

# Motion Planning Final Project Proposal

## Motion Planning for Quadrotor using RRT with Constraint

Dejun Guo, Xiang He, Shantnu, Roy

### 1 Introduction

Nowadays, unmanned aerial vehicles (UAVs) are applied to various tasks, such as search and rescue or surveillance. Quadrotor have been studied for use in such tasks due to the advantage in control that multiple rotors gives. Because of this, quadrotor can hover and fly through complex environments that traditional UAVs cannot navigate, for example in a damaged structure. Many problems need to be dealt with to achieve the autonomous navigation for quadrotor in such tasks. How to get the goal in a smooth path and avoid the obstacles as well as the dangerous area is the crucial issue we will focus on in this project.

Firstly, a map will be built based on an indoor environment which the user can edit through the interface. Secondly, a set of way points will be generated based on the RRT algorithm. These way points will guide the quadrotor from the starting point to the goal while avoiding the dangerous area and the obstacles. Thirdly, a path smooth algorithm will be applied to make the path satisfy the non-holonomic restriction of the quadrotor.

### 2 Map Generating and Visualization

Map generating will be done in MATLAB, using MATLAB GUI. A separate program will be created with GUI to help user drawing a new map easily, including map resizing, basic line, polygon, and circle element representing obstacles in the map. Map is stored temporary as picture to help user reviewing it. Then a cost map will be asked to point dangerous area on the generated map picture. After these two steps, three map files will be generated, a high resolution map, a lower resolution map and a cost map which has danger cost centered at the point user just point out. Map file generating is done in MATLAB by reading the map picture. High resolution is defined as grid of  $2cm$ . Lower resolution map consists grid of  $20cm$ .

3D visualization is built in ROS with OpenGL. Visualization node reads the stored high resolution map file and display it in 3D. Height of the environment is limited  $2.5m$ . Playback node will publish the real-time position and orientation of the quadrotor, based on planning result. Controls may be added at the end of the project to verify the constraint in RRT planning. Visualization node subscribes the localization and trajectory data and visualizes the quadrotor along the trajectory towards the goal.

### 3 Path Planning

In this project, the problem of collision-free path planning in environments with dangerous zones will be addressed. The path should avoid collision with obstacles but it is allowed to pass dangerous zones. However, this should be avoided as much as possible. For this purpose, first, we need to express concept of danger more specifically. For instance, in the case of this project, areas with high altitude can be dangerous for the quadrotor. Next, some extra cost can be defined for these zones regarding the level of danger.

Another contribution which will be considered in this project is implementing an algorithm for narrow spaces, such as doors or windows in a closed environment. Although at this moment it is not clear how this can be solved, we will take one of the algorithms for narrow passages which were discussed in class and modify it to be more optimal.

### 4 Path Smoothing

There are two requirements when deciding which curve function to use for the path smoothing function. The first is Local Support. When fitting a curve to the way-points, there are cases when the generated curve intersects with the obstacles. However, it is normally observed that not whole of the curve, instead only a part of it gets intersected. Hence it is logical to think that we need to reshape only that part of the curve which collides with the obstacle. To achieve this, we need a curve that can be reshaped only at certain regions. B-spline curve meets this requirement. Our second requirement is that we need continuity of velocity, acceleration and radius of curvature throughout or path. Since, 4th order B-spline curves are  $c^2$  continuous, we can use them to satisfy velocity, acceleration and radius of curvature constraints.

Another reason for choosing B-spline curve is that they can be easily implemented using De-Boors recursion algorithm, thus not adding much of any overhead.

The algorithm to generate this path smoothing cannot be disclosed for proposal submission, but after a thorough literature survey and feasibility study, it sounds like that it can be implemented to our problem.