Path Planning

Udacity Self Driving Car project #12

Goal

Goal is to drive the ego vehicle safely and as fast as possible around a track with other vehicles.

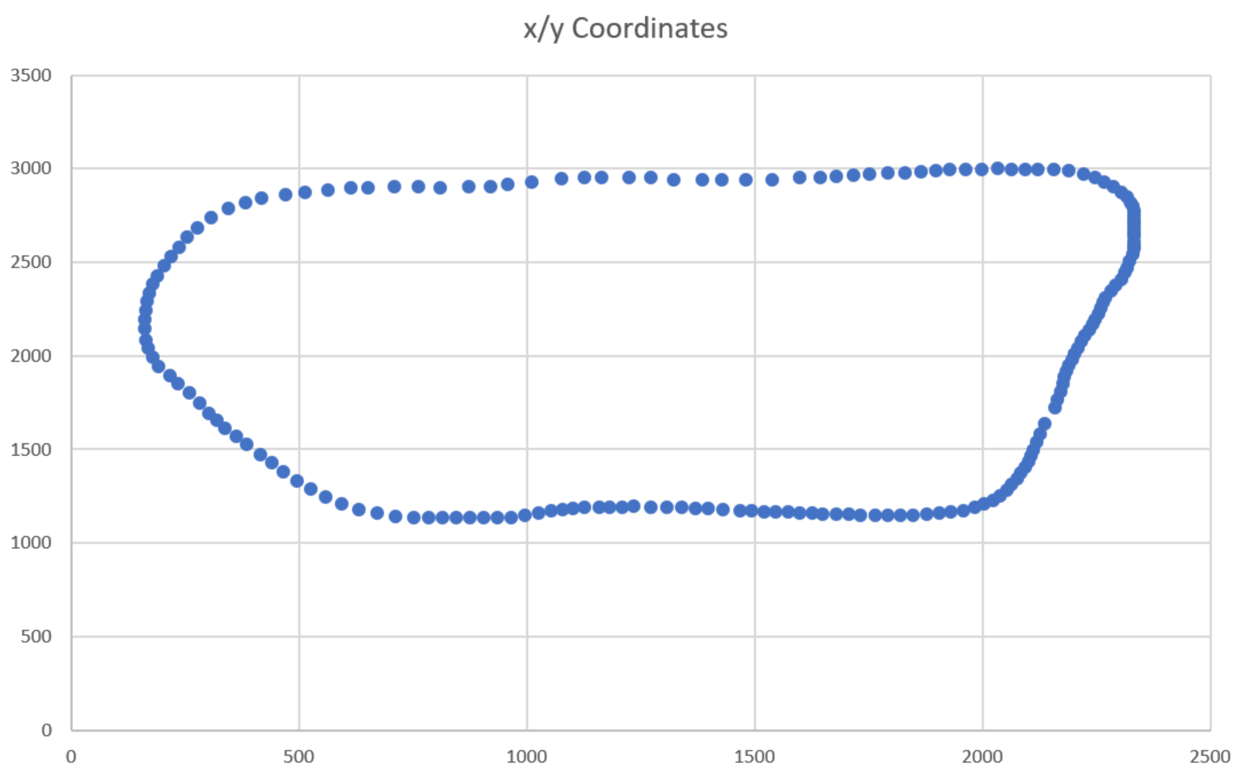
Observations

The simulator is an ideal driving situation on a three-lane highway with no entries nor exits, without long vehicles such as truck or buses, and without the potential for traffic jams. There are no fixed obstacles on the road.

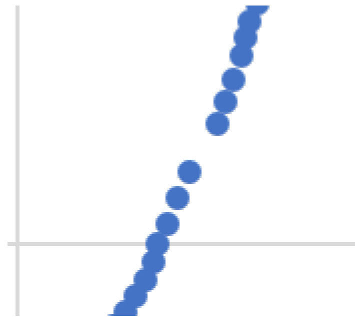
All of which are considered for the approach, and any change in the observations above would invalidate (portions of) the approach.

Map

The x/y coordinates outline the track yet visually they are not evenly spaced, they appear to reflect a racing car and its velocity.



At start and end point, the track does not appear to link up:



The map used is based on the waypoints provided with a spline overlaid. The vehicle will follow the spline and not attempt to hit waypoints provided.

Approach

At given time intervals, e.g. every 200ms, we execute a planning cycle, which execute the steps of

1. Trajectory Generation
2. Trajectory Pruning
3. Collision detection
4. Trajectory Selection

All steps are completed in Frenet Frame. It is expected the be less expensive to translate the expected low number of other vehicles onto a Frenet frame for Collision Detection, than it is to translate all non-pruned trajectories onto a cartesian frame. Thus, there is one more step to complete: translate the selected trajectory back onto a cartesian frame.

Trajectory Generation

During trajectory generation, the algorithm produces a manifold of trajectories available to the ego vehicle.

We split any trajectory into its lateral and longitudinal component where the lateral movement determines staying in a lane or changing lanes, and the longitudinal movement determines speed and acceleration along the road. See the proposed approach in Werling et al {2].

When below a certain speed, the vehicle dynamics change for lateral movements, and thus we do no attempt any lateral movement, i.e. stay in lane and gain speed, which closely mimics human behaviour on a highway where everyone else drives faster. The simulator does not feature a traffic jam scenario, where it may be a good strategy to change lanes at low speed, there the assumption is reasonable.

We discretise the target points to arrive at a limited number of potential trajectories: laterally to a given set of target points; longitudinally with a given set of time points.

The combination of lateral and longitudinal points provides a set of target points in the Frenet frame, each of which can be modelled with a quintic polynomial to be reached from a given starting position.

Mathematics

For both lateral and longitudinal movements a quintic polynomial can satisfy all requirements:

$$x(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3 + c_4 t^4 + c_5 t^5$$

With $c_0, c_{1,}, c_2, c_3, c_4, c_5$ being the optimal coefficients to solve for with initial and final conditions and the goal to minimize jerk.

$$\min \int_{t_0}^{T} \big(j(t)\big) \quad dt =$$
$$\min \int_{t_0}^{T} \big(j(t)\big) \quad dt =$$
$$\min \int_{t_0}^{T} (6\,c_3 + 24\,c_4\,t + 60 * \ c_5 t^2)\,dt =$$
$$\min(20\,c_5\,T^3 \ + \ 12\,c_4\,T^2 \ + \ 6\,c_3\,T - \ 20\,c_5\,{t_0}^3 \ + \ 12\,c_4\,{t_0}^2 \ + \ 6\,c_3\,t_0) =$$
$$\min(20\,c_5\,(T^3 - {t_0}^3) + \ 12\,c_4(\,T^2 - {t_0}^2) + \ 6\,c_3(\,T - t_0)\,)$$

Lateral

In a Frenet frame, lateral target velocity and acceleration are 0. The goal is to drive parallel to the lane at the end of an execution

$$v_{d_{target}} \ = \ \dot{d}_{target} = 0$$

$$a_{d_{target}} \ = \ \ddot{d}_{target} = 0$$

Lateral target position is one of the lane centers (assuming three lanes) since we want to end up nicely centered in a lane:

$$d_{target} \ \in \{center_{left}, center_{middle}, center_{right}\}$$

The exception is if the go vehicle is travelling below a given minimum velocity, e.g. 15mph, we will not attempt a lane change and therefore

$$d_{target} \ = \ center_{initial} \ if \ \dot{s}_{initial} < \ Min_v$$

Under the reasonable assumption that we start and accelerate in close to a given lane center.

The related cost function is (Werling M et at [2]):

$$C_d \ = \ k_j * J_t + \ k_T * T + \ k_d * (d_{Trajecory} - \ d_{target})^2$$

Total Jerk J_t is

$$\int_{t_0}^{T} (j(t))^2 dt = \int_{t_0}^{T} (6\,c_3 + 24\,c_4\,t + 60 * \ c_5 t^2)^2 dt =$$
$$36{c_3}^2 T + 240\,c_3 c_5\,T^3 + 144\,c_4\,c_3\,T^2 + 720\,{c_5}^2\,T^5 + 720 c_4 c_5\,T^4 + 192{c_4}^2 T^3) \text{ -}$$
$$36{c_3}^2 t_0 + 240\,c_3 c_5\,{t_0}^3 + 144\,c_4\,c_3\,{t_0}^2 + 720\,{c_5}^2\,{t_0}^5 + 720 c_4 c_5\,{t_0}^4 + 192{c_4}^2 {t_0}^3)$$

Longitudinal

There are multiple scenarios for longitudinal target position

1. Velocity keeping (no vehicle ahead or too far away)
2. Following (keeping distance to vehicle ahead)
3. Merging (target position is between two vehicles in adjacent lane)
4. Stopping at a certain point

For this project, only velocity keeping and following are considered. Merging may be a future enhancement and stopping in the given simulator is only a valid scenario if a leading vehicle stopped which will then turn it into a following scenario.

Following

According to Werling M et at [2], the target position for following, is to have a constant time gap to the leading vehicle at a given time t:

$$s_{target} = s_{leading} - [D_0 + \tau * \dot{s}_{leading}]$$

$$\dot{s}_{target} = \dot{s}_{leading} - \tau * \ddot{s}_{leading}$$

$$\ddot{s}_{target} = \ddot{s}_{leading}$$

With D_0 being a safety distance and τ the required time gap, for example

$$D_0 = 5m$$

$$\tau = 2s$$

We will reasonably assume, that for a short time horizon (less than 1 second), all other vehicles will not accelerate, i.e. $\ddot{s}_{leading} = 0$, which leads to the following equations

$$s_{target} = s_{leading} - [D_0 + \tau * \dot{s}_{leading}]$$

$$\dot{s}_{target} = \dot{s}_{leading}$$

$$\ddot{s}_{target} = 0$$

Maximum jerk is at (fourth derivative set to 0):

$$t = -\, {}^{c_4}\!/_{c_5}$$

Maximum acceleration is at (third derivative set to 0):

$$t = \frac{-24 * c_4 \pm \sqrt{576 * c_4 - 1440 * c_3 * c_5}}{120 * c_5}$$

The related cost function is (Werling M et at [2]):

$$C_v = k_j * J_v + k_T * T + k_s * (s_{trajectory} - s_{target})^2$$

Total jerk J_v is the same formula as total jerk J_t for the lateral movement, only with the longitudinal coefficients.

Velocity Keeping

For velocity keeping, the target velocity is the given speed limit Max_v and target acceleration is 0.

$$v_{s_{target}} = \dot{s}_{target} = Max_v$$

$$a_{s_{target}} = \ddot{s}_{target} = 0$$

According to Werling M et at [2], for velocity keeping the target polynomial is quartic (not quintic as in all other cases) as the target position is arbitrary, i.e. $c_5 = 0$

Jerk is linear for a quartic polynomial, and depending on the sign of c_4, jerk is at maximum either at the initial time t_0 (negative sing) or the final time T (positive sign).

$$j = 24c_4 t + 6$$

Maximum acceleration is at (third derivative set to 0):

$$t = -\ {}^{c_3}\!/_{4 * c_4}$$

The cost function depends on the target velocity $\dot{s}_{target}$ and not the target position s_{target}:

$$C_v = k_j * J_v + k_T * T + k_{\dot{s}} * (\dot{s}_{trajectory} - \dot{s}_{target})^2$$

Observations

With three lanes and movements discretised to those three lanes, there always a limited number of trajectories. For example, for three lateral points and 10 timesteps, there is a set of 30 trajectories

At t=0 (initial state), there needs to be a test if the ego vehicle is already following or at speed limit. For following that means we are comfortably in a lane and have reached a target position and velocity behind a leading vehicle.

$$s_{initial}\ ?= s_{leading} - [D_0 + \tau * \dot{s}_{leading}]$$

$$\dot{s}_{initial}\ ?= \dot{s}_{leading}$$

$$\ddot{s}_{initial}\ ?= 0$$

$$\dot{d}_{inital}\ ?= 0$$

$$\ddot{d}_{initial}\ ?= 0$$

If so, there is no need to generate any more trajectories for this lane. Should the leading velocity $\dot{s}_{leading}$ have changed since the last observation, the above equations will not hold true and a new set of trajectories will be generated.

Trajectory Pruning

During simulation we prune trajectories where

- Projected velocity, acceleration, or jerk are unacceptable for the trajectory

- Projected velocity, acceleration, or jerk are unacceptable for the trajectory when combined with previous trajectory

Projected velocity should not exceed the given speed limit based on the proposed trajectory generation.

Acceleration is at maximum or minimum where jerk (equal to acceleration dot) is zero; jerk is at maximum where jerk dot is zero.

In the event of having no valid trajectory at the end of the trajectory pruning phase, we can assume that we face an emergency, i.e. there must be a slow vehicle in front or a vehicle is on collision course and we adapt a conservative lane keeping strategy irrespective of maximum jerk and acceleration, which may result in an emergency braking manoeuvre. This is the expected behaviour for a human driver.

Trajectory Costing

For all viable trajectories, apply a cost function to the remaining trajectories. Cost is defined by the combined cost function of lateral and longitudinal trajectories (see above).

Collision detection

For all available trajectories, find the best trajectory (minimum total cost), and check if there is a potential collision with another vehicle. There is no need to check for stationary obstacles, the simulator does not include any.

Available data is a set of other vehicles:

- Position in cartesian coordinates
- Position in frenet coordinates
- Velocity in x and y direction, which determines absolutely velocity and heading

We represent each vehicle, including ego vehicle with a rectangle based on pre-configured vehicle length and vehicle width.

For all other vehicles, we make the reasonable assumption that their short-term acceleration is 0, i.e. they keep their speed. Therefore, we only need the vehicles current velocity, position, and heading to determine its future position. Those parameters are available from the simulator. A complex model to determine the future position will only give the illusion of being able to predict the future position which is not determined by a vehicle's past behaviour, while such a model greatly increase the compute time required.

To cater for uncertainty of the vehicle's behaviour, we propose to increase the rectangle length based on time and velocity, e.g. the higher the velocity and the longer the time, the longer the rectangle representing the vehicle.

At maximum speed of 50 mph, a vehicle will move at 22ms. The proposed algorithm will check for collision from time 0 (initial state) to time T (available from trajectory generation) in small enough steps to avoid a collision between 0 and T. The proposed time interval is 0.5s, which equates to 11m at maximum speed and is a reasonable trade-off between checking more often and thus use more computing power, and checking less often and potentially miss a collision.

Not all vehicles have to be checked for collision. Only the leading vehicles behind the ego vehicle will be checked. “Behind” is defined as longitudinal position being same or less than the ego vehicle at the beginning of the planning cycle. This is since a vehicle cannot jump through another vehicle, i.e. there is a maximum of three vehicles to be checked for a collision: one for each of the three lanes behind or next to.

For vehicles ahead, we will have already generated a strategy of following. “Ahead” is defined as longitudinal position being larger than the ego vehicle at the beginning of the planning cycle.

If there is a collision detected, choose the next best available trajectory and repeat until there is a collision free trajectory, or if no collision free trajectory is available.

If no collision free trajectory is available, implement an emergency stop, effectively a following lane-keeping trajectory with a target velocity of 0.

Hyperparameters

The following hyperparameters are set manually:

Cost function

- Weighting factors

Planning Horizon

- Time in seconds to plan ahead

Vehicle Prediction

- Growth Factor of vehicle length = distance of circles representing vehicle

Implementation

Initialization

Precalculate curvature of map waypoints for all waypoints required for planning in Frenet frame.

Observe

Process all sensor fusion data and split into vehicles ahead and behind for each lane. Only keep the closest vehicle ahead and behind, that is a maximum of six vehicles.

It is assumed that a vehicle’s velocity in a Cartesian frame is the same velocity in a Frenet frame. This is strictly speaking not true but it works as a good approximator as the expected difference is small.

For the ego vehicle, observe readings and generate current acceleration from previous points.

Planning Cycle

Trajectory Generation

If above a given minimum velocity, generate lateral and longitudinal trajectories:

- Laterally, check for each of the three lane centers
- Longitudinally, check for times 1s to 10s in 1s increments.

Otherwise, generate longitudinal trajectories for current lane only.

Trajectory Pruning

If there is more than one trajectory available. check maximum jerk and acceleration for each trajectory and dismiss any trajectory that exceeds the maximum values

If no trajectory is left, use trajectory for current lane, even if it exceeds maximum values.

Trajectory Costing

Calculate cost for all available trajectories. Sum lateral and longitudinal cost.

Collision Detection

For all available trajectories, check collision with all vehicles from observation phase in 0.5s steps. Dismiss any trajectories with a collision. Start with best (lowest cost) trajectory and continue until a collision free trajectory is found.

If no trajectory is left, use a velocity keeping trajectory for current lane with a target velocity of 0. This emulates an emergency brake.

Result

The ego vehicle moves smoothly and performs lane changes as required.

The is the occasional speed limit violation, which can only be due to the road curvature when translation frenet coordinates to cartesian coordinates. To minimize the occurrence, global speed limit is set to 20.5 m/s in frenet space.

References

[1] Shai Shalev-Shwartz, Shaked Shammah, Amnon Shashua *On a Formal Model of Safe and Scalable Self-driving Cars* 2017

[2] Werling Moritz, Ziegler Julius, Kammel Stoeren, Thrun Sebastian *Optimal Trajectory Generation for Dynamic Street Scenarios in a Frenet Frame* 2010 IEE International Conference on Robotics and Automation