PID Controller Project – Udacity Self-Driving Car Engineer Nanodegree

By Alejandro Trigo Rosón

Figure 1: The car in the center of the lane during a curve

1. Describe the effect each of the P, I, D components had in your implementation.

P: The P component is making the car steer proportional to the error in the distance to the middle of the road. In this case, I measure this error with the cte variable by the Kp coefficient. (p_error = cte * Kp). With this information, we are able to keep the car close to the middle line.

D: To avoid overshot, we use another constant Kd to reduce the error when we are close to the middle. This is done by subtracting the error we have in the P component to the previous cte and then multiplying by the constant Kd. (d_error = (previous_cte – p_error) * Kd).

I: The I is a safety component since the objective is to deal with the systematic bias the direction can have. If we assume that the car has no deviation in the direction we have to set the constant Ki to 0. Because of this parameter is important to make a function that looks for the best value of the parameters, like twiddle.

Figure 2: The critical part, here it does the curve quite positioned to the right

2. Describe how the final hyperparameters were chosen.

All the hyperparameters were chosen by trial / error, since I am a bit busy this month (I have to say thank you to Udacity because I am working in a Self-Driving Car Company) I had no time to finish the twiddle function.

The Kp coefficient: I saw that the best numbers here are close to -0.1

The Ki coefficient: Since there are no deviation in the direction this parameter is set to 0.

The Kd coefficient: I reached that the perfect values are between -0.75 and -1.

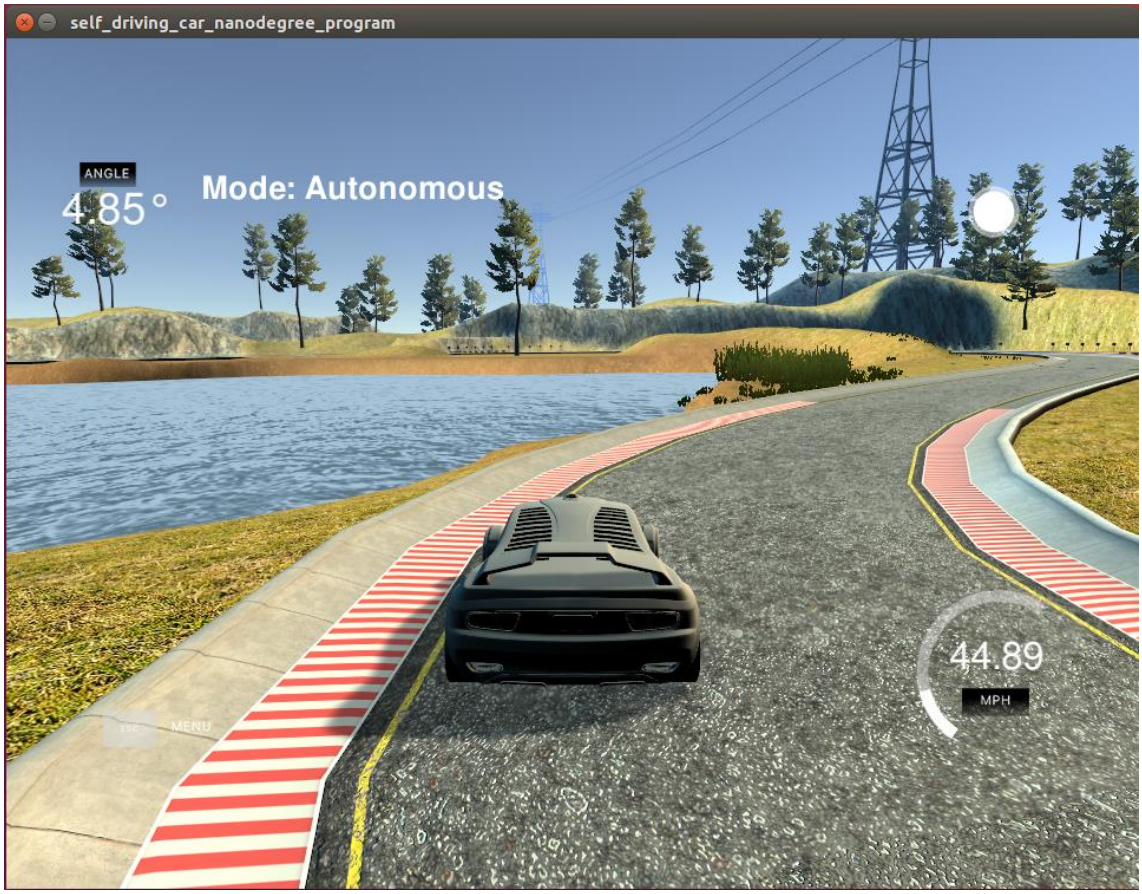

Figure 3: Other critical point, the right curve is done in the right of the lane.