

Intro

This project is to design a online motion planning algorithm for a quadmotor in a random complex grid node map in C++/ROS.

Design

My design is as followed:

1. Path planning with A *(**Project 1**).
2. Simplify the path with the Ramer-Douglas-Peucker algorithm
(https://en.wikipedia.org/wiki/Ramer-Douglas-Peucker_algorithm).
3. Generate a smooth trajectory with "Minimum Snap Trajectory Generation and Control for Quadrotors" by Daniel Mellinger and Vijay Kum. (**Project 4**)
4. Check the safety of the flight path.
5. If the trajectory is not certain, the trajectory is regenerated using the method from "Polynomial Trajectory Planning for Aggressive Quadrotor Flight in Dense Indoor Environments" by Charles Richter, Adam Bry and Nicholas Roy.

Dependencies

1. System dependencies

```
...
```

```
sudo apt-get install cmake libopenblas-dev liblapack-dev libarpack-dev libarpack2-dev  
libsuperlu-dev
```

```
...
```

2. Install Armadillo

```
...
```

```
xz -d armadillo-9.870.2.tar.xz  
tar -xvf armadillo-9.870.2.tar  
cd armadillo-9.870.2  
mkdir build  
cd build  
cmake ..  
make  
sudo make install
```

```
...
```

Introduction to all important functions

- * `random_complex`: randomly generate obstacle point cloud map;
- * `waypoint_generator`: given target point;
- * `odom_visualization`: quadrotor visualization;
- * `pcl_render_node`: a simple version of the local sensor model, which returns the obstacle point cloud in the local area;
- * **`trajectory_generator_node`**: The core part of the project, which generates a feasible polynomial trajectory;
- * `traj_server`: Convert polynomial trajectory into control instructions;
- * `so3_control`: Convert the control instruction into the actual control quantity;
- * `quadrotor_simulator_so3`: UAV simulation model.

Result

