Path Planning Project Report

November 27, 2017

1 Path Planning Project

My path planning consists of two components: behavior planning and trajectory generation.

1.1 Behavior Planning

I first decide high level behavior (keep lane, change lane change right, lane change right).
I use the following simple rules to decide behavior. There is no particular goal lane in this project, then I decided to apply simple rules rather than using a cost function because it is intuitive and close to my decision making of driving. Let l_{front} is the distance to a car that is in front of my vehicle.

- If $l_{front} >= 30m$, my car keeps the current lane.
- If $l_{front} < 30m$, my car keeps the current lane or changes lane based on safety checks of lanes.
- For each lane, I compute the distance (only s value in Frenet) between the future position of a car in the lane and the future position of my car. If the distance is less than $15m$ at any moment, I consider the lane is not safe.
- For each lane, if the current distance (in only s) between other cars and my car is less than $15m$, I consider the lane is not safe.
- Only when the lane next to the current lane is safe, the car changes lanes.

1.2 Trajectory generation

This part is the same as the one used in the project walk-through video. I take a spline approach instead of a jerk minimization approach. Investigation of jerk minimization method is future work. However, the spline approach is good enough for this project and computationally efficient.

Let $p^F = (s, d)$ be a point in Frenet coordinate and $P_t^F = (p_0^F, p_1^F, p_2^F)$ be future waypoints in Frenet coordinate computed at time t. Let $p^C = (x, y)$ be a point in the Cartesian coordinates and P_t^C be waypoints in the Cartesian coordinates. P^F is computed based on the high level behavior. Let $p_t^F = (s_t, d_t)$ is the current my car's position. Let d_r be the center position of the right lane, d_c be the center position of the center lane, and d_l be the center position of the left lane. Let $d_{current}$ is the center of the current lane. I compute P_t^F using the following rules.

- $P_t^F = ((s_t + 30, d_{current}), (s_t + 60, d_{current}), (s_t + 90, d_{current}))$ if the car decides to keep the current lane.
- $P_t^F = ((s_t + 30, d_r), (s_t + 60, d_r), (s_t + 90, d_r))$ if the car decides to go into the right lane.
- $P_t^F = ((s_t + 30, d_l), (s_t + 60, d_l), (s_t + 90, d_l))$ if the car decides to go into the left lane.
- $P_t^F = ((s_t + 30, d_c), (s_t + 60, d_c), (s_t + 90, d_c))$ if the car decides to go into the center lane.

I convert P_t^F to P_t^C. Note that the frame for this step is the reference frame (car frame). Then, we append P_t^C to the last two points of the last path to form anchor points for spline. Then, the number of anchor points is five. I compute spline with these five anchor points. Using the spline function $f(x)$, we create a path that consists of 50 points. If M points in the previous path remain, I append $50 - M$ points to the previous path (in reference frame) to generate a new path $((x_0, y_0), \cdots, (x_k, y_k), \cdots (x_{49}, y_{49}))$. Let $p_{target} = (30, f(30))$ be a target point. The path is computed as follows.

$$\begin{aligned} l_{target} &= ||p_{target}||_2 \\ N &= \frac{l_{target}}{0.02 \times \frac{v_{ref}}{2.24}} \\ x_{k+1} &= x_k + \frac{30}{N} \\ y_{k+1} &= f(x_{k+1}) \end{aligned}$$

where v_{ref} is the reference velocity. Then, I transform these points to the world frame.