Project Proposal: Geometric flight formations using Swarm Drones

Aneesh Chodisetty
M. Eng. Robotics
University of Maryland
College Park, MD, 20740
Email: aneeshch@umd.edu

Bhargav Kumar Soothram
M. Eng. Robotics
University of Maryland
College Park, MD, 20740
Email: bsoothra@umd.edu

Joseph Katakam
M. Eng. Robotics
University of Maryland
College Park, MD, 20740
Email: jkatak73@umd.edu

Overview:

An increased usage of Unmanned Aerial and Surface Vehicles (UAVs and USVs) has been witnessed for various autonomous aerial applications in recent years. A swarm of these unmanned autonomous vehicles can perform safe collision-free navigation by adjusting the positions of drones using controlled localization and formation. In this proposal, our team has focused on implementing one of the trending applications of swarm drones, which is to form different geometric shapes using a swarm of 20 or more drones.

Simply put, a bunch of drones form geometric shapes such as a circle or rectangle midair autonomously without any collisions amongst themselves. In order to achieve this, state-of-art path-planning swarm algorithms will be deployed. The performance validation of this project will be done using the Gazebo simulation depicting a real-time demo of the application. Acme can then utilize this package in its 5-year robotics-based product roadmap.

Our team will be using '*Kamikaze Drones*' as the project code name to help protect Acme's secret product plans. The name means 'Divine Wind' in Japanese and is inspired from the '*Kamikaze swarm operation*' by company 'STM'. The GitHub repository for this project can be found *here*.

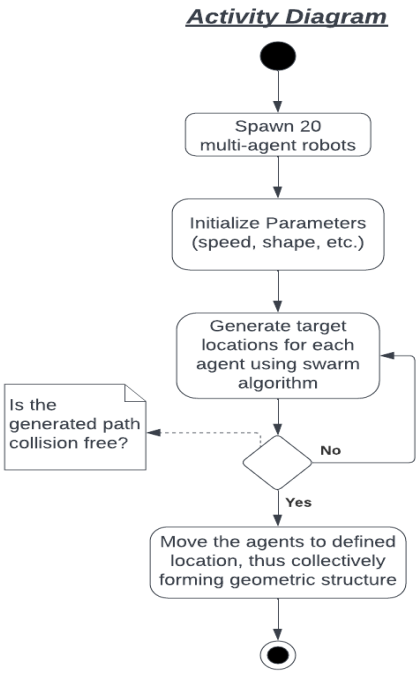

Fig. 1. Flow of our proposed system

System Design:

Considering the complexity of the project to be developed, our team will be developing 3 systems in a progressive manner in order to achieve the desired objective. These three systems are as follows.

Minimal System: To kick things off, in this system we will be using turtle Bots with simplified sensing aspects, where each agent broadcasts its location using GPS. Then using swarm algorithm called 'boids', path planning will be implemented.

Target System: Next, developing on the previous system, the turtle Bot agents will be replaced by drones using 'adaptive swarm' path planning algorithm to make the drones make simple geometric formation such as Circle or a rectangle without any collisions.

Dream System: Finally, the dream system would be to improvise the 'Target system' such that complicated structures can be dynamically formed using drones such as portraits using structure from motion.

Note that the team's goal is to achieve the 'Target System'. If time is available and things go as planned, then our team would try achieving the 'Dream System'.

Potential Risks and Mitigation:

- The entire project is memory and power expensive. A powerful configuration of desktop is required to handle the processing requirement of simulating 20 drones in gazebo environment without crashing. Seeing that the prices of RAM have been coming down over the years, the solution can be a simple inexpensive purchase. Another alternative could be to remove any unnecessary components from the gazebo world whose only purpose is to serve as aesthetic objects in background. This way less memory is consumed, and it remains fast.
- The drone formation's accuracy might be low when forming complicated structures midair. Sometimes there is a change of getting a 'singularity' in the planned path. Therefore, the application of this module has to factor this in, before deploying it in real-time.
- Various aerodynamic factors such as wind speed or environmental factors such as rain as not considered when simulating this project. Therefore, the real-time execution of this product should tweak a few parameters to adapt to these changes.

Design And Development:

Agile software development model will be used for the development process where tasks will be tracked using a backlog table. The software is designed in a Test-Driven Development fashion and implemented using Pair programming technique. The tasks will be outlined for every sprint and after each sprint, the roles of the pair-programming group will be interchanged.

Technical Process:

'C++ 14' Language will be used to develop the project using the 'Visual Studio Code' environment. To perform simulations and create working code modules, the open-source framework 'ROS 2' is used, and the entire project is built using COLCON which is an iteration on the ROS build tools 'CATKIN_MAKE' running CMAKE (3.16.3).

Before the project is released, the module will go through an extensive check for bugs, coding style, documentation, and working version. Unit tests will be created to check the proper functionality of the code and complete code coverage will be achieved. Documentation will be done using Doxygen and maintained for each component, with proper code commenting.

Deliverables:

1. Proposal Documentation
2. UML Diagrams
3. Project Package with demonstrated OOPs and ROS concepts
4. CI using Travis
5. Code Coverage using Coveralls
6. Unit Tests using Google Test Framework
7. Developer Level Documentation
8. Static code analysis with cppcheck
9. Google C++ Style guide with cpplint validation