

# PX4-Aerial-Swarm-Reconstruction

## ENPM700 Final Project

Kshitij Aggarwal  
Apoorv Thapliyal  
Mohammed Munawwar

---

**[Date: 13 November 2024]**

The project aims to implement a system where multiple drones work together to capture data from an environment and reconstruct a 3D model of the scene. Each drone will be equipped with LiDAR sensors to capture high-resolution data, which will be processed collaboratively to create a 3D reconstruction.

---

### Scope:

Scope of our project is to implement a collaborative multi-drone system where each drone independently captures data from different parts of an environment that will be processed collaboratively to generate a unified 3D reconstruction. The entire pipeline will be simulated in Gazebo with a similar world and scenario.

---

### Initial Design:

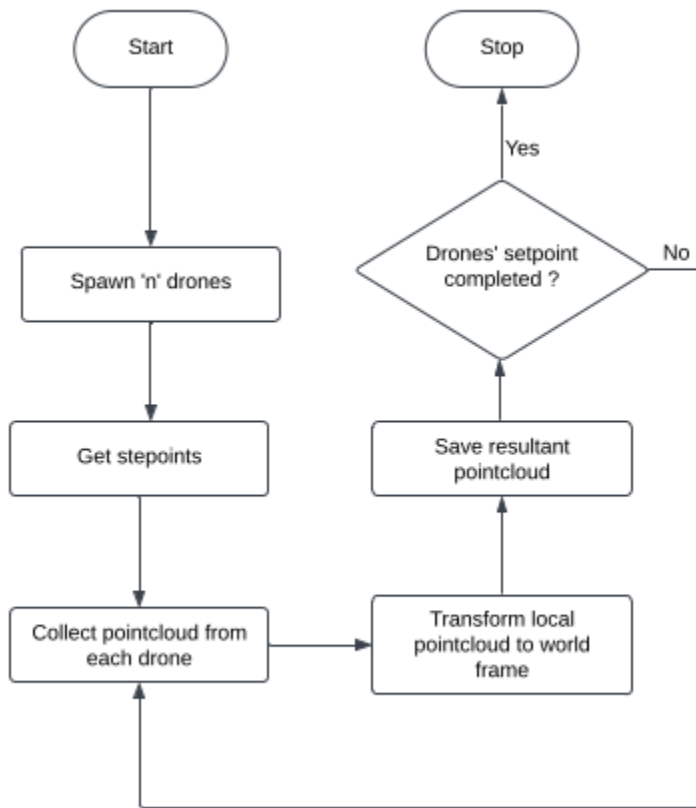
Our proposed pipeline consists of four primary subtasks:

1. **Data Capture:** Each drone will be equipped with LiDAR sensors to capture depth Data.
2. **Coordination:** Each drone will be given a pose on the map and the PX4-autopilot will be used for the controls of the drones motion, ensuring that drones cover different areas without overlap.
3. **Data Fusion:** Data from all drones will be merged into a single dataset for 3D reconstruction.
4. **Reconstruction:** A 3D model of the environment will be generated using point cloud processing techniques.

The Activity diagram below shows the basic outline and pipeline of our proposed algorithm.

---

### Algorithm:



---

### Design Process:

#### System Setup:

- Set up a Gazebo simulation environment with multiple drones using URDF models.
- Integrate LiDAR sensors into each drone model for environmental perception.
- Configure PX4 Autopilot for each drone to handle flight dynamics and motion Control.

#### Data Capture Implementation:

- Assign each drone a specific region or waypoint to avoid overlap in data capture, ensuring efficient area coverage.
- Implement autonomous navigation using PX4 Autopilot to allow drones to explore different regions of the environment autonomously.
- Capture depth data using LiDAR sensors.

### **Data Fusion and Reconstruction:**

- Develop algorithms to merge individual datasets from each drone into a unified point cloud using Open3D.
- Use point cloud processing techniques provided by Open3D to generate a final 3D model of the environment.
- Implement sensor fusion techniques to correct any discrepancies between datasets before merging them into a single point cloud.

### **Testing and Validation:**

- Test the system in various simulated environments within Gazebo, including different terrain types and obstacle configurations, to validate performance under varying conditions.
- Validate the accuracy of the 3D reconstruction by comparing it against known ground truth models in the simulation.

### **Optimization:**

- Optimize inter-drone communication using ROS2's DDS-based communication system with appropriate Quality of Service (QoS) settings to ensure real-time collaboration without network congestion or delays.
- Fine-tune navigation parameters in PX4 Autopilot to improve flight efficiency and reduce energy consumption during exploration.

---

### **Design Choices:**

#### **[Date: 16 November 2024]**

Munawwar: Issues with spawning custom world with PX4 SITL, SITL unable to find Gazebo worlds so topics are not being generated

Found solution: Keep custom world in worlds folder of PX4-Autopilot

#### **[Date: 17 November 2024]**

Munawwar: Drone camera topics unavailable, some issue with ROS GZ Bridge, asked on Discord for guidance

#### **[Date: 18 November 2024]**

Munawwar: Fixing SDF files, seems like it was built for ROS1 but the models were not ported for ROS2 dependencies

#### **[Date: 22 November 2024]**

Munawwar: After editing camera topics in the Models file directly, we are able to get topics, however it still doesn't work for multiple drones

**[Date: 23 November 2024]**

Munawwar: Pointcloud available in RViz, works well and not noisy, however still issues with using for multiple drones

**[Date: 24 November 2024]**

A namespace issue occurs, spawning for multiple drones needs the name of the plugin to keep changing, however the name stays static which is causing problems

**[Date: 25 November 2024]**

Still struggling with spawning multiple drones with the camera working. Standard swarm drone is spawning well. However the camera topic is not working.

Work will now be split: Munawwar would work on the camera gazebo topics, Apoorv would try to arm the drones in simulation to get some sort of control working

**[Date: 26 November 2024]**

Topics are not showing up, MicroXRCE agent is not connecting, the reason is unknown so far. Both Apoorv and Munawwar are not able to echo topics, which has stunted all progress

**[Date: 27 November 2024]**

Issue still persists, PX4-Autopilot and MicroXRCE Agent was rebuilt but this problem still exists. ROS\_LOCALHOST\_ONLY was set to 1, this however did not solve the problem.

**[Date: 28 November 2024]**

This problem was posted to PX4-Autopilot discord, however no official solution was found.

This was the comment posted to Discord:

*Hey,*

*Me and my friend were using the PX4 autopilot with ROS2 for a project, and initially we were getting each others topics when running `ros2 topic list`, so we then added `ROS\_LOCALHOST\_ONLY=1` in our zshrc files. But recently, our MicroXRCE agent is giving small issues:*

*Running `make px4\_sitl gazebo-classic`, the MicroXRCE agent connects, but running `ros2 topic list` does not give any topics*

*Running `./Tools/simulation/gazebo-classic/sitl\_multiple\_run.sh -m iris -n 1`, the MicroXRCE agent simply does not connect, and so we dont get topics*

*May I know any possible reason for this problem? Thanks*

Similar question was posted to PX4 github issues:

<https://github.com/PX4/PX4-Autopilot/issues/24053>

### [Date: 28 November 2024]

For a completely unknown reason, Munawwars topics are being created by the MicroXRCE agent but running ros2 topic list returns no topics. Somehow even camera topics are unavailable.

### [Date: 29 November 2024]

With some settings changes:

# set this

```
export PX4_SIM_HOST=127.0.0.1
```

# do not set this

```
export PX4_SIM_PORT=
```

# If using Fast DDS:

```
export RMW_IMPLEMENTATION=rmw_fastrtps_cpp
```

Due to space issues on Apoorvs laptop, he has committed his progress to dockerhub:

<https://hub.docker.com/r/apoorv1009/px4-swarm-city-mapper>

Munawwar is able to get topics. For now, the setup is divided as so:

Simulation: running on Munawwars laptop

Controller and ROS2 nodes: running on Apoorvs laptop

This is also due to the weird situation where Munawwars topics are being published to Apoorvs laptop, again for **unknown reasons**.

### [Date: 30 November 2024]

Task 2.01 was completed and pushed to github. Started Task 2.03, Task 2.02 is being done parallelly by Munawwar.

### [Date: 1 December 2024]

Task 2.03 was completed, drones are arming and switching to offboard mode. Task 2.04: Make all drones takeoff and hover at 5m, has been started.

### [Date: 2 December 2024]

Task 2.04: Make all drones takeoff and hover at 5m completed. Few things must be noted:

- When arming with multiple drones the *target\_system* values of the **px4\_msgs::msg::VehicleCommand** message type must be set to **0** or **ID** number of the drone. Else all messages will be ignored
  - This small change in code fixed all the issues with regards to arming the drones
  - Read here for more information: [Link](#)
- The arm and offboard mode command may need to be sent multiple times before the drone actually arms and takes off. For our code, the same arm and offboard command was sent 5 times.

Task 2.05: Provide setpoints for each drone, has been started.

**[Date: 3 December 2024]**

Task 2.05: Provide setpoints for each drone, completed. The drones spawn and takeoff to custom heights and positions. Task 2.06: Conduct sweeps of the simulated city has also been completed. Once we were able to arm the drones, we set a config.yaml to define the trajectory of the swarm for a cleaner and more elegant operation.

**[Date: 4 December 2024]**

Task 2.07: Write L1 unit tests for the control class, was completed, working on adding L2 tests and would hope to increase coverage too by tonight.

Slight mishaps with the coverage, despite tests passing, there is a 0% coverage. Probably something we overlooked. However this would be fixed in the Task 2.09: Adjust tests to increase code coverage. For now Task 2.08: Write L2 tests for the px4\_swarm\_controller package is the focus.

**[Date: 5 December 2024]**

Task 2.07: Added more L1 tests to improve the code coverage. Increased the coverage from 77% to 84%. Task 2.08: Facing issue with the file structure while running the Level 2 tests.

**[Date: 6 December 2024]**

Task 2.08: Modified the CMakeLists.txt for Level2 test to run. Deleted the CMakeLists.txt in the test folder. Now the Level2 tests are running.

**[Date: 7 December 2024]**

Task 2.08: Wrote more Level2 tests that check the publishers and subscribers. Now both the tests are running and we are getting coverage report.

Task 2.1: Updated the README file with the bash lines to run the tests and view the report.

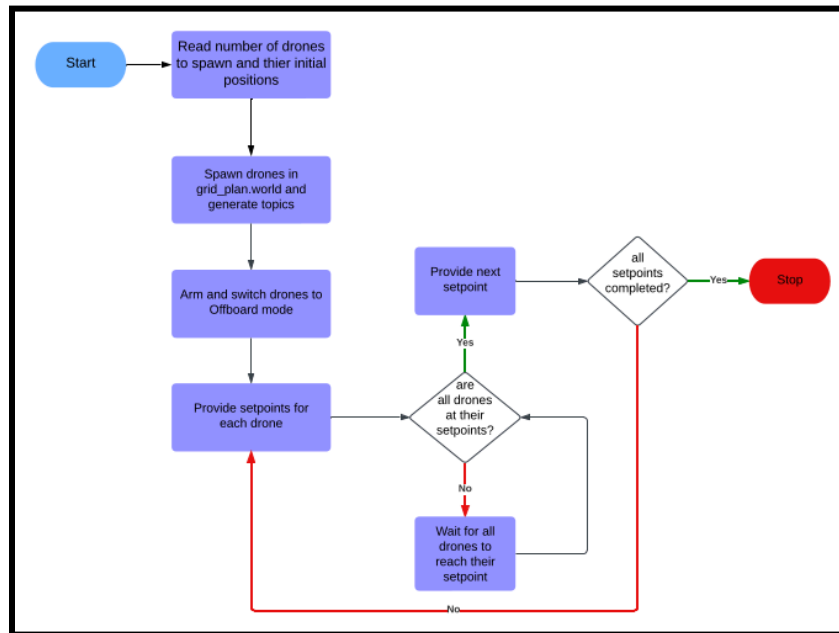
**[Date: 8 December 2024]**

Task 2.07/Task 2.08: Modified both L1 and L2 tests to get better coverage across the package and added the files that were needed to be excluded from the code coverage.

**[Date: 9 December 2024]**

Updated README, Tasks 2.09 to 2.012 completed

**Updated Algorithm:**



**[Date: 10 December 2024]**

Merge phase 2 development branch with main, rename commits to match convention. Modify the main branch for submission.

**[Date: 11 December 2024]**

Merged Kshitijs repo which has codecov tests and doxygen generation setup, made final changes to repo and submitted the final repository.