MPC

Compiling

Once the install for uWebSocketIO is complete, the main program can be built and run by doing the following from the project top directory.

1. ./build.sh - build project
2. ./clean.sh - clean build folder
3. ./run.sh - connect pid to Term2 simulator

The Model

We used the Kinematic model without forces between the wheels and the road. It follows below:

```
x_[t] = x[t-1] + v[t-1] * cos(psi[t-1]) * dt
y_[t] = y[t-1] + v[t-1] * sin(psi[t-1]) * dt
psi_[t] = psi[t-1] + v[t-1] / Lf * delta[t-1] * dt
v_[t] = v[t-1] + a[t-1] * dt
cte[t] = f(x[t-1]) - y[t-1] + v[t-1] * sin(epsi[t-1]) * dt
epsi[t] = psi[t] - psides[t-1] + v[t-1] * delta[t-1] / Lf * dt
```

It includes x and y coordinates, orientation angle (psi), velocity (v), as well as the cross-track error (cte) and psi error (epsi). Actuator outputs are acceleration (a) and delta (steering angle).

Timestamp Length and Elapsed Duration

I chose 10 and 0.1 values for N and dt, as was recommended by course materials. I tried other values, such as 20 / 0.5, 8 / 0.125, 5 / 0.2, but they produced worse results.

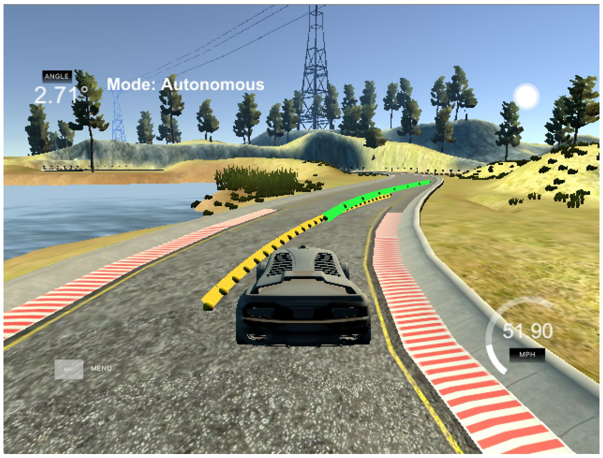
Polynomial fitting and Preprocessing

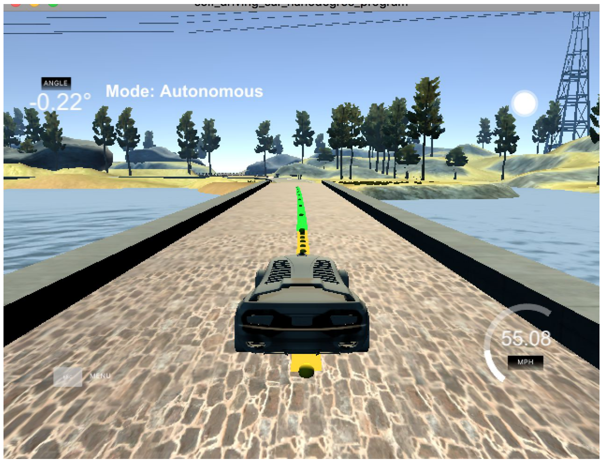
I preprocessed waypoints by transforming them to car’s perspective in the ./src/main.cpp 105 - 113 lines. Then I fitted a third-degree polynomial to the transformed coordinates.

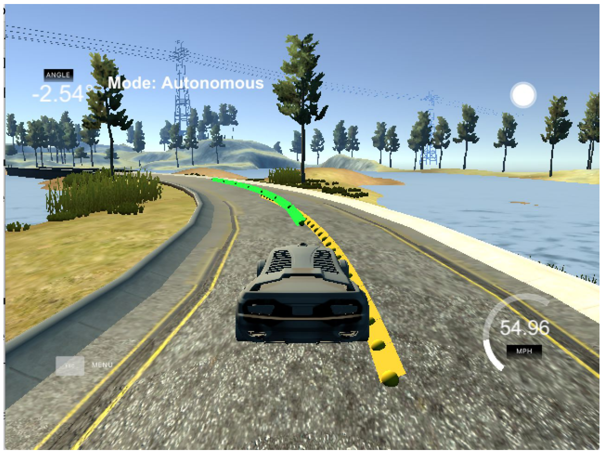
Model Predictive Control with Latency

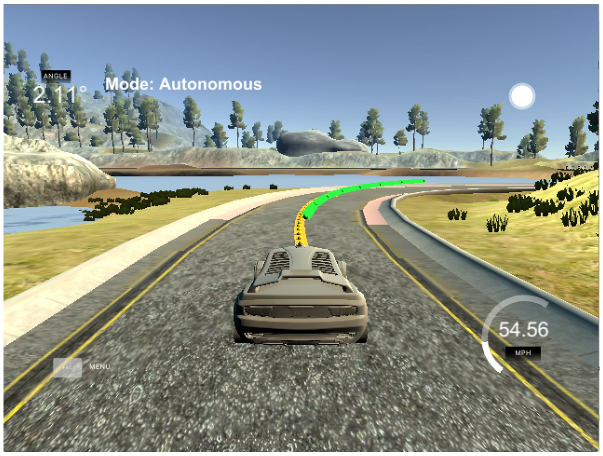
I used a Kinematic model to deal with the delay (114 - 134 lines in ./src/main.cpp)

Results


ANGLE
2.71°
Mode: Autonomous
51.90
MPH
MENU


ANGLE
-0.22°
Mode: Autonomous
55.08
MPH
MENU


ANGLE
-2.54°
Mode: Autonomous
54.96
MPH
MENU


ANGLE
2.11°
Mode: Autonomous
54.56
MPH
MENU