**Project Report**

Jiahe Xu ([jxu109@jhu.edu](mailto:jxu109@jhu.edu)) Ray Zhang ([zzhan190@jhu.edu](mailto:zzhan190@jhu.edu))

1 Introduction

This project is intended to implement planning and control of a quadrotor in 3-D indoor environment. In this report, we will discuss how we design a trajectory and how we use a controller to follow the trajectory.

2 Modeling

The coordinate systems and free body diagram for the quadrotor are shown in Fig. 1. Since The heading (*yaw*) angle of the robot can be chosen freely without directly affecting the robot’s dynamics, we use *Z-X-Y* Euler angles to describe the rotation transform.



Figure 1, Coordinate systems and forces/moments acting on the quadrotor from [1].

To get from *W* (world frame) to *B* (body frame), we first rotate about *zW* by the *yaw* angle ψ, then rotate about the intermediate *x*-axis by the *roll* angle *φ*, finally rotate about the *yB* axis by the *pitch* angle θ. The rotation matrix for transforming coordinates from *B* to *W* is given by:

Where *ROTX, ROTY* and *ROTZ* are rotation matrix that only rotate about X-axis, Y-axis and Z-axis.

The center of mass in the world frame is denoted by vector ***r***. In the system, forces are gravity, in the *-zW* direction, and forces come from rotors *Fi*, in the *zB* direction. The equation involves the acceleration of the center of mass are:

Then, we have control input .

The angular velocity of the robot in the body frame is vector [*p, q, r*]T , it can be calculated by derivatives of the *roll, pitch,* and *yaw* angles with:

As for forces, each rotor generates a moment perpendicular to the *xB-yB* plane. Rotors 1 and 3 (in figure 1) rotate in the −*zB* direction while rotor 2 and 4 rotate in the *zB* direction. We let *L* be the distance from the axis of rotation of the rotors to the center of the quadrotor. We denote *I* asthe moment of inertia matrix referenced to the center of mass along the *xB*, *yB* and*zB* axes.

Each rotor has an angular speed i and produces a force *Fi*according to:

The moment produced by rotors is:

The angular acceleration determined by the Euler equations is:

we can rewrite it as:

where

Then, we can have our second control input *u2*:

Finally, we have the quadrotor’s equations of motion:

3 Trajectory Generator

In our project, we first generate a path represented by a set of 3-D points, which minimize the distance from the start point to the goal, and then we use flat output method build a polynomial function to represent the trajectory.

3.1 our method

We use cross-entropy method to generate the crude path. When the environment is complex (passage ways are narrow), using cross-entropy method find a valid path is quite inefficient. Despite we can use sample-based methods like RRT, PRM to explore the entire configuration space, it might need a large number of middle knots in the path, which causes CEM methods hard to converge. We first use little cubes to represent the environment, then use shortest path algorithm find a valid path (the path is not strictly the shortest in 3-D environment). Since the number of knots on the shortest path is too large to be the knots on CEM, we use ‘corner’ knots that change the path’s direction, as initial knots on CEM method.

After finding those turning points in the path, we directly connect knots with straight lines by adding more knots on each line.

4 Controller Description

5 Result

reference

**1 Trajectory generation and control for precise aggressive maneuvers with quadrotors**