1-10-2024

Implementing RRT* Path Planning with the Crazyflie Drone

Beichen Zhou
Washington University in St. Louis, beichen@wustl.edu

Implementing RRT* Path Planning with the Crazyflie Drone

Beichen Zhou, Department of Electrical & Systems EngineeringResearch Advisor: Professor Yiannis Kantaros

McKelvey School of Engineering

INTRODUCTION

Background

Crazyflie drone is a compact and agile platform ideal for advanced autonomous navigation. RRT*, an optimization of the standard RRT, offers efficient path planning in complex environments, enhancing the drone's ability to navigate dynamically around obstacles. This implementation showcases significant potential for applications in areas such as search and rescue, surveillance, and environ-mental monitoring.

Objectives

1. Path Planning with RRT*:

To utilize the RRT* algorithm for initial path planning in an area with unknown obstacles for the Crazyflie drone.

2. Obstacle Detection and Adaptation:

To enable the Crazyflie to detect obstacles during its flight using onboard sensors.

3. Dynamic Path Re-planning:

To implement real-time RRT* re-planning whenever the Crazyflie encounters obstacles, ensuring continuous navigation towards the target.

METHOD& PROCEDURE

Hardware platform - crazyflie

Fig.1 crazyflie

The crazyflie is a compact UAV(see fig.1), which is very suitable for practical testing of algorithms in limited sites, such as this experiment will be RRT* realized with the crazyflie, the crazyflie is equipped with an ultrasonic ranging module, which can detect the distance of front and back, left and right obstacles, and through the ultrasonic distance and angle, the width of the obstacle can be predicted.

The Flow deck V2 (see fig.2) enhances the Crazyflie 2.X by providing it with the capability to detect motion in any direction. It includes a VL53L1x Time-of-Flight (ToF) sensor, which precisely measures the distance from the drone to the ground.

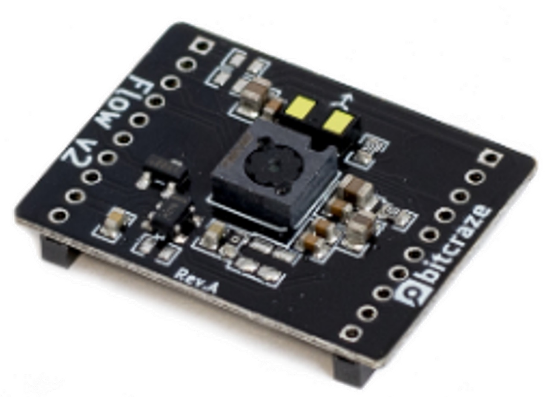

Fig.2 flowdeck

Additionally, the PMW3901 optical flow sensor tracks the drone's movements relative to the ground. These features transform the Crazyflie into a 3D flying robot capable of pre-programmed flights in various directions. They also make it a stable and user-friendly platform for beginners.

PLAN THE PATH

Principles of RRT*

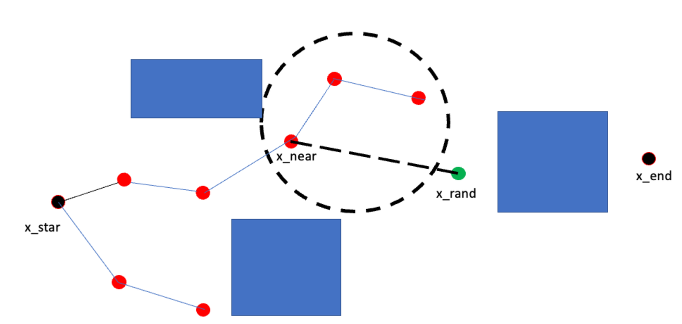

Fig.3: Randomization point

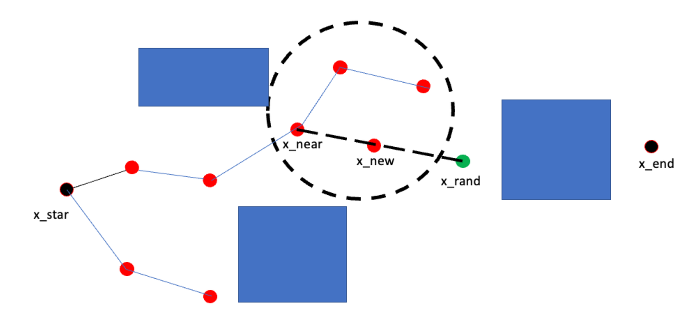

Fig.4: New node

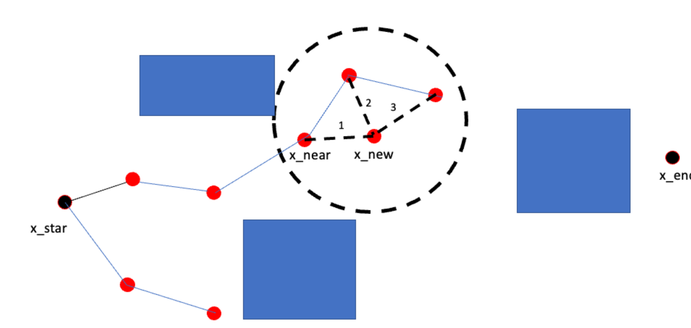

Fig.5: Optimized path

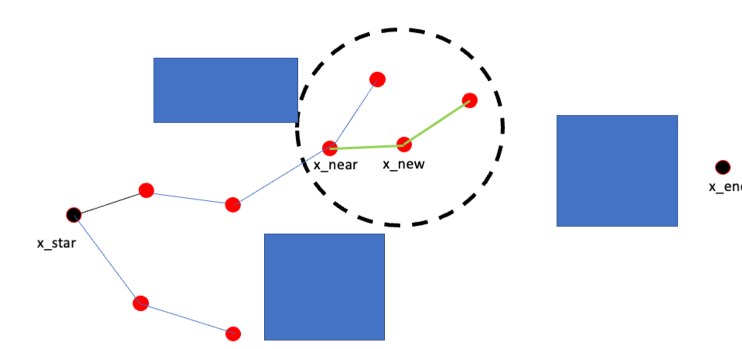

Fig.5: Rewire

Algorithm 1: RRT*

```
procedure RRT*(start, goal, obstacleList, area, maxIterations)
    Initialize tree T with start node
    for i = 1 to maxIterations do
        randomPoint ← getRandomPoint(area)
        nearestNode ← getNearestNode(T, randomPoint)
        newNode ← steer(nearestNode, randomPoint, stepSize)
        if isCollisionFree(nearestNode, newNode, obstacleList) then
            neighborNodes ← getNeighborNodes(T, newNode, radius)
            minCostNode ← findMinimumCostNode
            (nearestNode, neighborNodes, newNode)
            addNode(T, newNode, minCostNode)
            for all neighbor in neighborNodes do
                if isCollisionFree(neighbor, newNode, obstacleList) thennewCost ←
                    cost(newNode) + distance(newNode, neighbor)
                    if newCost < cost(neighbor) then
                        parentNode ← getParent(T, neighbor)
                        removeEdge(T, parentNode, neighbor)
                        addEdge(T, newNode, neighbor)
                    end if
                end if
            end for
        end if
    end for
    path ← getPath(T, start, goal)
    return path
end procedure
```

RRT* pseudocode

Experimental process

- **Initial Path Simulation**: Drone takes off and uses RRT* to simulate a pre-liminary obstacle-free path from start to target.
- **Flight Execution:** Drone follows the initial simulated route.
- **Obstacle Detection:** Ultrasonic sensors scan for and detect obstacles during flight.
- **Obstacle Response:** Drone halts upon detecting an obstacle.
- **Route Replanning:** RRT* recalculates a new path avoiding the detected obstacle.
- **Adaptive Navigation:** Drone resumes flying, adapting its route based on any new obstacles encountered.
- **Destination Reach:** Continues this process until the target is successfully reached.

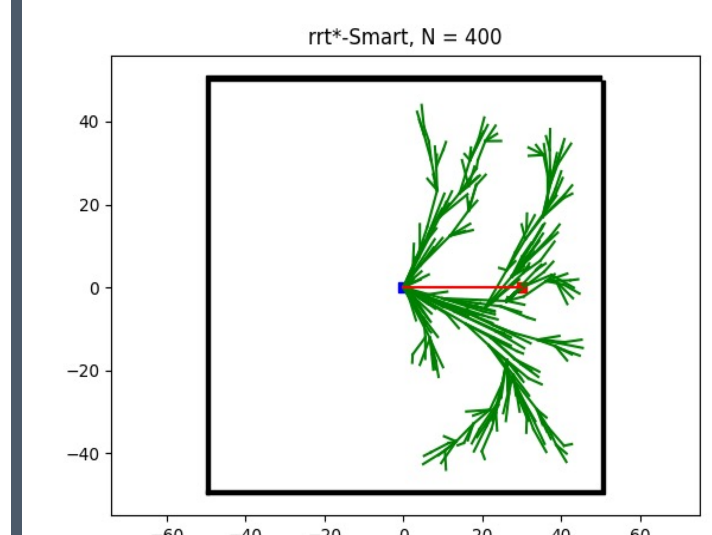

Fig.6: RRT* approach route without obstacles

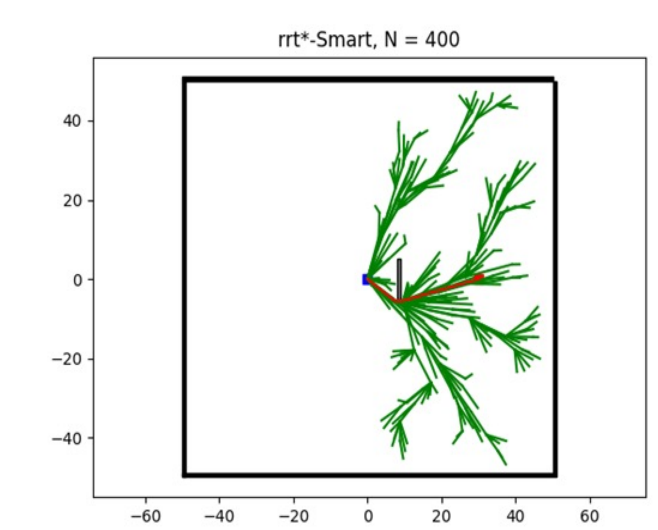

Fig.7: Drone detects part of an obstacle

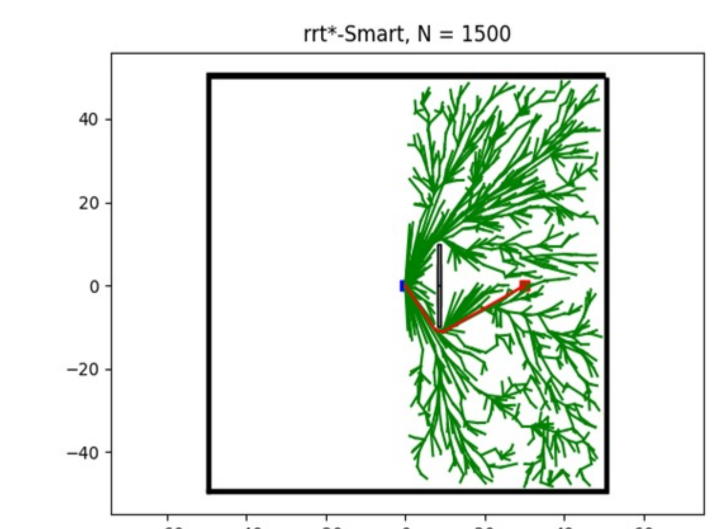

Fig.8: Drones detect more obstacles

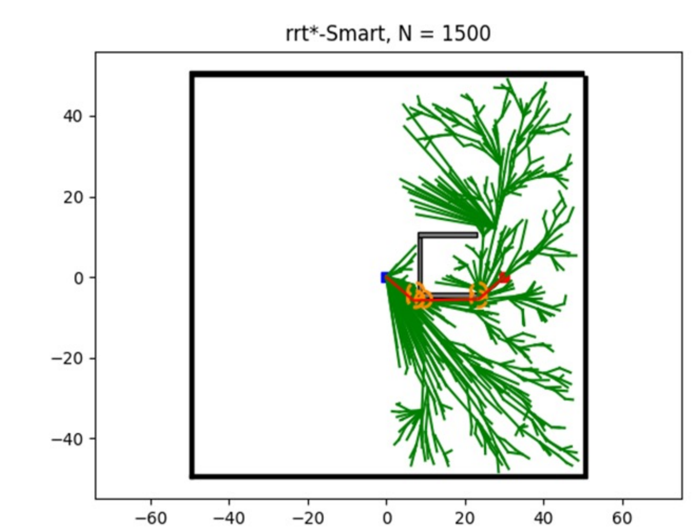

Fig.9: Fly to the target point after several probes

CONCLUSION

In our drone project, we faced challenges with the drone's limited multitasking capability, leading to time inefficiencies due to segmented path navigation for obstacle detection. This also increased RRT* iterations and computational load. To overcome these limitations, we plan to implement OptiTrack and ROS for enhanced multitasking and precision control, aiming to streamline the navigation process and reduce inefficiencies.

Reference：

S. Karaman and E. Frazzoli, "Incremental Sampling-based Algorithms for Optimal Motion Planning," arXiv preprint arXiv:1005.0416, 2010. [Online]. Available: https://arxiv.org/abs/1005.0416