Path Planning Project

Requirements:

Use sensor fusion data, and localization data to generate valid trajectories that would allow the main vehicle to drive according to the speed limit, stay under $10m/s^2$ of acceleration and $10m/s^3$ of Jerk, avoid collisions, and is able to change lanes.

Solution:

The simulator used requires 50 points to be sent per iteration, where each point is the location of the vehicle for 0.02s.

In order to handle latency the simulator, and the generated trajectories, previously generated and unused path points are used to determine the location of the vehicle more accurately for given time.

Trajectory Generation is done in Frenet Coordinates due to the simplicity of being able to mathemalticaly describe a trajectory on any road, however the trajectory sent to the simulator, needs to be in Cartesian Coordiantes.

Converting between Frenet Coordinates and Cartesian Coordinates is done using with the use of waypoints. Waypoints from the highway are read from the csv. In order to lower the error associated with the non-linear conversion from one coordinate to the other, I interpolated 5 waypoints behind, and 5 waypoints ahead of the vheicle.

In my implementation, reading the waypoints and interpolating waypoints in the vehicles vacinity is done by the Map class.

Each iteration of the program, updates the vehicles local waypoints, gets the position, speed, and accelaration of the vehicle in Frenet Coordinates.

Afterwards, Trajectory Predictions are made for other vehicles whose positions and speeds are acquired with sensor fusion data, the current state of the vehicle is used to determine what possible next states we can transition to and determine an associated end point in frenet coordinates, and then generate a trajectory from the current location to the enpoints of each possible state, and lastly determine which of the trajectories is the most optimal with a cost function.

With an optimal endpoint and trajectory determined, we can produce a path to this optimal location in Cartesian coordinates to append on to previously generated path points to make a total of 50 points sent to the Simulator.

Detail on each Component are described below.

Prediction:

An accurate prediction of each vehicles position during the duration of the trajectory we are generating is necessary. With a trajectory of other vehicles, we can determine if the trajectory we are generating is colliding with them.

The model I used to predict other vehicles positions is a constant velocity model. Other more accurate models can be used, but in principal predicting the position of other objects in the future won't be accurate over long horizons, so I tried to keep my prediction horizon small.

Behaviour Planning:

I used 5 states to represent the main vehicle, a Keep in Lane state, a Prepare to Change to Right state, a Prepare to Change to Left state, a Change to Right, and a Change to Left state.

A keep lane state can only transition into another keep in late state, and a prepare to change right or left state. The Change to left or right state will can only be entered from a prepare to change state.

For each iteration, the current state is used to determine the next possible states. With the next possible state we can determine a trajectory from the current position of the vehicle to a desired point.

Trajectory Generation:

A Jerk Minimizing Trajectory is created to connect an intial state to a final state. A Jerk Minimizing trajectory is a great algorithim for the given enviornment of a highway. The enviornment is strucutred with a standard set of rules that make it great for a Jerk Minimizing Trajectory as opposed to A* and other alogrithims. Once a trajectory is generated for a given state, a cost function is used to determine wheter or not one trajecotry is more optimal than another.

Cost Function:

The cost function I used to determine wheter one trajectory is more optimal than another is made of many different factors.

Collision Cost – Applies a cost to the trajectory if it is colliding with the trajectories of other vehicles.
- The most important cost and is therefore the one that has the highest factor associated

Inefficiency Cost – Applies a cost to trajectories that have an intended, and final lane with a slower average speed. In a given situation where the main vehicle is in the center lane, which is slowing down, and the left lane is much slower than the right, the prepare to change to the left trajectory will have a higher cost associated than a prepare to change to the right trajectory

Speed Cost – Applies a cost based on the final speed of the trajectory. A higher speed is prefered to reach the destination quicker.

Distance costs – A distance cost to any other vehicle, a cost to other vehicles in the same lane, and the distance to the goal are also used to determine an optimal trajectory.

Not middle lane cost – A cost is also applied to trajectories that are not in the center lane. In the situation that we are driving on a clear highway on the left lane, we would prefer to be in the middle lane so that we can handle situation with more flexibility by switching to the left or the right lane.