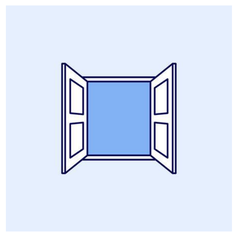
Delivery Drone through Narrow Passage

Weiyuan Hsieh, Chris Lee

INTRODUCTION/MOTIVATION

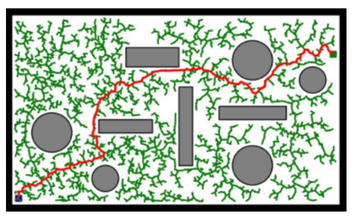
- **Introduction**
  - Delivery drones can be approximated as a quadrotor with a pendulum. Our project uses a path planning algorithm to find a dynamically feasible initial path from start to goal through narrow passages while avoiding obstacles. The width of the narrow passages are smaller than the maximum/nominal dimension of the quadrotor-pendulum system. The initial path is then fed into a trajectory optimizer to minimize the input cost, distance, and angular rotation of the quadrotor.

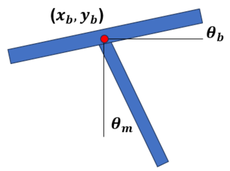
- **Motivation**
  - This problem is important because there are often situations where an autonomous system needs to pass through a narrow passage that requires the system to be in more compact configurations. An example would be a delivery drone with a package tied to it that needs to pass through a narrow window. The delivery drone would need to swing itself so that the package is at an angle when passing through the window. In these cases, an underactuated system results in a more complicated trajectory.
  - The problem is hard because of narrow passages and underactuation. Typical path planning algorithms do not deal well with narrow passages. Correctly considering obstacle constraints of a narrow passage is also challenging when the system is underactuated.



BACKGROUND

- **Quadrotor with a pendulum** – A delivery drone with payload can be modeled as a 2D quadrotor with a pendulum. The system is underactuated since the payload is free to swing as the quadrotor moves. The state consists of the x and y positions, the body and pendulum angular displacements, and their derivatives.
- **Rapidly exploring Random Tree (RRT)** - RRT searches non-convex spaces by randomly building a space-filling tree. The tree is constructed incrementally from samples drawn randomly from the search space.
- **Linear Quadratic Regulator RRT (LQRRT)** - LQRRT enhances RRT by integrating LQR control. It grows the tree by using LQR at each step to ensure feasible paths.



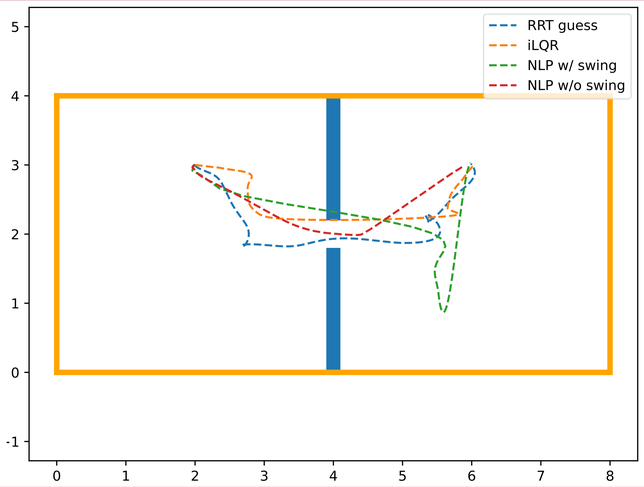


METHODS

- **Contributions**
  - Proposed a novel *convex-segmented* LQRRT that improves upon the regular LQRRT that results in closer-to-optimal initial trajectories and can be used to find paths through narrow passages.
  - Showed that LQRRT trajectory can work as an initial guess for trajectory optimization.
  - iLQR trajectory optimization with convex approximation can be used to optimize passage through narrow passages.
  - NLP trajectory optimization with elliptical obstacle approximation can be used to optimize passage through narrow passages.

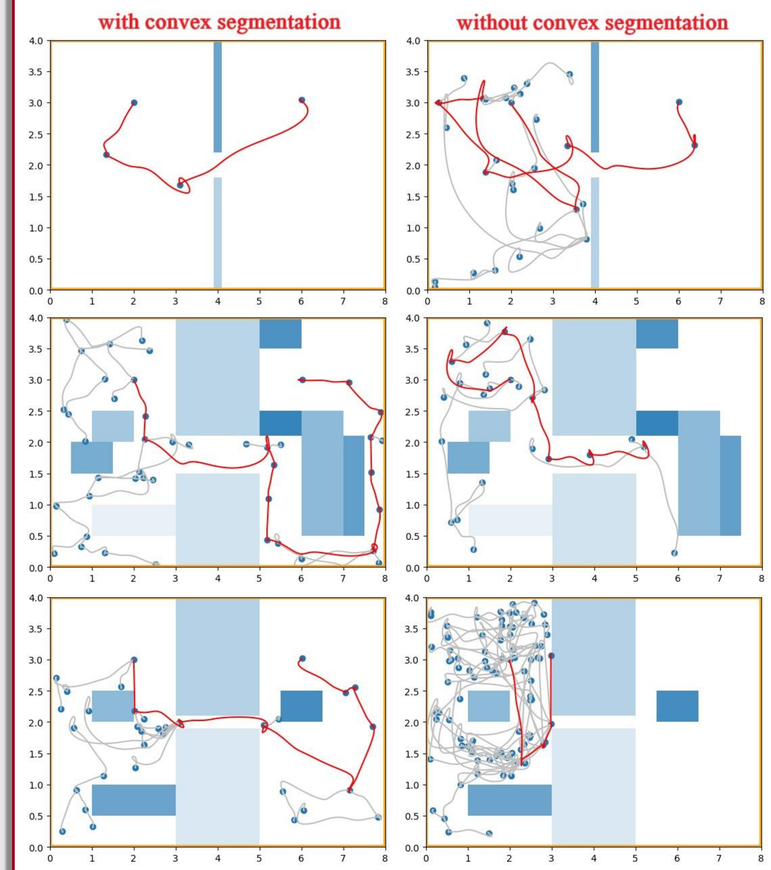
- **Key technical details**
  - Convex-segmentation - Chop the empty place into convex (rectangular) segments, and randomly sample states on the boundaries of the segment where the start state is. This makes the sampled states more likely to be reachable and greatly reduces the number of required RRT nodes. When we reach a new segment, we add its boundaries to the sampling space, and resample again until we reach the segment including the goal state.
  - LQRRT to ensure that the quadrotor can fit through narrow passages
  - We use iLQR to minimize trajectory distance (sum of norm(x[k], x[k+1])), terminal cost, input energy, and barrier function.
  - Within each iteration, we flatten the trajectory once to remove sharp corners.
  - Barrier function: $B(x) = \begin{cases} q_1 d^2(x), & if\ d(x) < 0 \\ 0, & otherwise \end{cases}$, where d is the signed distance field.
  - We additionally tried SciPy's CasADi and ipopt, and using elliptical approximation of box obstacles, can optimize trajectory for minimum distance, input energy, and velocity. Subject to initial state, input limit, boundary, and dynamics constraints. Uses discrete time linearized dynamics linearized around the goal state. Uses the tips of the wing and pendulum as points to calculate obstacle penalty.
  - Elliptical function: $C(x,y) = \frac{W}{(\frac{(x-c_x)^2}{r_x} + \frac{(y-c_y)^2}{r_y})^{\lambda+1}}$

RESULTS



RESULTS Continued

LQRRT With vs. Without Convex Segmentation



- **What worked**
  - Convex-segmented LQRRT- Chop the empty pace into convex (rectangular) segments, and randomly convex segmentation helps overcome narrow passages. State sampling becomes more uniform over the state space despite the presence of narrow passages.
  - iLQR with obstacle penalty for trajectory optimization.
  - NLP with elliptical approximation for trajectory optimization and an attempted swinging motion through the passage.

- **What didn't work**
  - Dynamics formulation not compatible with direct collocation in PyDrake
  - Swinging motion through the narrow passage that strictly obeys obstacle constraints in LQRRT
  - Strictly satisfying box obstacle constraints in NLP and other obstacle penalty methods

- **Potential extensions**
  - Consider non-box obstacles (e.g., irregular shapes)
  - Extend to 3D
  - Consider dynamic obstacles