Kodavanti Rama Sravanth
Indian Institute of Technology Hyderabad

SM5033 Internet of Things - Project Report

Instructor: Dr.P.Rajyalakshmi

TA: Sanju Kumar

Name: Kodavanti Rama Sravanth

Rollno: CS20BTECH11027

Topic : Optimizing B - Spline curves for robot navigation

October 16, 2023

Contents

1 Introduction

1.1 Problem Statement

- **Robots aren't Humans**, They need to have a definite path priorly to travel from start to end without hitting the obstacles in the middle.
- So the problem is how we guide the robot to travel between two points safely & it shouldn't damage the mechnical parts of it while travelling.

1.2 Possible Solution

- We need to create an algorithm which gives the path that should be taken by the robot for the save travel.
- We have taken **B - Spline** curves as the baseline for path planning algorithm. Also we need to optimize the algorithm such that robot's travel is smooth & safe.

1.3 Motivation behind choosing B - Spline curves

- B-spline curves are known for their smoothness and continuity properties. This characteristic allows for seamless and continuous motion planning, reducing the likelihood of sudden jerky movements or abrupt changes in direction, which is crucial for the stability and safety of robot navigation.
- B-spline curves allow for local control, meaning that changes to one segment of the curve do not affect the entire curve. This property is beneficial in dynamic environments where the robot needs to adapt its path quickly in response to changes in the surroundings without affecting the overall navigation trajectory.

2 Literature Survey

- Paper B-spline path planner for safe navigation of mobile robots proposed a path planning algorithm using B - Spline curves by the reduction of the conservatism in the local convexity bound and in the integration of these constraints into a convex quadratic optimization problem, which minimizes the curve length.
- Paper Vision-aided UAV Navigation and Dynamic Obstacle Avoidance using Gradient-based B-spline Trajectory Optimization proposes gradient based B - Spline trajectory utilizing robot's onboard vision.The depth vision enables the robot to track and represent dynamic objects geometrically based on the voxel map.

3 Methodology & Results

3.1 Basic Implementation of path planning algorithm using B - Spline

- Implemented Gazebo Navigation Stack for navigating a robot which was left to travel in a maze.The robot is equipped with **odometry** & **laser** sensors for detecting the environment around it.Achieved good results by the algorithm implementation after running it for few iterations.

- Image 1 shows how robot travels in tha path & the sensors of the robot can only have upto limited distance which is called look ahead distance based on that distance algorithm plans the future path of the robot.

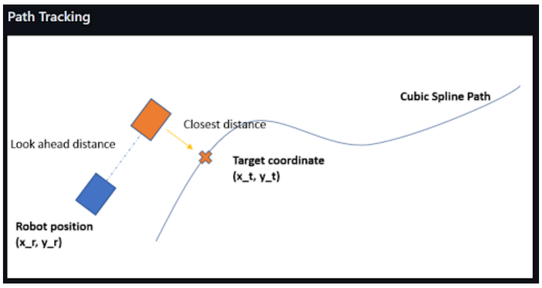


Figure 1: Path taken by robot following B - Spline curves

- Image 2 shows how the algorithm changes the path for the robot over iterations. Blue colored path indicates the intial path with which we start the algorithm & the red colored path is the path after few iterations of the algorithm.

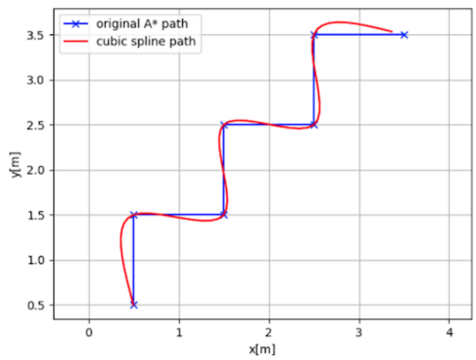


Figure 2: Refining the path of robot over iterations

- I have implemented UAV - Motion - control algorithm. This algorithm generates the path for the robot which matches with the global path in the case of no obstacles.

3.2 Optimizing path planning algorithm

- There is ongoing by Sanju sir in which algorithm has been designed for path planning of the robot. I have worked on optimizing the algorithm by tweaking various values.
- The below algorithm - 1 is a small part of optimizing the path planning algorithm. there we can see when the several cases for handling the drone when it detects a obstacle which is D distance apart through it sensors, then drone is directed according to the value of D.

Algorithm 1: B-Spline Optimization Algorithm

Input: V (Velocity), A (Altitude), GPS, IMU (Attitude), C (Position)
Output: D (Distance), L (Latitude), Y (Yaw), A (Altitude)
// Initialization
if $D < Threshold$ **then**
  Starting Initial control points
end
else if $D > Threshold$ **then**
  Avoid the obstacle
end
else if $D = Threshold$ **then**
  Default acceleration
end
else
  Drone moves towards the destination according to the given source to destination
end

3.2.1 Experiment 1 - Tweaking Control Points

- Control points are the ones through which we had the command over the B - Spline curves, Having more control points indicates the more command over the graph.
- Image 3 depicts the path followed by the drone, in the image Violet Path indicates the ground truth path & Blue Path indicates path followed by drone till now & Green Path indicates the future path predicted by the algorithm.

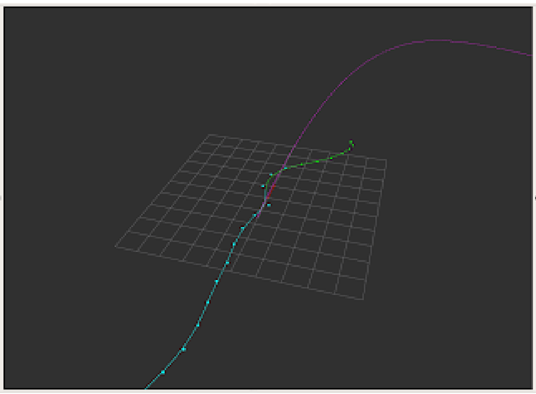


Figure 3: Path followed by the drone

- Image 4 shows that the path given by the algorithm is matching with global path when there are no obstacles.

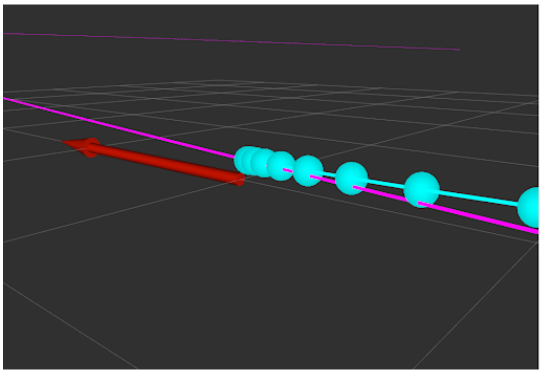
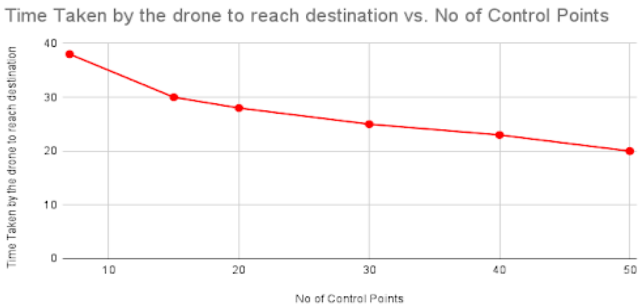


Figure 4: Global path matching with optimal path

- I have experimented by increasing number of control points of the curve & I have found that with **increase** in number of control points time taken for the drone to reach the destination **decreases**.

No of Control Points	Time Taken by the drone to reach destination	Value of dt	Time Taken for the drone to start
7	38 sec	0.5	7 sec
15	30 sec	0.5	9 se
20	28 sec	0.5	10 sec
30	25 sec	0.5	13 sec
40	23 sec	0.5	15 sec
50	20 sec	0.5	20 sec

(a) Variation of time taken with control points



(b) Time Taken vs No of Control Points

3.2.2 Experiment 2 - Tweaking Velocity Gains

- Velocity gains are ones which are responsible for the speed of the robot. Increasing them will make the robot faster.
- I have experimented with increasing the velocity gains & found the changes in drone trajectory with the changes & also found that time taken to reach destination is **decreasing** with **increase** in velocity gains.

Velocity Increment	Time Taken by the drone to reach destination	Value of dt
2	27.3 sec	0.5
5	24 sec	0.5
7	20 sec	0.5

Figure 6: Time taken vs Velocity gains

- **Failure Case:** I have observed that increasing the velocity gain to certain extent, the drone is crashing beacuse the drone is not able to high speeds.

3.2.3 Experiment 3 - Refining algorithm for dynamic obstacle handling

- I have worked with Sanju sir for improving the path planning algorithm, so that the drone can handle the dyanmic obstacles which were introduced while the journey of drone.
- **Drawbacks:** Even after improving for some cases the drone is being crashed by hitting some of the obstacles. This case was araised when a lot of dynamci obstacles were introduced.

4 Findings & Conclusion

- Increment in number of control points & velocity gains results in a better path planning algorithm.
- We can even more optimize the algorithm by tweaking other parameters (Work is still going)
- We have also improvised the algorithm which handles the dynamic obstacles.

5 References

1. Ngoc Thinh Nguyen and Lars Schilling, "**B-spline path planner for safe navigation of mobile robots**" *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*
2. Zhefan Xu and Yumeng Xiu, "**Vision-aided UAV Navigation and Dynamic Obstacle Avoidance using Gradient-based B-spline Trajectory Optimization**" *2023 IEEE International Conference on Robotics and Automation (ICRA 2023)*
3. **Gazebo Navigation Stack**: https://github.com/zhidateh/gazebo-navigationstack
4. **UAV Motion Control**: https://github.com/Logan-Shi/UAV-motion-control/tree/time-scaling