Motion Planning in Dynamic Environments

Yegeta Zeleke, Katherine Driggs-Campbell, Ruzena Bajcsy, and Shankar Sastry

Abstract

Motion planning for risk calculation can be used in autonomous systems to yield better dynamic control than traditional methods. The overall goal of this project is to create a safe prediction and path calculation based on system inputs and vehicle dynamics. Sampling- based algorithms such as rapidly exploring random trees (RRT) and probabilistic Roadmaps (PRM) are widely used for motion planning and are proven to provide probabilistic completeness. The motion planning in this project is obtained by using RRT algorithm, which provides feasible trajectories for systems with differential constraints and non-holonomic dynamics.

Background

Today most modern vehicles are equipped with wheel slip control systems and advanced sensing systems that will monitor the driving situation. Modeling dynamic system is challenging for nonholonomic systems in safety critical and dynamically changing environments. One way to ensure safety is to implement robust motion planning algorithms. However, these sampling-based algorithms often produce suboptimal, non-smooth solutions.

Objective

The goal of this project is to generate safe trajectories in uncertain, dangerous environments using traditional motion planning techniques.

Problem Statement

Given a start point, the dynamics of the vehicle, and obstacle information, motion planning algorithms find a feasible trajectory from the start position to the goal position while obeying the rules of the environment, and avoiding obstacles.

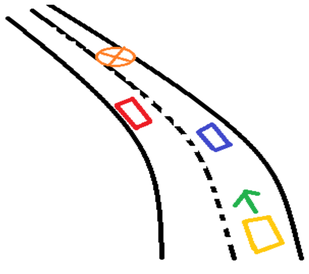

Fig. 1: Image from dd3d.hr

Method

Vehicle dynamics:

$$\dot{x} = u_s cos\theta$$
$$\dot{y} = u_s \, sin\theta$$
$$\dot{\theta} = \frac{u_s}{L} \tan u_\emptyset$$

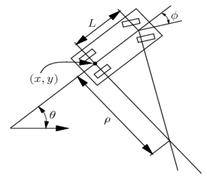

Fig. 2: Simple Car [1]

Algorithm : RRT

V← {Xint}; E←∅;
For i = 1…n-1 **do**
nodes ← generate_nodes(path,u_s_rand,u_phi_rand,i);
Cflag ← collisionchecker(nodes,road_bnds,obstacle);
if(Cflag)
$nodes_{pruned}$ ← deletenode(Cflag,nodes);
endif
Vi+1 ← nodespruned;
Ei,i+1 ← nearest(V,E);
path ← generate_path(V,E);
return path, V, E
Where: n = number of samples
V = vertices in the graph
E = edges in the graph
Xinit = initial position of vehicle
path = generated trajectories

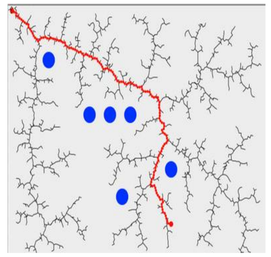

Fig. 3: RRT Example. Image from nakkaya.com

The following procedure was used to implement RRT:

- Generate nodes
- Avoid obstacle and stay within road bounds
- Generate trajectory

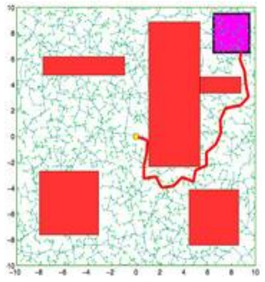

Fig. 4: Avoiding Obstacles. Image from mit.edu

Disadvantages

While motion planning has the advantage of being able to handle the complex constraints, it has a number of drawbacks:

- Sampling based methods generate non-smooth paths
- Time to find solution may vary depending on constraints
- Although there are theoretical guarantees, path may be suboptimal when implemented in real-time
- The algorithm might return no solution for a problem with a bottle neck constraints.

Results

In every iteration, new vertices on the free space are generated to create a potential trajectory.

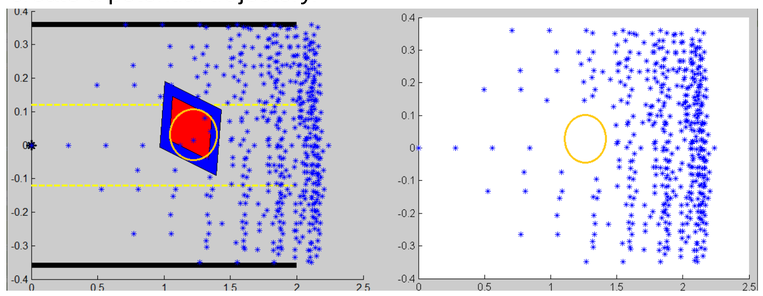

Fig. 5: Generate nodes and perform collision check

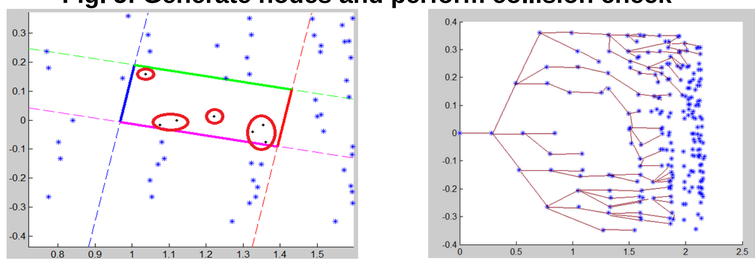

Fig. 6: Identify collisions and connect nodes to create a series of potential trajectories.

Conclusion

Using RRT, we were able to generate trajectories with in a given time horizon that avoids obstacle and obey the rules of the road. While this implementation has its disadvantages, we can pass the RRT trajectory to a control algorithm like model predictive control (MPC) to execute an optimized and smooth path

Future Work

To build a complete and provably safe system, we plan to:

- Optimize end point selection for the motion planning algorithm
- Implement path smoothing with MPC for optimality
- Implement in real-time with more realistic dynamics, as well as more, possibly asymmetric obstacles

Reference

1. L. E. Dubins. On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents. American Journal of Mathematics, vol. 79, no. 3, Jul.