Technical Report: Evaluation and Selection of Simulators for the SkyVision Multi-Drone Project

Ahmed Belal, Abdllah Ammar, Yousef
Al Alamein International University, New Alamein City, Egypt

December 7, 2025

Abstract

This technical report documents our evaluation process and findings regarding several open-source simulators and middleware for the SkyVision multi-drone project. We systematically explored Gazebo, AirSim, and Isaac Sim as our simulators, and PX4, QGroundControl, and ROS2 as middleware, comparing their features, integration complexity, sensor and physics realism, and suitability for collaborative mapping and control tasks specific to SkyVision. The report details our hands-on experiments, highlights the strengths and limitations of each platform, and explains our decision to adopt AirSim with ROS2 as the foundation for SkyVision's simulation and development work. Our experience aims to guide other drone researchers and students in selecting appropriate simulation tools for aerial robotics projects.

1 Introduction

SkyVision is a specialized graduation project in collaboration with JMU, focused on developing an autonomous swarm of Unmanned Aerial Vehicles (UAVs) for rapid disaster response. The system leverages Artificial Intelligence to coordinate multiple drones for search-and-rescue operations, utilizing real-time Computer Vision (CV) and Simultaneous Localization and Mapping (SLAM).

This report documents the selection, configuration, and evaluation of the simulation environment used to develop and test SkyVision before physical deployment.

2 Project Requirements

To successfully simulate the SkyVision swarm, the simulation platform needed to meet specific criteria:

- **Multi-Vehicle Support:** Capability to spawn and control multiple drones simultaneously (swarm functionality).
- **ROS2 Integration:** Native compatibility with ROS2 (Humble/Jazzy) for communication between agents. We are currently using Humble for better compatibility.
- **Sensor Simulation:** High-fidelity simulation of RGB cameras, LiDAR, and IMUs for Computer Vision and SLAM tasks.
- **Physics Fidelity:** Accurate flight dynamics compatible with the PX4 Autopilot stack.
- **Customizability:** Ability to modify drone models (e.g., Holybro X500) and environments (disaster scenarios).
- **Visualization:** Real-time visualization of drone states, sensor data, and mapping results.

3 Simulator and Middleware Options

We evaluated three primary candidates for the SkyVision simulators:

- **AirSim:** Built on Unreal Engine by Microsoft, offering photorealistic rendering and APIs for controlling drones.
  Pros: Extremely high visual fidelity (great for CV).
  Cons: Deprecated/archived by Microsoft; heavy resource usage; difficult setup with modern ROS2 versions. Some community-maintained forks are compatible with newer versions of Unreal and ROS2.
- **Gazebo (Harmonic):** The open-source standard for robotics simulation, previously known as Ignition. Default simulator for ROS2.
  Pros: Native ROS2 integration; lightweight; official support from PX4 Autopilot; large community.
  Cons: Visuals are less realistic than Unreal Engine (though Harmonic is significantly improved).

- **NVIDIA Isaac Sim:** Scalable robotics simulation based on NVIDIA Omniverse.
  Pros: Photorealistic physics and visuals; GPU-accelerated.
  Cons: Requires high-end NVIDIA hardware; steeper learning curve for custom drone integration compared to Gazebo.

4 Middleware

For middleware and autopilot, we considered:

- **PX4:** Open-source autopilot stack for drones, widely supported in simulation and real hardware.
- **QGroundControl:** Ground station software for monitoring and controlling PX4-based drones.
- **ROS2:** Modern robotics middleware for distributed communication, sensor integration, and control.

5 Evaluation Process

We compared the simulators and middleware based on the following metrics:

- **Integration Ease:** How easily does it connect with PX4 and ROS2?
- **Swarm Scalability:** Can it run 3+ drones without crashing the host PC?
- **Sensor Data:** Ease of accessing camera/LiDAR streams in Python/OpenCV.
- **Long-term Viability:** Is the software actively maintained?
- **Community Support:** Availability of documentation, forums, and troubleshooting resources.

6 Setup and Experiments

6.1 Installation and Configuration

- **Middleware:** Installed ROS2 Humble on Ubuntu 22.04.
- **Simulator:** Set up Gazebo Harmonic using official OSRF repositories.
- **Autopilot:** Cloned PX4-Autopilot repository and configured the build toolchain.

6.2 Integration Steps

- **SITL Bridging:** Configured MicroXRCE-DDS Agent to bridge topics between PX4 (running in simulation) and ROS2 nodes.
- **Model Creation:** Created a custom SDF (Simulation Description Format) file for the Holybro X500 frame, attaching virtual sensors (2D LiDAR and RGB camera).
- **Swarm Launching:** Wrote a shell script to spawn multiple instances of px4_sitl with unique namespace IDs (e.g., drone1, drone2) and distinct UDP ports for MAVLink communication.
- **Visualization:** Used RViz2 and QGroundControl to monitor drone states and sensor data in real time.

7 Comparison and Decision

After extensive testing, we found:

- **Gazebo Harmonic** provided the best ROS2 integration and swarm scalability, with reliable sensor simulation and active community support.
- **AirSim** offered superior visual fidelity but was harder to maintain and integrate with ROS2 and PX4, especially for multi-drone scenarios.
- **Isaac Sim** was powerful but required hardware resources and time beyond our project scope.

We selected **Gazebo Harmonic with PX4 and ROS2** as our primary simulation stack for SkyVision, enabling us to develop, test, and validate swarm algorithms, sensor fusion, and disaster scenario mapping before hardware deployment.

8 Implementation and Results

- Developed Python scripts for multi-drone control, path planning, and SLAM integration.
- Successfully simulated collaborative mapping missions with 3+ drones in Gazebo.
- Validated sensor data streams (LiDAR, camera) for real-time CV and SLAM tasks.
- Used QGroundControl for mission monitoring and manual override.
- Documented all setup steps and troubleshooting for reproducibility.

9 Lessons Learned

- Early testing with multiple simulators is crucial for identifying integration issues.
- Community support and documentation save significant development time.
- ROS2 and PX4 are essential for scalable, real-world drone projects.
- Visual fidelity is less important than reliability and integration for swarm research.

10 Conclusion

This report summarizes our evaluation and selection of simulation tools for the SkyVision multi-drone project. By choosing Gazebo Harmonic, PX4, and ROS2, we established a robust, scalable, and well-supported simulation environment for developing autonomous drone swarms. Our experience and documentation aim to assist future teams in making informed decisions for aerial robotics research.

Acknowledgments

We thank Al Alamein International University and JMU for their support and collaboration, and the open-source communities behind Gazebo, PX4, ROS2, and AirSim for their invaluable resources.