The Model

I used model from the class.

$$x_{t+1} = x_t + v_t * cos(\psi_t) * dt$$

$$y_{t+1} = y_t + v_t * sin(\psi_t) * dt$$

$$\psi_{t+1} = \psi_t + \frac{v_t}{L_f} * \delta * dt$$

$$v_{t+1} = v_t + a_t * dt$$

Not sure what to comment on. Equations are implemented in file main.cpp from line 149.

Initially I tried to add velocity for X and ψ calculations, but that always lead to behaviour where car simply drove in circle. Circles are general issue with current implementation. If car increase speed from current 30mph it drives well, but when leaving the bridge controls go completely crazy. Bot sure how to debug, so input would be appreciated.

Non zero values that are passed to MPC

- **Velocity** = velocity + throttle value * latency
- **Cross tr err** = cte + velocity * sin(epsi) * latency
- **Epsi** = epsi + velocity * steering value / Lf * latency

Latency is set to target 0.1 seconds. Velocity is factoring in throttle, since that what it is about. Errors are just incorporating velocity into standard equasions.

Timestep Length and Elapsed Duration (N & dt)

- **N = 12** empirically found sweet spot to match planned trajectory length. If I further increase this number predicted path will exceed planned path and that lead to 180 degrees turn.
- **dt = 0.1;** 0.2 does not work, as car moves too far and predictions got useless, bellow is pointless. Considering that we also have a latency of 0.1 overall fair number, however only for lower speeds.

Polynomial Fitting and MPC Preprocessing

I did not do any distinct preprocessing for MPC. Only transformed into the proper coordinate system with

```
for (int i = 0; i<ptsx.size(); ++i)     {

        double x = ptsx[i] - px;

        double y = ptsy[i] - py;
```

```
        rel_x[i] = x*cos(psi) + y*sin(psi);

        rel_y[i] = y*cos(psi) - x*sin(psi);

}
```

It is code from exercises, but I made some optimizations to cut down number of operations in the loop.

Model Predictive Control with Latency

Latency was implemented into the functions that were discussed above. In short, latency is used as a discount factor to all values that passed to MPC.

The vehicle must successfully drive a lap around the track.

2 laps.

https://youtu.be/kEoPQ3B9dLo