Udacity Car Nanodegree Term 2
Model Predictive Control Project

Young Mo Kang

The model

- The simulator provides the vehicle position (px, py) in the global coordinate, i.e., the map coordinate; the simulator also provides the car's direction in the global coordinate
- The simulator provides the waypoint positions (ptsx, ptsy) in the global coordinate
- The waypoint positions are converted to the vehicle coordinate, where (px,py) is set as the origin (0,0) and phi is set as 0; they are saved as next_x_vals and next_y_vals vectors
- Coefficients for 3rd order polynomial (cubic) is calculated from the waypoints in the vehicle coordinate system
- For given x-coordinate from the waypoints, the polynomial-fit y-coordinate is evaluated and saved as mpc_x_vals, mpc_y_vals vectors for the predictive trajectory in the vehicle's coordinate system
- The CTE is simply calculated as the y-difference between the waypoint and the current position of the vehicle. Since both coordinates are in the vehicle's coordinate system, this result is approximated as polyeval(coeffs, 0)
- The error in psi (epsi) value is calculated as the difference in the predictive trajectory's angle and the current direction. Using the vehicle's coordinate system, this becomes simply atan(coeffs[1]), for any polynomial fit with 1+ order. This is because the current position of the vehicle is at the origin (0,0) in the vehicle's coordinate system
- State vector in the vehicle's coordinate system is fed into MPC::Solve method; the state vector consists of vehicle position x,y, direction psi, velocity, CTE, and epsi in the vehicle's coordinate system
- For calculation of the cost function, all of CTE, epsi, and v_ref values are used. To improve driving quality, coefficient of 0.1 is multiplied to v_ref cost
- Also, actuator values are included in the cost so that it minimizes the actuators
- Lastly, actuator value differences between two consecutive time intervals are incorporated into the cost in order to smooth out driving
- The calculated actuator output values are steering wheel angle and throttle value, both of which are in the range of [-1, 1]; these values are fed into the simulator

Timestep Length and Elapsed Duration & Latency

- Because there is latency of 100ms, any value of dt smaller than 0.1 will cause the vehicle to wobble
- Too large of dt value is not good because the vehicle's response is too late
- For this reason, I have set the elapse duration of dt = 0.1
- Similarly, too large of N will not drive the vehicle successfully, as there is too much time delay between changes in control and it is pointless to predict horizon too far ahead as there will be large environmental changes as the vehicle drives
- Too small value of N will not work due to large discretization error
- I experimentally chose N = 10 and seems to work well