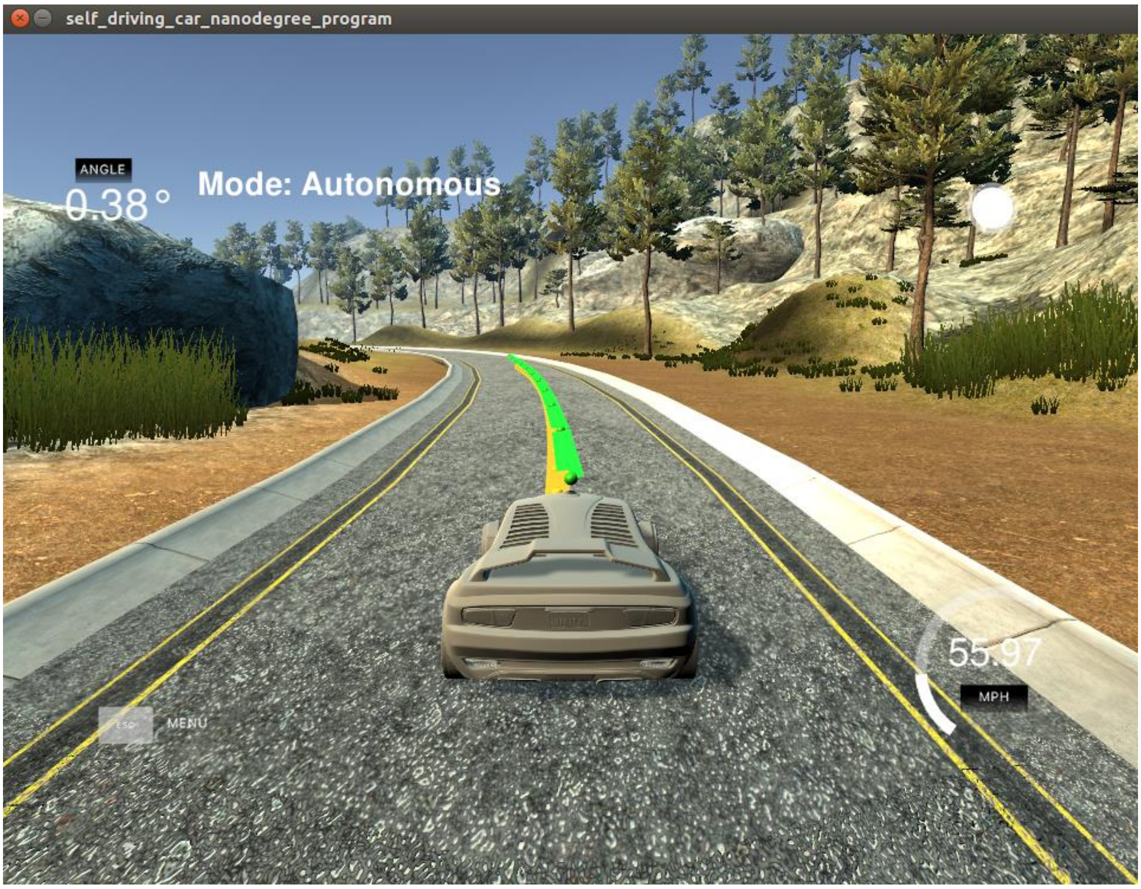
Model Predictive Controller Project – Udacity Self-Driving Car Nanodegree.

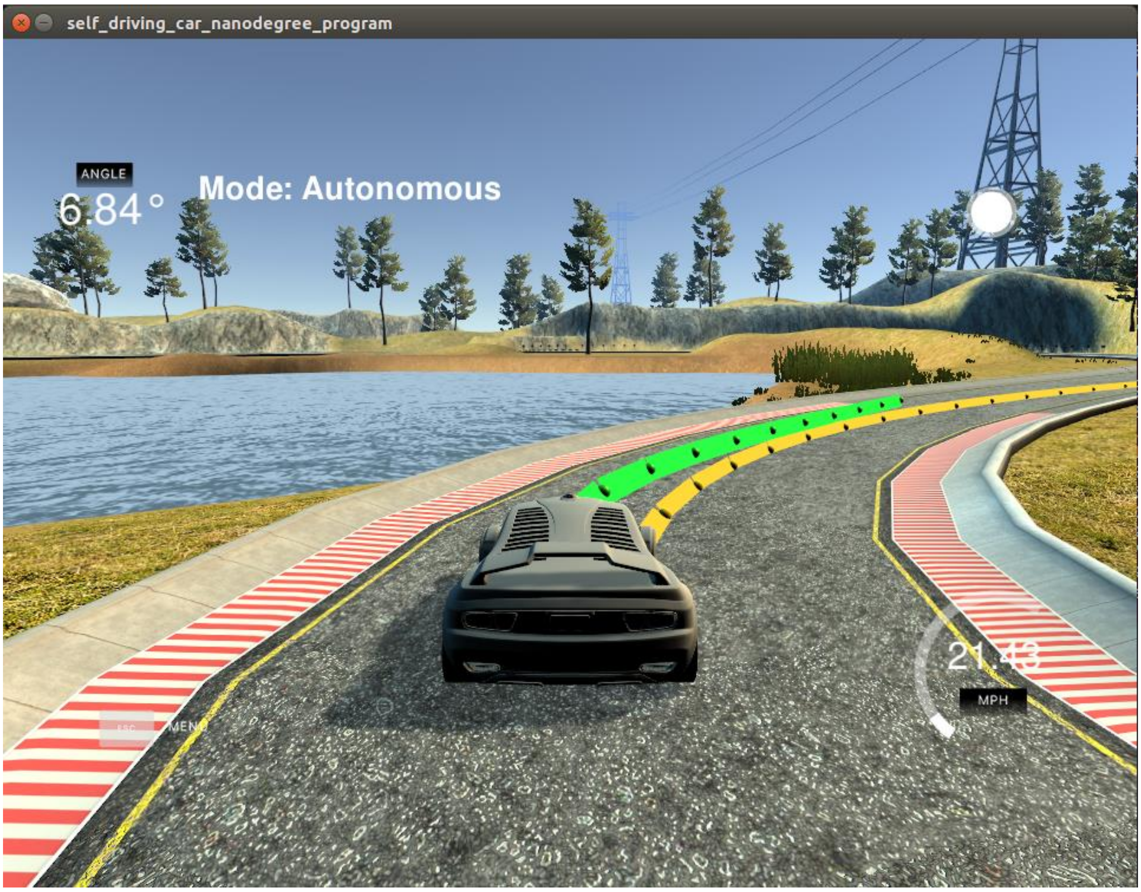
By Alejandro Trigo



1. The model

I think the most important part of the model is the cost function; here is where all the magic happens. By increasing the cost when the car wants to speed up and steer in the same time, I can fight the delay of 100ms. The other constants I set for the cost function where selected by trial and error.

The initial state is set to 0 for the x, y and psi values, the speed is got in the JSON data. The cte is calculated by the coeffs (polynomial vector of 3er degree of the points got in the JSON message) and the epsi is the atan of the coeffs negated.

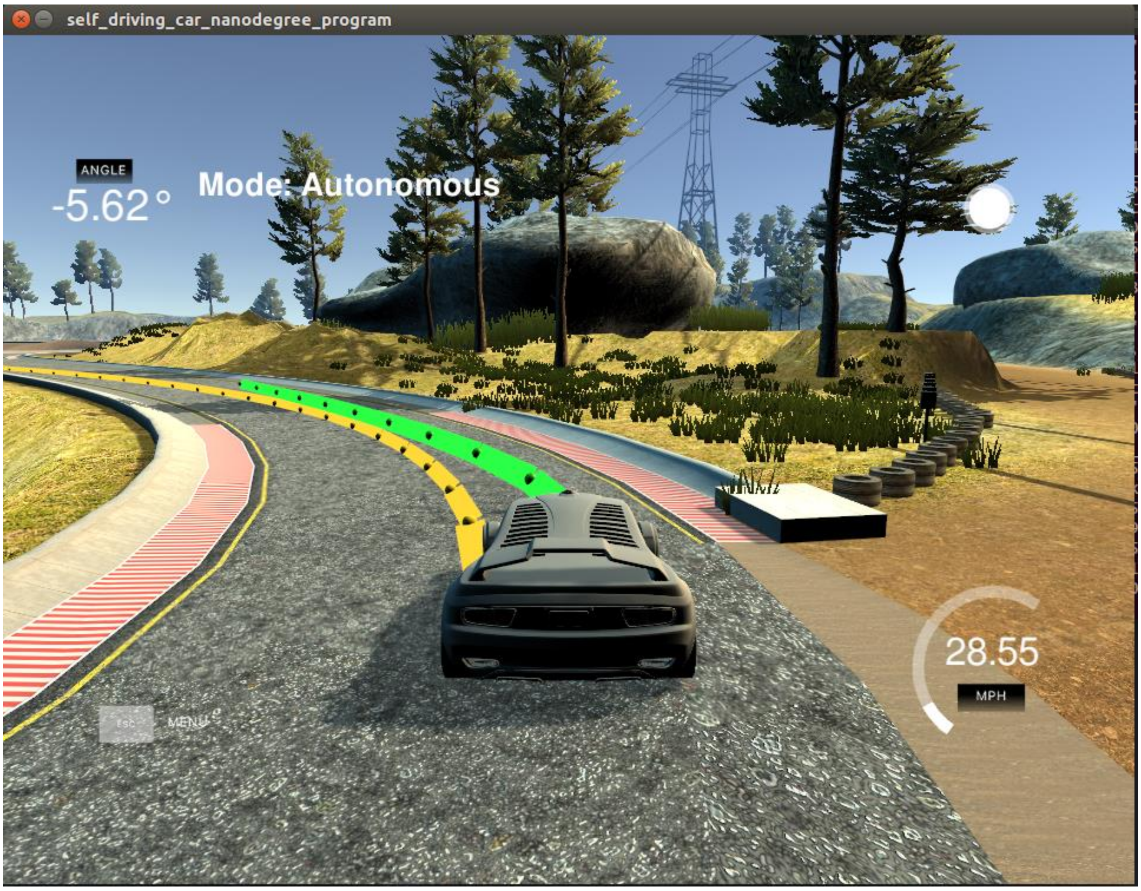


2. Timestep Length and Elapsed Duration (N & dt)

In the last moment, I found that the perfect N for my model was 12 and dt 0.12, first I tried with 10 and 0.1 and it was working, but sometimes it was going over the ledges with one wheel. Because of that, I thought that with a bit more steps it could fight a little bit more with the delay.

3. Conclusion

I have to say that I don't think my model is the best, because I don't really fight with the delay as good as I want, I just tell the car to slow down in the curves. I will try to improve the model in the future to get a nice solution.


self_driving_car_nanodegree_program
ANGLE
-5.62°
Mode: Autonomous
28.55
MPH
Esc
MENU