Path Planning Project

The project is to design and implement path planning solution in C++ for a car driving on a 3-lane highway. The project includes various aspects like

1) Perception of surrounding environment with processing of sensor fusion data
2) Behavior prediction of surrounding vehicles
3) Behavior planning of host vehicle while within the constraints of road rules, passenger comfort and vehicle safety
4) Trajectory planning of host vehicle based on the chosen behavior.

The constraints imposed on the vehicle are

1) Vehicle speed can't exceed 50 MPH
2) Vehicle acceleration and jerk don't exceed 10 m/s2 and 10 m/s3.
3) No collisions
4) Vehicle stays within lane markers except when changing lanes.

Localization

The starting point for path planning is to locate the host vehicle. First, I read sensor data for the host vehicle. I convert speed to m/s unit from mph since all the car positions are in meter. The d coordinate data is used to determine current lane of the host vehicle since lane lengths are known (4 m).

Perception

In highway driving, it is important to know the lanes of the surrounding vehicles. I extracted lane info for all the surrounding vehicles. Also, their speed is calculated.

Behavior Prediction

Once the lanes of the surrounding vehicle are known, it is simple to predict their behavior since vehicle can only move forward or stop in the highway driving. For more complex environment, a Bayesian or model based prediction would be needed but for highway driving, a simple approach is good enough.

I predict their future location by projecting their current speed in the current lane. This future location is used to decide whether the lanes are blocked on not for the host vehicle. Three Boolean flags: front_blocked, left_blocked, and right blocked are set depending on the distance between predicted position of the host vehicle and surrounding vehicles.

The distance between cars in current lane should be more than the maximum distance a car can travel at max velocity in the given time period. For left and right lane, there is an additional buffer of 10 m to the distance threshold because of additional maneuver of lane changing.

Behavior Planning

There are five possible behaviors (states) for the host vehicle

1) Maintain the speed
2) Accelerate up to max speed
3) Decelerate to avoid collision
4) Change to left lane
5) Change to right lane

Rather than using cost-based approach for behavior selection, we can use rule-based methods based on the three Boolean flags from the “Behavior Prediction” section above. The default rule is to drive at maximum velocity, set at 22.2 mps, which is converted from 50 mph. If front lane is blocked, then try to change lane to left if left is not blocked, or to the right lane if right lane is not blocked. Finally, if all lanes are blocked, slow the vehicle gradually till there is enough distance to front vehicle.

I chose 0.19 m/s as incremental step for increase and decrease in the speed because it satisfies acceleration to be less than 10 m/s2 during any 20-millisecond time-step.

Trajectory Generation

To generate the trajectory of the selected behavior, I wanted to use quintic polynomial approach that minimizes jerk. However, such approach would require to know car’s acceleration term from simulator. Since that was not possible, I started with last two points of the previous trajectory. To these two points, I added three points at 30, 60 and 90 m distances and create a spline function.

Spline helps us interpolate the necessary points for trajectory. I conservatively chose 25 m as reference distance in x direction. The number points would depend on the reference speed selected by the behavior. Trajectory with these additional points (total 50 points) is sent simulator.

Results

Results from the simulator show that the host vehicle can drive the entire loop without any incidents while staying within constraints. It executes multiple lane changes wherever feasible.