Model Description

The model consists of the state of the vehicle. The state is comprised of the following:

- X position (x)
- Y position (y)
- Orientation (psi)
- Velocity (v)
- Cross track error (cte)
- Orientation error (epsi)

The actuator controls are modeled using the following two parameters

- Steering angle (del)
- Acceleration (a) -> Braking is modeled using negative values of acceleration.

Given the state at any time (t), the state at time (t+1) can be calculated using the equations

- x_t+1 = x_t + v_t * cos(psi_t) * dt
- y_t+1 = y_t + v_t * sin(psi_t) * dt
- psi_t+1 = psi_t + v_t/Lf * del_t * dt
- v_t+1 = vt + a_t * dt
- cte_t+1 = cte_t + v_t * sin(epsi_t) *dt // cte_t = f(x) – y
- epsi_t+1 = epsi_t + v_t/Lf * del_t * dt

Time Step length and elapsed duration

A default value of N = 25 and dt = 0.05 was initially chosen. The selected value led to large run times. The dt was increases to 0.1 and no degradation on accuracy was observed. The run time was still high and as such the value of N was subsequently decreased from 25 to 15 and then to 10. At N = 10 an dt=0.1, the project goals were met with fast execution speed.

MPC Preprocessing

The preprocessing includes converting from the map's coordinate system to the car's coordinate system using the equations below

- ptsx_v[i] = (ptsx[i] - px) * cos(psi) + (ptsy[i] - py) * sin(psi); //px -> x position of car, py -> y position of car, psi -> orientation of car

- ptsy_v[i] = (ptsy[i] - py) * cos(psi) - (ptsx[i] - px) * sin(psi);

Handling Latency

Latency is handled by adding a state prediction using a time step of 100 ms.

The vehicle models are used to update the state before invoking MPC.Solve(). The equations below are used.

- px = px + v * cos(psi) * latency_in_sec;
- py = py + v * sin(psi) * latency_in_sec;
- psi = psi - v * delta/Lf * latency_in_sec;
- v = v + acc * latency_in_sec;