**Model**:

I have used a kinematic model which consists of state inputs and control inputs (actuators).

State: x, y, psi and v

State also includes errors : cross track error cte and orientation error

Actuators: delta (steering angle) and a (throttle/acceleration)

Update equations are taken from the lesson.

x​t+1​​=x​t​​+v​t​​∗cos(ψ​t​​)∗dt

y​t+1​​=y​t​​+v​t​​∗sin(ψ​t​​)∗dt

ψ​t+1​​=ψ​t​​+​L​f​​​​v​t​​​​∗δ​t​​∗dt

v​t+1​​=v​t​​+a​t​​∗dt

**Implementation**:

1. Initial state and trajectory points are read from json input.
2. To deal with delay in actuation command execution, a latency of 100ms is used. Update equations are used to predict future state.
3. Then all the trajectory points are transferred/shifted to car’s coordinate system.
4. As the path includes lot of curves, used 3rd order degree polynomial to fit and coefficients are calculated using *polyfit* function.
5. Use the *polyeval* to calculate errors cte and epsi.
6. Use MPC solve to optimize and predict the steering angle and path.

**MPC**

By following walkthough video and lessons, I implement MPC by setting constraints and IPOPT to solve the optimization.Also I have set delta to be in between (-25,25) radians and a to (-1,1).

*Tuning the cost parameters:*

This step is the most time consuming part and I have implemented after lot of trial and error. I initially started with very large values for cte,epsi (~5000) but the car was veering off at the curves a lot.

Then after reading lot of forum posts, below are the observations:

|  |  |  |
| --- | --- | --- |
| CTE | Epsi | observation |
| 300 | 500 | Veering off |
| 50 | 600 | Reduced but not much |
| 30 | 700 | No change |
| 50 | 800 | No of oscillations reduced |
| 30 | 800 | Away from the center at curves |
| 70 | 800 | Cte increased so improved, more aligned to center |

I also tried different cost values to minimize change rate and the gap between sequential actuations.

Above were tuned by setting N=10 and dt = 0.1, and ref\_v = 30mph

I tried to change N=15 and dt=0.05, but the prediction seems to be veering off at the curves and taking more computational time

So I changed to N=7, this reduced number of prediction points and more aligned to the waypoints. But the number of MPC waypoints are less and as a result, car won’t be able to see the road ahead especially in case of curves.

As per review suggestions, for larger dt=0.15 and N=10, car was not able to complete the lap and going out of the lane.

So I decided to keep initial values as N=10 and dt=0.1.

Used update equations to add latency and observed that prediction improved, CTE and number of oscillations reduced, little veering off at the curves and more aligned to the center of the lane.

As per review suggestions updated latency calculation for psi.