Aerial Robotics Kharagpur Documentation Template
Dynamic pathplanning

Dynamic Path Planning using rrt:
Rapidly-exploringRandom Tree is used to find the path from start to goal where the obstacles are dynamic. When drones fly, there encounter a lot of obstacles, both static and dynamic.RRT is a fast algorithm and helps in finding path in dynamic environment

I. INTRODUCTION

For Dynamic path planning i used rrt. For each frame of the video, the algorithm is applied. The moving obstacles create a difference in the various frames and the algorithm generates a random line from a certain point of a specific length checking whether the line goes through obsctacles. If it does then ignore that line. I have tried Dynamic window approach but that appeared slower. RRT with a small paath length was taking a lot of time and the parameters should be handled well.

VI RESULTS AND OBSERVATION

The algorithm used is slow and can improvised by various methods, like starting the algorithm from both start and end. The problem with my algorithm is that the length of a line is small and the range of random point generation is limited and that is the reason it is taking a lot of time.Dynamic window approach and RRT in optimised forms can be used for better results

VII FUTURE WORK

My future work includes optimising my approach for the path planning by working on the length on path required and reducing the time for computation. Random points should be a little more scattered. I wish to learn and work on graph theory.That might help in better path planning

CONCLUSION

Path planning is a basic requirement for autonomous vehicles and there is a lot of scope for learning in this field and unlike theory there will be a lot of practical difficulties which should be tackled with experience Computational speed and randomising are the problems faced For ARK, path planning should be having little error since the bots will never be in our “hands”. Some **obstacles** even if hit can cause a little damage to the bot. So we can assign penalties and make things like a game

For ARK path planning are where things start and in a dynamic environment improvising the existing algotihms is very necessary.

II. PROBLEM STATEMENT

The given problem statement is on planning a path in dynamic environment where the obstacles move. In the video feed, the blue obstacles move and plan has to be made for a path from start(red)ot end(green).

IMAGE ->Fig1

the black walls are to be avoided.

III. RELATED WORK

Dynamic window approach was thought of where the robot motion parameters are given and an arrow moves on the image making sure it avoids the obstacles and reaches the goal. The website is mentioned in the **REFERENCES**

fig 2

IV. INITIAL ATTEMPTS

I wanted to scan the entire image and get the centre and radius of the obstacles and their position as a function of time using contour detection .And for the walls their parameters using hough transform But in reality, all the 3D projections on 2D need not have the same radius so the idea was rejected. Then i learned about dynamic window approach and that seemed a good idea with the robot parameters given, i move an arrow on the image to find the path. But the robot paramters should be chosen properly and a lot of practical difficulties occur due to inertia and stuff and it should be optimized a lot

REFERENCES

-> OpenCV documentation by several authors

-> https://ieeexplore.ieee.org/document/6108732

-> https://www.hindawi.com/journals/complexity/2018/8420294/

->https://github.com/AtsushiSakai/PythonRobotics(Dynamic window)

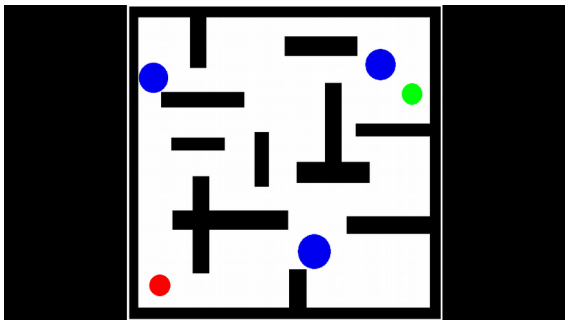
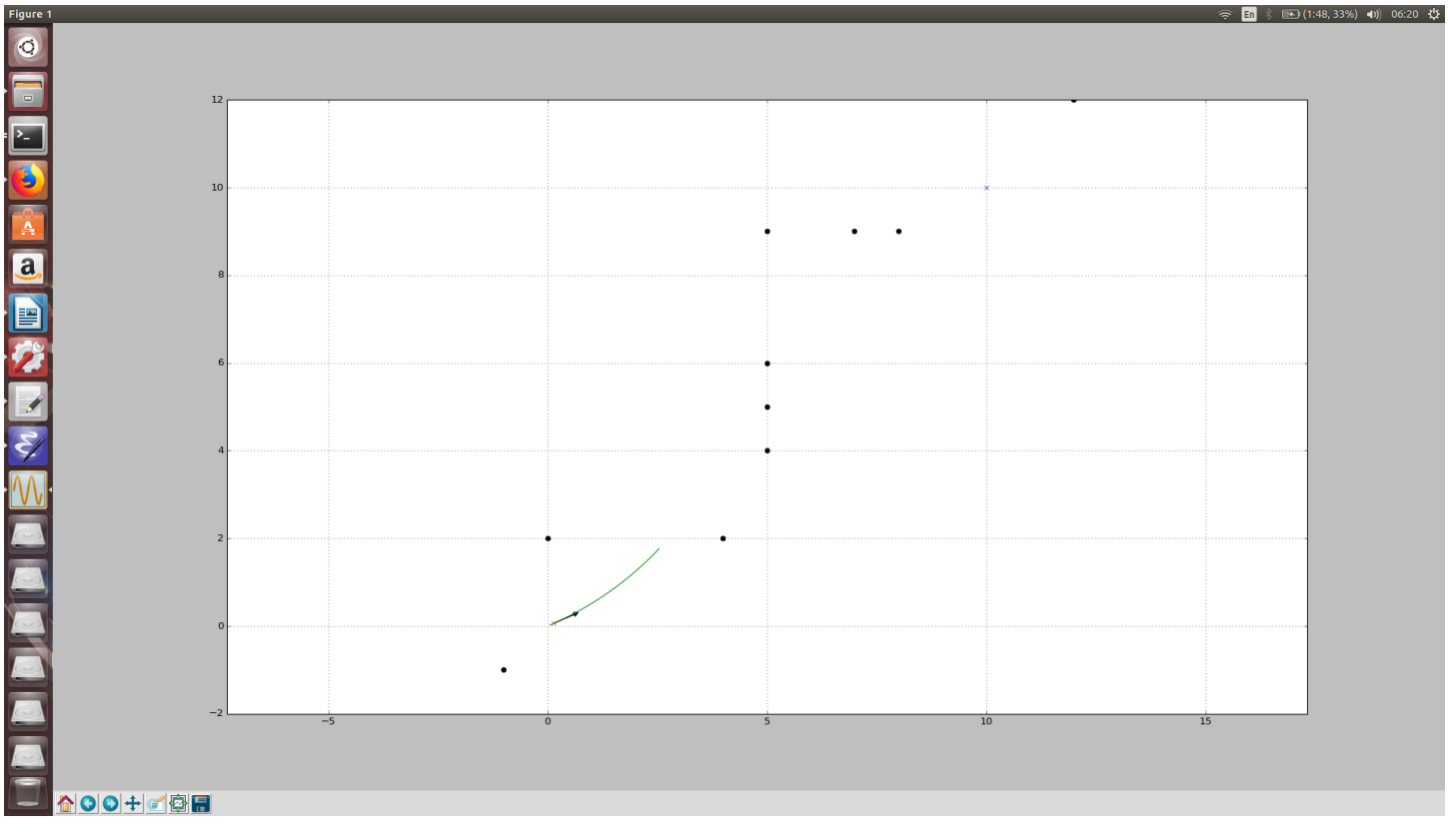


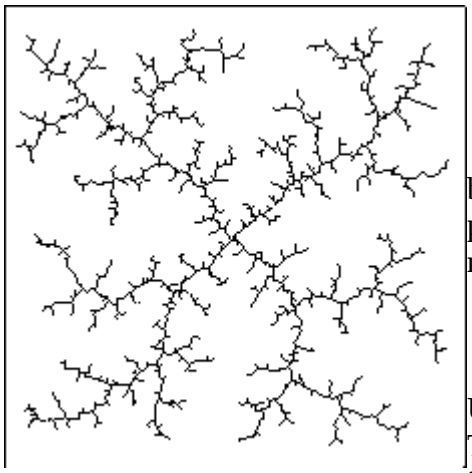
Fig1

Fig 2 \ /



V FINAL APPROACH
RRT

A Rapidly-exploring Random Tree (RRT) is a data structure and algorithm that is designed for efficiently searching spaces.RRTs are constructed incrementally in a way that quickly reduces the expected distance of a randomly-chosen point to the tree. RRTs are particularly suited for path planning problems that involve obstacles.
Usually, an RRT alone is insufficient to solve a planning problem. Thus, it can be considered as a component that can be incorporated into the development of a variety of different planning algorithms.
Pseudo code: Generate a random point p
compare with the existing points(in some array a) for which is the closest
min(a[i],p)
now plot a line from a[i] to p
move by a fixed distance d
and keep generating random points till it reaches the goal



The initial errors faced were with the length of the line which was too small compared to the size of image. It was not visible. So I had to use a bigger length value. The line is plotted in grey and once the frame is skipped it disappears. So i had to plot it on a black image to get a plot like Fig 3. In my code i kept storing the points in an array along with their parent so that once the goal is reached i can traverse back and plot one line from end to start
RRT appears in various optimised forms.

Usually, an RRT alone is insufficient to solve a planning problem. Thus, it can be considered as a component that can be incorporated into the development of a variety of different planning algorithms.

^Fig 3