and Engineering (IJRSCSE), 7(1), pp.13-21.DOI:http://dx.doi.org/10.20431/2349-4859.0701002*

**Copyright:** © 2020 Authors, This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.