# RL and Optimal Control for Robotics

Project -2

Abhimanyu Suthar

December 20, 2024

## Methodology

This report details the implementation of a reinforcement learning controller for a quadrotor system. The goal was to develop a controller that could navigate the quadrotor from an initial position of $[-2, 0, 0, 0, 0, 0]^T$ to a target position of $[2, 0, 0, 0, 0, 0]^T$ while avoiding obstacles in the environment.

## 1 Reward Function Design

The reward function was carefully designed to encourage desired behaviors while penalizing unsafe actions. It consists of three main components.

### 1.1 Target-Reaching Reward

The primary reward component uses an exponential form:

$$reward = \exp(-\frac{1}{2}(x - x^{)})^T Q(x - x^{)}) - \frac{1}{2}(u - u_{gravity})^T R(u - u_{gravity})) \tag{1}$$

This reward structure provides several advantages:

- It is bounded between 0 and 1, providing stable learning signals
- The exponential decay ensures higher rewards as the quadrotor approaches the target
- The quadratic form (Q matrix) allows different weights for position and velocity terms
- The action error term (R matrix) encourages smooth control inputs

The Q matrix was designed with different weights:

$$Q = \text{diag}([1, 0.1, 1, 0.1, 1, 0.1]) \tag{2}$$

This weighting scheme places:

- Higher emphasis (1.0) on position errors ($p_x, p_y$) and orientation ($\theta$)
- Lower emphasis (0.1) on velocities ($v_x, v_y$) and angular velocity ($\omega$)

## 1.2 Safety Penalties

Two safety-related penalties were implemented:

### 1.2.1 Out-of-bounds Penalty (-100.0)

This penalty enforces strict boundaries on the quadrotor's movement and helps maintain the system within safe operating limits:

- Position: $[-4, 4]$ meters
- Velocities: $[-10, 10]$ m/s
- Angle: $[-2\pi, 2\pi]$ radians
- Angular velocity: $[-10, 10]$ rad/s

### 1.2.2 Collision Penalty (-1.0)

A relatively mild penalty compared to boundary violations to discourage contact with obstacles while still allowing exploration.

## 1.3 Action Regularization

The action error term uses:

$$R = \text{diag}([0.01, 0.01]) \tag{3}$$

This component:

- Encourages efficient control by penalizing excessive actuator inputs
- Accounts for gravity compensation to allow stable hovering
- Uses small weights (0.01) to prioritize task completion over control efficiency

# 2 Training Configuration

The PPO algorithm was configured with the following key parameters:

- Learning rate: 9e-3 (relatively high to encourage rapid learning)
- Batch size: 32 (small enough for stable updates)
- Steps per update: 2048 (sufficient for exploring the state space)
- Entropy coefficient: 1e-2 (encourages exploration)
- GAE lambda: 0.95 (balances bias and variance in advantage estimation)

# 3 Results

The trained policy successfully learns to:

- Navigate from the starting position to the target
- Avoid obstacles along the path
- Maintain stable flight within the specified bounds
- Complete the task within the 200-step episode limit

# 4 Conclusion

The implemented reward function successfully balances the competing objectives of reaching the target, avoiding obstacles, and maintaining stable flight. The exponential reward structure, combined with appropriate safety penalties, provides a smooth learning signal that enables effective policy learning through PPO.