Multi UAV Obstacle Avoidance, Formation-Aware and Communication-Aware Path-finding Algorithm in a Known Environment

Asif Rizwan and Sampurak Talukdar
Instructor: Prof. Ketan Rajawat

IIT Kanpur

2025/04/25

Summary

Problem Statement

This algorithm is for navigating N UAVs from point A to B in a 2D map with known environment, while satisfying safety, formation, and communication-connectivity constraints. The algorithm modifies the classical A* search to include swarm-specific metrics like formation feasibility and algebraic connectivity.

Inputs

- A: Start position of the virtual leader (VL)
- B: Target position
- Map: 2D grid with known static obstacles and boundaries
- N: Number of UAVs
- $Formations$: Predefined set of formation templates $\{F_1, F_2, F_3\}$
- r_{min}: Minimum inter-UAV separation
- r_{comm}: Communication range
- d_{goal}: Arrival radius threshold around B

Constraints

1. **Safety:** No inter-UAV or UAV-obstacle or UAV-boundary collisions
2. **Formation:** Maintain one of the valid formations:
   - F_1: Regular Polygon (Preferred)
   - F_2: Two-Line
   - F_3: Single-Line (Fallback)
3. **Connectivity:** UAVs must maintain a connected communication graph with algebraic connectivity $\lambda_2 > \epsilon$

Heuristic Function

The total cost function for A* is defined as:

$$f(n) = g(n) + h(n)$$

where:

$$h(n) = w_1 \cdot \text{SafetyScore}(n, F) + w_2 \cdot \text{FormationPriority}(F) + w_3 \cdot \text{ConnectivityScore}(n, F) + w_4 \cdot \text{Distance}(n, B)$$

- $g(n)$: Path cost from start to node n
- $\text{SafetyScore} = \begin{cases} 0 & \text{if formation is safe} \\ \infty & \text{otherwise} \end{cases}$
- $\text{FormationPriority}(F_1) = 0, \text{FormationPriority}(F_2) = 10, \text{FormationPriority}(F_3) = 20$
- $\text{ConnectivityScore} = \frac{1}{\lambda_2}$, λ_2 is the algebraic connectivity of the UAV communication graph
- $\text{Distance}(n, B)$ is the Euclidean distance from node n to goal B

Algorithm

Input: A, B, Map, N, $Formations$, $r_{\min}$, r_{comm}, d_{goal}
Output: Path from A to B with formation sequence

```
Initialize open list with (A, g = 0, f = h(A), F_1);
Initialize closed list and came-from map;
while open list is not empty do
    current ← node with lowest f in open list;
    if distance(current, B) ≤ d_goal then
        return reconstructed path and formation sequence;
    end
    Move current from open to closed list;
    foreach neighbor of current do
        if neighbor in closed list then
            continue
        end
        foreach formation F in {F_1, F_2, F_3} do
            if F is not feasible at neighbor due to obstacles/boundaries then
                Attempt to squeeze F while preserving r_min;
                if still not feasible then
                    break
                end
            end
            Compute UAV positions at neighbor using formation F;
            if violates safety constraint then
                break
            end
            Compute connectivity graph and Laplacian matrix L;
            Compute λ_2(L);
            if λ_2 < ε then
                break
            end
            Compute g(neighbor) and h(neighbor) as defined above;
            f(neighbor) = g + h;
            if neighbor not in open list or has lower f then
                Update open list with f, g, F;
                Update came-from and formation history;
            end
            break (use first valid formation)
        end
    end
```

Stopping Criteria

The algorithm stops when:

$$\forall i \in [1, N], \quad \|UAV_i - B\| \leq d_{\text{goal}}$$

Limitations and Future Work

1. Formation orientation
2. Formation switching dynamics
3. Smoothening of outlier formation switching
4. Trajectory Optimization

Implementation

Demo: Implementation in MATLAB and ROS+Gazebo