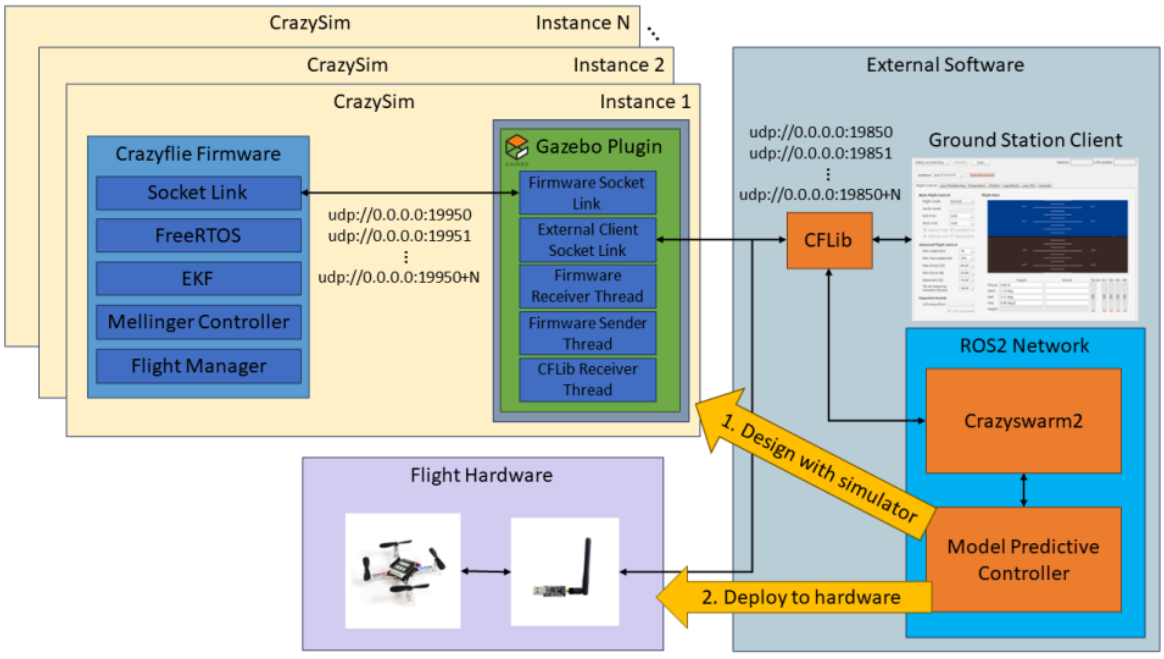
**Problem Statement** Develop and implement an efficient multi-stage swarm control algorithm using the Crazyflie drones in the Crazyswarm simulation framework. The algorithm should progress through five challenge stages, starting from basic single-drone flight control to advanced swarm behavior with dynamic collision avoidance. Participants must utilize the pycrazyswarm API and ROS 2 Humble to execute coordinated maneuvers, dynamic trajectory planning, and real-time swarm reconfiguration in a simulated environment. The final solution should be optimized for precision, stability, and efficiency in executing swarm-based tasks.



| Stage/Component | Marks | Criteria |
|---|---|---|
| **Quiz on Swarm Concepts & Crazyswarm API** | 25 | Already allotted based on understanding of theoretical concepts and API usage. |
| **Stage 1: Basic Flight (Single Drone)** | 10 | - Correct implementation of takeoff, hover, and landing.<br>- Proper use of pycrazyswarm API commands. |
| **Stage 2: Trajectory Tracking (Single Drone)** | 10 | - Smooth execution of the goTo command.<br>- Precision in reaching the specified target position. |
| **Stage 3: Coordinated Flight (Multiple Drones)** | 15 | - Synchronized takeoff, formation, and landing.<br>- Proper group command execution using Crazyswarm. |
| **Stage 4: Dynamic Trajectory Control (Single Drone)** | 15 | - Real-time trajectory re-planning and execution.<br>- Correct implementation of circular motion. |
| **Stage 5: Advanced Swarm Behavior with Collision Avoidance** | 20 | - Proper swarm coordination and collision avoidance logic.<br>- Dynamic trajectory adjustments in response to neighboring drones. |
| **Code Quality & Documentation** | 5 | - Clean, well-commented, and structured code. |
| **TOTAL** | 100 | |