UAV Swarm Coordination for Tracking and Surrounding Dynamic Targets

Problem Statement/Abstract:

Exploring various UAV swarm algorithms subsequently implement and test them in UAV simulation environments. Extend such algorithms for object tracking in swarm scenarios.

Introduction:

UAV swarm coordination for tracking and surrounding dynamic targets has several potential applications across various fields. These systems leverage multiple drones working together to monitor, track, and surround moving targets effectively, which is valuable in both civilian and military settings. Here are some notable applications:

1. Military and Defence

• Surveillance and Reconnaissance: UAV swarms can provide real-time intelligence by continuously tracking enemy vehicles, personnel, or equipment. Multiple drones can surround and monitor the target from different angles, providing enhanced situational awareness.

• Air Defence: UAV swarms can be used for detecting and intercepting hostile aerial threats, such as enemy drones or aircraft, by surrounding and tracking them with precision.

2. Agriculture

• Crop Monitoring: UAV swarms can be deployed to monitor large agricultural areas and track changes in crop health, detecting potential threats such as pests or disease outbreaks, and even assessing soil conditions or water needs.

• Precision Spraying: UAV swarms can work in coordination to apply fertilizers or pesticides precisely, surrounding a specific area to ensure even coverage.

3. Law Enforcement

• Target Pursuit: UAV swarms can help law enforcement track fleeing suspects or vehicles in real-time, surrounding the target and providing intelligence for ground units.

• Surrounding and Detaining: In certain scenarios, such as high-risk arrests, UAV swarms can assist in surrounding suspects to ensure they are effectively contained, while also providing live intelligence to officers.

4. Entertainment and Film Making

• Aerial Cinematography: UAV swarms can track and follow moving objects or people while filming, providing dynamic and synchronized shots from multiple perspectives without the need for human operators.

• Light Shows: Drones can be used in large-scale coordinated light displays, tracking moving patterns or surrounding specific areas to create visually stunning effects.

5. Space Exploration and Satellite Operations

• Satellite Constellation Coordination: UAV swarms can assist in the coordination of satellite fleets or constellations, ensuring they are correctly positioned to track dynamic targets such as asteroids or space debris.

• Astronomical Surveying: Swarm-based UAVs could potentially assist in astronomical research, tracking celestial objects or phenomena in a coordinated manner.

Technologies/Tools:

MavLink protocol using UDP Communication

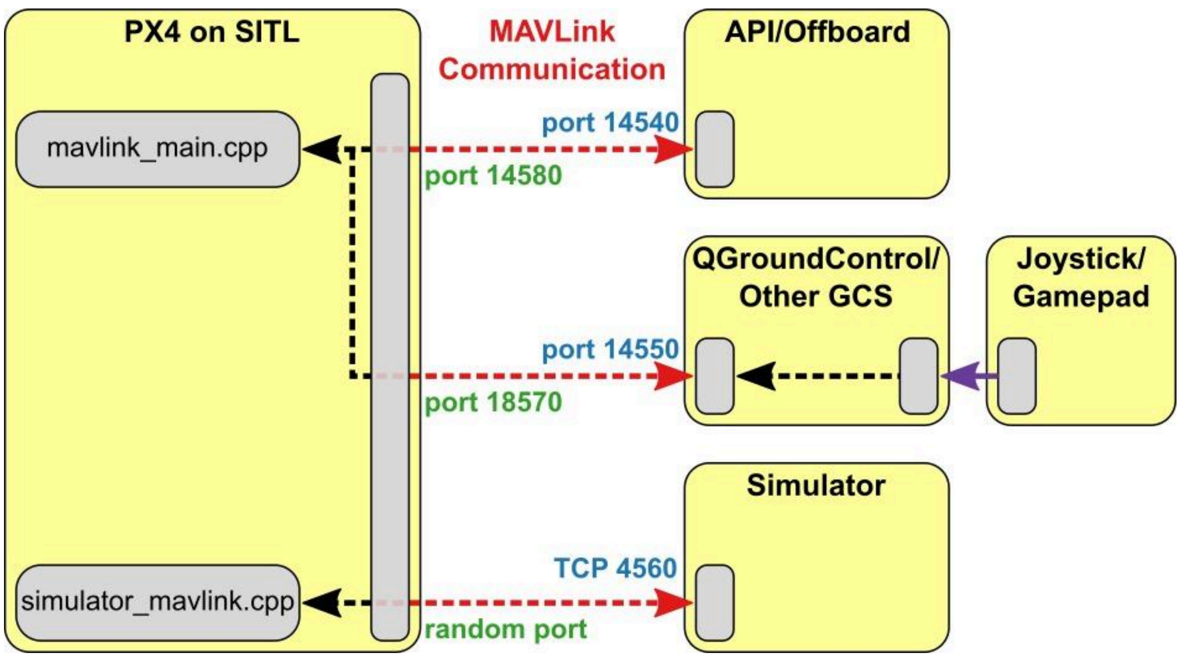
ROS (Robot Operating System)

ROS is an open-source framework for robot software development. It provides tools, libraries and conventions for developing and controlling robots, including UAVs.

Path planner, Map server node are essential parts in ROS.

PX4

PX4 is an open-source autopilot firmware widely used in UAVs. It provides control algorithms, flight modes, and hardware support for various flight controllers.



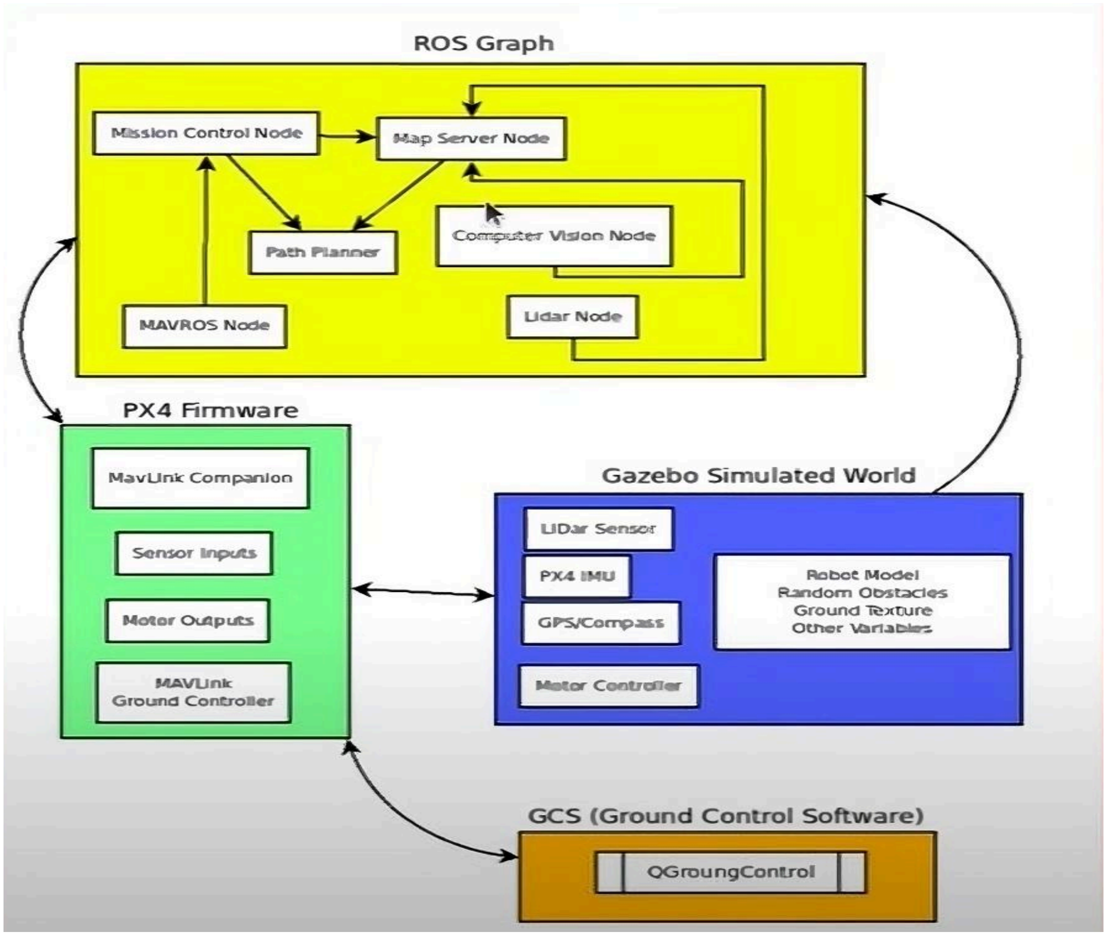
QGC (QGroundControl)

QGroundControl (QGC) is an open-source ground control station software for UAVs, providing a user-friendly interface for mission planning, monitoring, and controlling UAVs.

Gazebo

Gazebo is a powerful 3D robotics simulator that allows you to model, simulate, and test robot systems in complex environments. It supports realistic physics, high-quality 3D rendering, and a wide range of sensors and actuators.

Creates a simulated world that allows the model to work on certain constraints.



Plan of Action:

Sharath Chandra B: Responsible for conducting research, programming the UAV swarm system, and testing the hardware components, including integration of sensors, communication modules, and control systems.

Sohan Krishna G: Focused on developing the software for the UAV swarm, including algorithm implementation for object detection, tracking, and coordination, as well as performing software testing to ensure system functionality and reliability.

Sheethal M: Overseeing report preparation, documenting progress and results, providing non-technical support, simulation analysis and observing the social and ethical aspects of the project, including the impact of UAV swarms on privacy and safety concerns.

Project Timeline Distribution:

Software Development (Phase 1):

week 1- 26/01/2025 - 01/02/2025

week 2- 02/02/2025 - 08/02/2025

week 3- 09/02/2025 - 15/02/2025

week 4- 02/03/2025 - 08/03/2025

Hardware Development (Phase 2):

week 5- 09/03/2025 - 15/03/2025

week 6- 16/03/2025 - 22/03/2025

week 7- 23/03/2025 - 29/03/2025

week 8- 26/03/2025 - 01/04/2025

Swarm Coordination Algorithm Implementation:

A. Boids Algorithm for Flocking

Importance:

The Boids algorithm ensures collision avoidance, velocity matching, and cohesive movement within the swarm. This is useful for collective UAV movements like patrolling or area coverage.

Code:

```
import numpy as np

from pymavlink import mavutil

# UAV Flocking Rules: Separation, Alignment, Cohesion

class UAVFlock:

    def __init__(self, uav_id, position, velocity):

        self.uav_id = uav_id

        self.position = np.array(position)

        self.velocity = np.array(velocity)


    def update_velocity(self, neighbors):

        separation = np.mean([self.position - n.position for n in neighbors], axis=0)

        alignment = np.mean([n.velocity for n in neighbors], axis=0)

        cohesion = np.mean([n.position for n in neighbors], axis=0) - self.position

        self.velocity += 0.1 * separation + 0.1 * alignment + 0.05 * cohesion
```

B. Leader-Follower Formation

Importance:

This algorithm ensures that UAVs follow a designated leader, forming structured formations during movement. It is used in surveillance, search-and-rescue, and convoy security applications.

Code(function):

```
def send_waypoint(master, lat, lon, alt):
    master.mav.mission_item_send(
        master.target_system,
        master.target_component,
        0,
        mavutil.mavlink.MAV_FRAME_GLOBAL_RELATIVE_ALT,
        mavutil.mavlink.MAV_CMD_NAV_WAYPOINT,
        0, 0, 0, 0, 0, 0,
        lat, lon, alt
    )
```

C. Potential Fields for Target Coordination

Importance:

Potential field algorithms help in obstacle avoidance and precise target approach by computing attractive and repulsive forces between the UAV, targets, and obstacles.

Code(function):

```
def calculate_force(uav_position, target_position, obstacles):

    attraction = target_position - uav_position

    repulsion = sum([uav_position - obs for obs in obstacles])

    return 0.5 * attraction + 0.3 * repulsion
```