Emerson Hall

Ramana M. Pidaparti

1 September 2025

Problem Formulation & Dataset Identification

Problem Statement

We want a UAV to fly from a start pose to a goal pose while avoiding obstacles and using as little time and energy as possible. The environment may have wind and moving obstacles. The planner must output a safe, flyable trajectory that respects vehicle limits.

Input at each time step

- UAV pose and velocity
- Sensor map or occupancy grid around the UAV
- Estimated wind vector
- Goal pose

Output

- Next action for the controller (for example desired velocity, waypoint, or thrust/attitude command), or the full trajectory if using an offline planner.

Assumptions and Operating Conditions

- 3D flight, indoor or outdoor, simulated first.
- Known hard no-fly zones and static obstacles.
- Possible moving obstacles with bounded speeds.
- Quadrotor with limits on speed, acceleration, jerk and tilt angle.
- State estimates are available from the simulator (ground truth or filtered).

Variables, Constraints and Objectives

- Decision variables: Control actions or way points
- Hard Constraints
  - Dynamics: respect UAV kinematics/dynamics (speed, accel, tilt).
  - Safety: no collisions, maintain a minimum clearance d_{min}.
  - Boundaries: stay inside flyable airspace; respect no-fly zones.
  - Actuation: actions within motor and controller limits.
- Soft Constraints
  - Smoothing: avoid sharp turns and high jerk.
  - Energy: prefer low thrust and short hover time.
- Objectives
  - Time: minimize total flight time
  - Energy: minimize energy or integrated thrust.
  - Path quality: minimize path length and maximize smoothness.
  - Reliability: maximize success rate under disturbances.

A simple weighted objective for classical optimization:

$$\frac{min}{u^{0:T-1}} \alpha * time + \beta * energy + \gamma * path_{length} + \delta * smoothness$$

Subject to the hard constraints above.

RL Formulation

- MDP state: stacked observations (UAV pose/velocity, goal vector, local occupancy grid or lidar/range image, wind estimate)
- Action:
  - Continuous: desired body-frame velocities $\{v_x, v_y, v_x\}$ and yaw rate, or thrust and attitude

    - Discrete: small set of motion primitives (forward, strafe, ascend, turn)
- Transition: given by the simulator physics
- Reward:
    - Large at goal completion
    - Penalty for collision or entering no-fly zone (episode ends)
    - Small step penalty to encourage short time
    - Path curvature/jerk penalty for smoothness
    - Energy proxy (thrust or control effort)
    - Progress reward proportional to reduction in distance-to-goal
    - Penalty when violating safety buffer
- Episode Termination: reached goal, collision, time limit, or leaving bounds.

Observation and Action Details

- Observation: UAV state, goal vector, wind estimate, and local occupancy map.
- Action: continuous velocity commands or motion primitive IDs.

Chosen ML Approach

- Primary: Reinforcement Learning (PPO, SAC) for adaptability
- Secondary: Imitation/Supervised Learning to pretrain from expert paths.
- Hybrid: Global A* or PRM planner plus local RL policy for dynamic avoidance.

Evaluation Metrics

- Success rate, time to goal, energy use, path length, smoothness, constraint violations, inference time, robustness under wind and moving obstacles.

Dataset Plan

- Simulation-generated date using ROS2, Gazebo and PX4 SITL, including static maps, moving obstacles and wind conditions.
- Store states, actions, outcomes and metadata per episode.
- Supervised learning uses expert labels (A*, RRT*), RL uses rewards only.
- Split: 70% train, 15% validation, 15% test with unseen maps in test set.
- Preprocessing:
  - Normalization
  - Standardized maps
  - Encoding no-fly zones
  - Data augmentation with noise and wind variation

Baselines and Ablations

- Baselines
  - A*
  - RRT*
  - A* with smoothing
  - Possibly MPC
- Ablations
  - Curriculum vs no curriculum
  - Hybrid vs local only RL
  - With/without potential field reward shaping