Autonomous Line Following Racecar
Homework 2: Car Steering

EE192 Spring 2018, Professor Ron Fearing

Trey Fortmuller, Joey Kreoger, Charlene Shong (Team 6)

March 23, 2018

1 Steering Simulation - Proportional Control

With pure position control, $\delta = k_p y_a$, we chose a fixed speed $V = 1.25$ m/s and a $k_p = 500.0$ which allowed the car to successfully complete the track without hitting any of the cones. The worst-case overshoot was 0.3049 m or 3.049 cm, and is indicated as a point in Fig. 1 and a red-highlighted path in Fig. 2. A comparison between the lateral error and the actual position of the car is shown in Fig. 3.

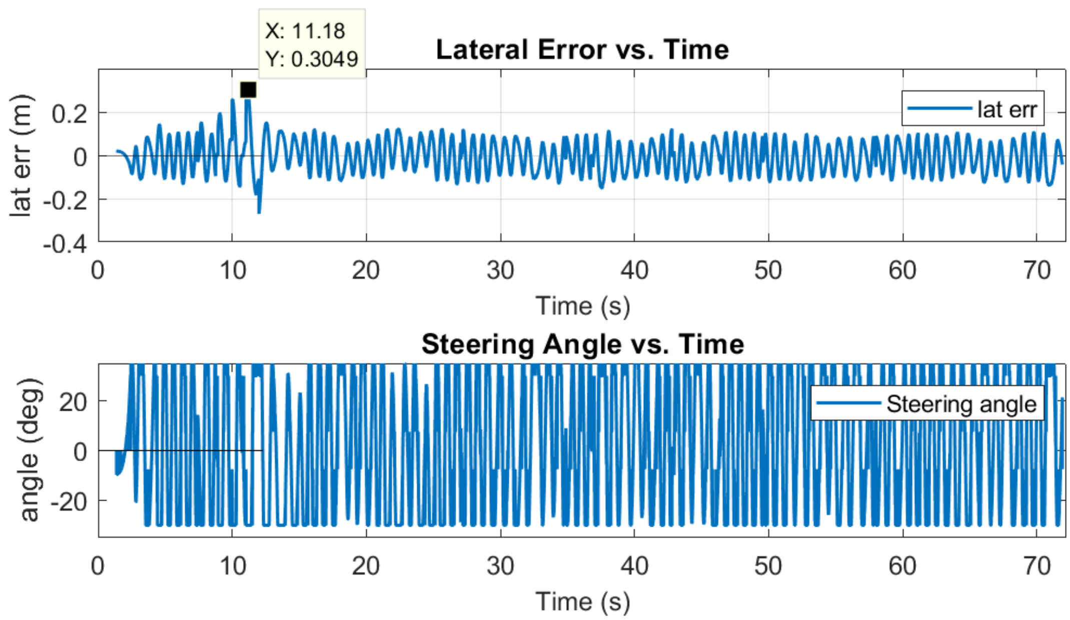


Figure 1: Time Plots of the Lateral Error and Steering Angle

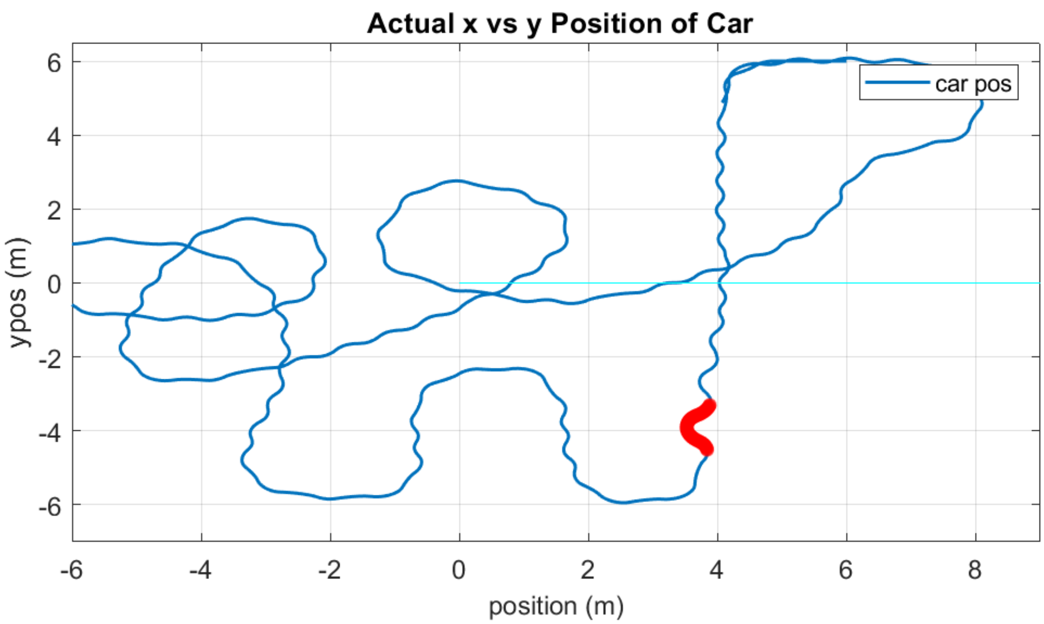


Figure 2: Actual x vs. y Position of Car

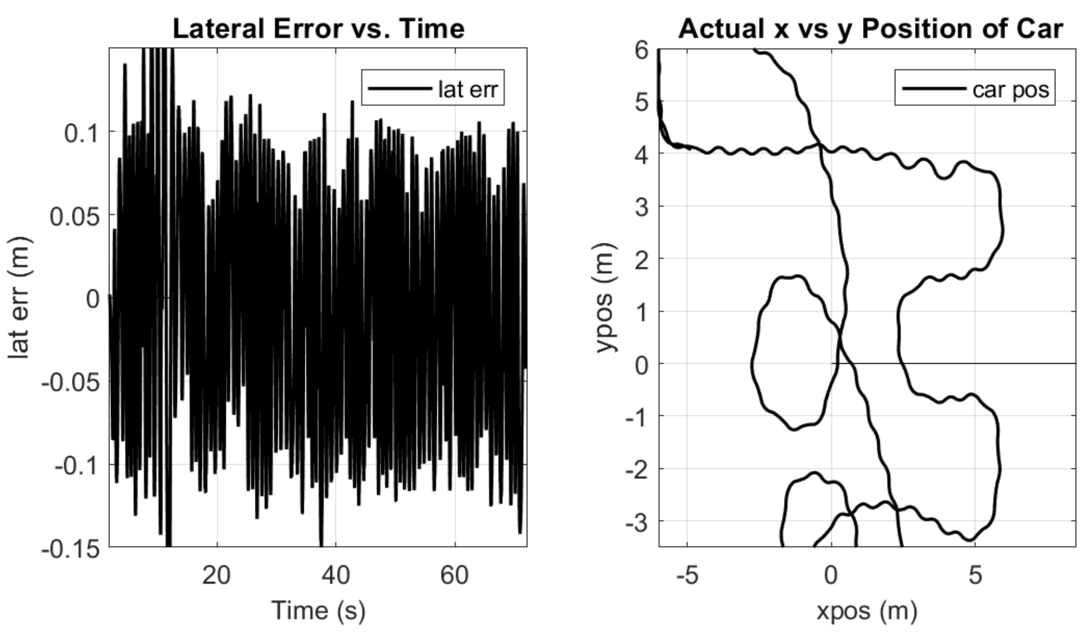


Figure 3: Lateral Error and Actual Position Plots for Comparison

2 Steering Simulation - Proportional & Derivative Control

With PD control, $\delta = k_p y_a + k_d \dot{y_a}$, we chose a fixed speed $V = 2.2$ m/s (almost double our earlier speed), a $k_p = 500.0$, and a $k_d = 50.0$ which allowed the car to successfully complete the track without hitting any of the cones. The worst-case overshoot was 0.1766 m or 1.766 cm, and is indicated as a point in Fig. 4 and a red-highlighted path in Fig. 5. A comparison between the lateral error and the actual position of the car is shown in Fig. 6. PD control was much better at following the line, "swayed" significantly less (i.e. the derivative control reduced the overshoot), and was able to handle higher speeds as a result.

We also tried PD control with a fixed speed $V = 3.75$ m/s (more than triple our first speed), a $k_p = 500.0$, , a $k_p = 500.0$, and a $k_d = 200.0$ which allowed the car to successfully complete the track without hitting any of the cones (though it came close). The worst-case overshoot was 0.3855 m or 3.855 cm, and is indicated as a point in Fig. 7 and a red-highlighted path in Fig. 8. A comparison between the lateral error and the actual position of the car is shown in Fig. 9. In this case, the overshoot is similar to our first run with only position control, but at a much faster speed.

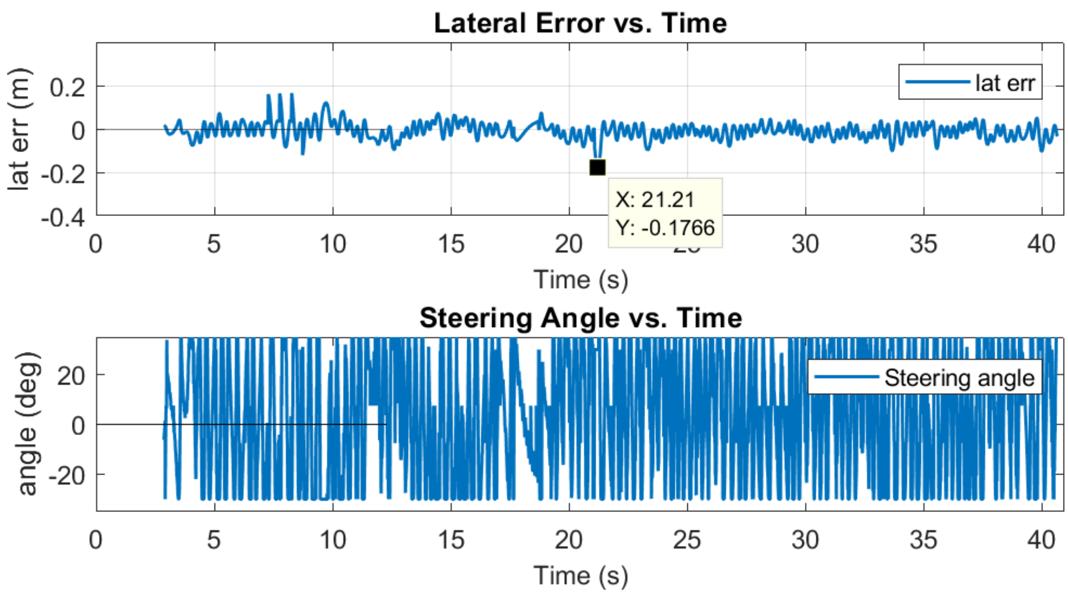


Figure 4: Time Plots of the Lateral Error and Steering Angle

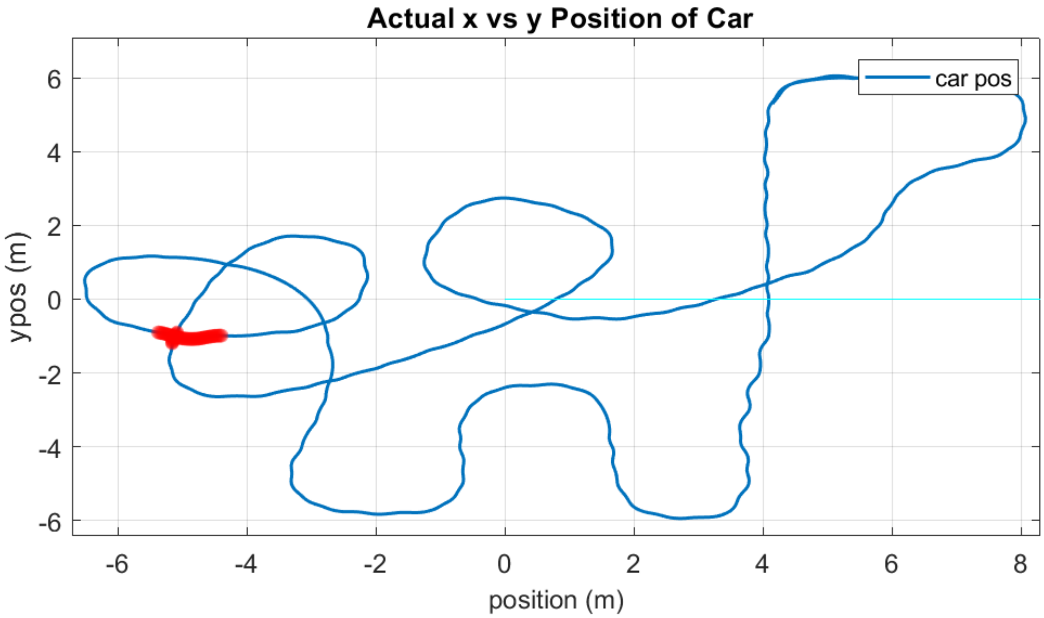


Figure 5: Actual x vs. y Position of Car

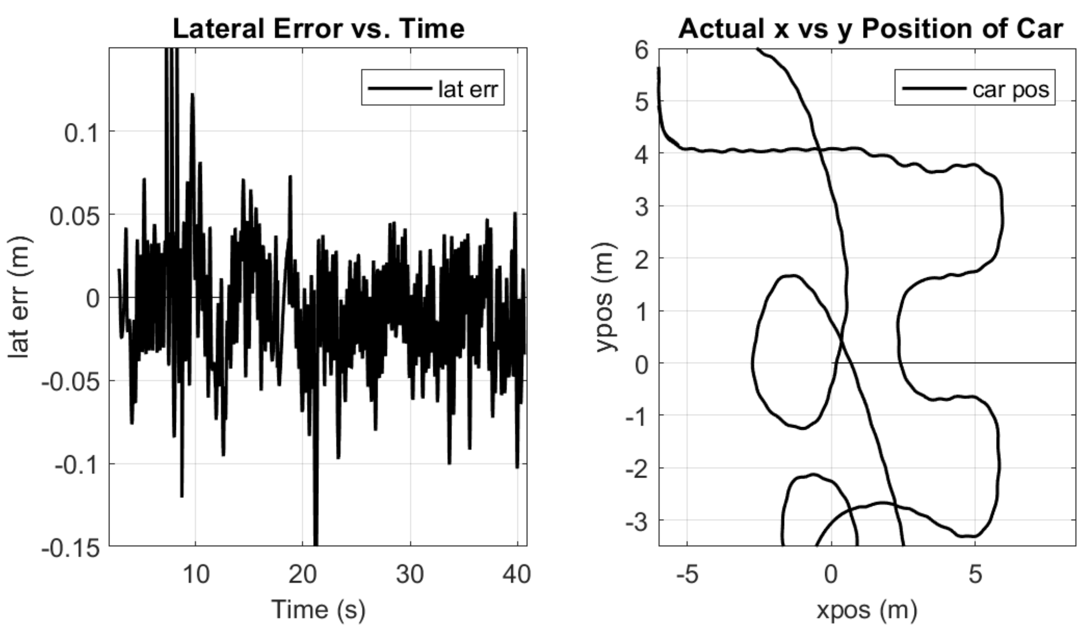


Figure 6: Lateral Error and Actual Position Plots for Comparison

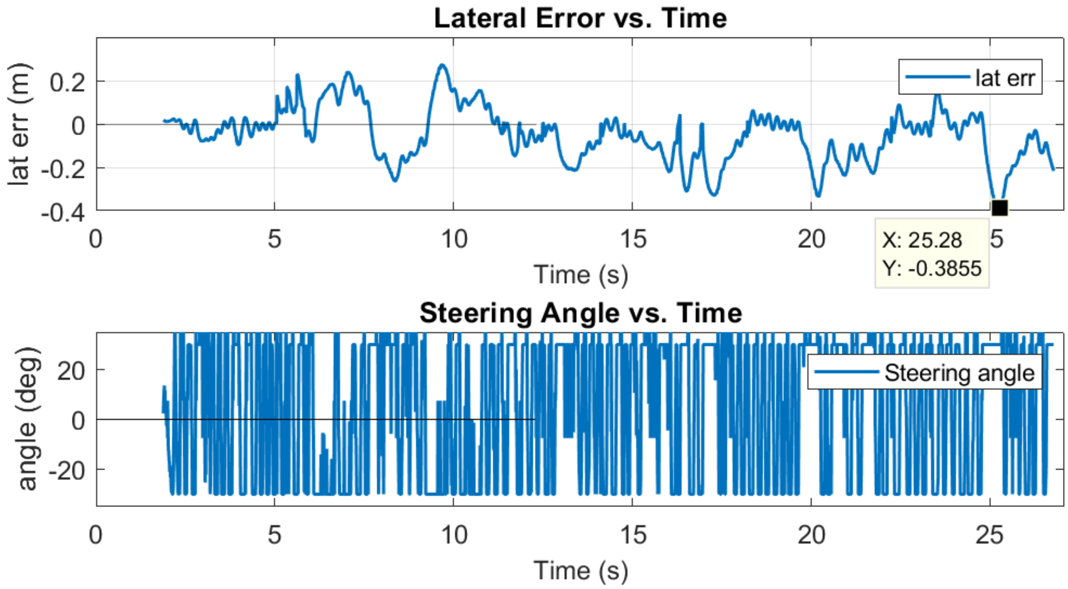


Figure 7: Time Plots of the Lateral Error and Steering Angle

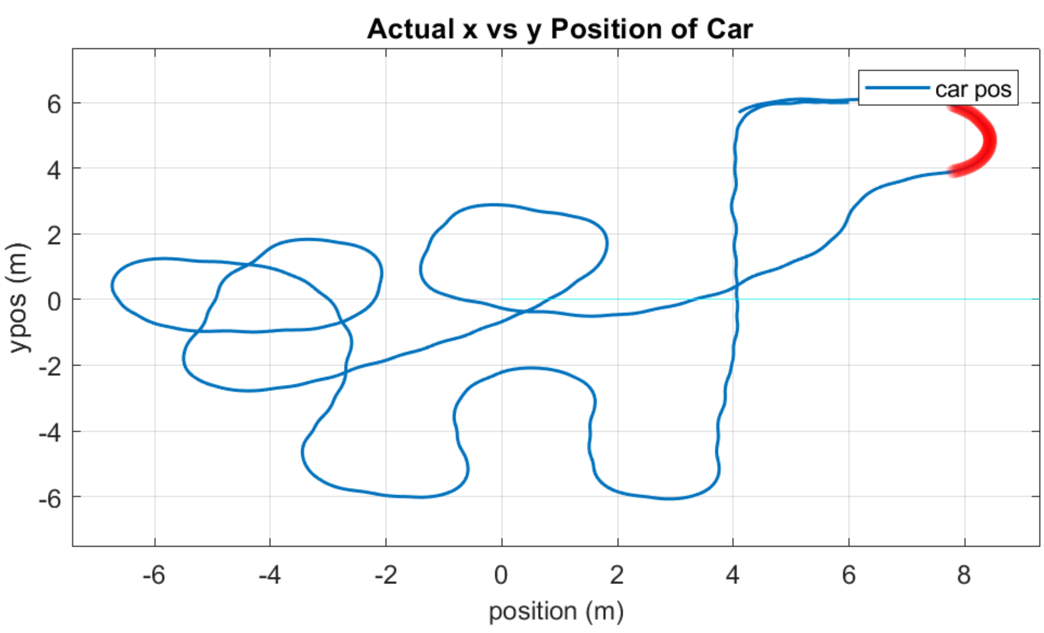


Figure 8: Actual x vs. y Position of Car

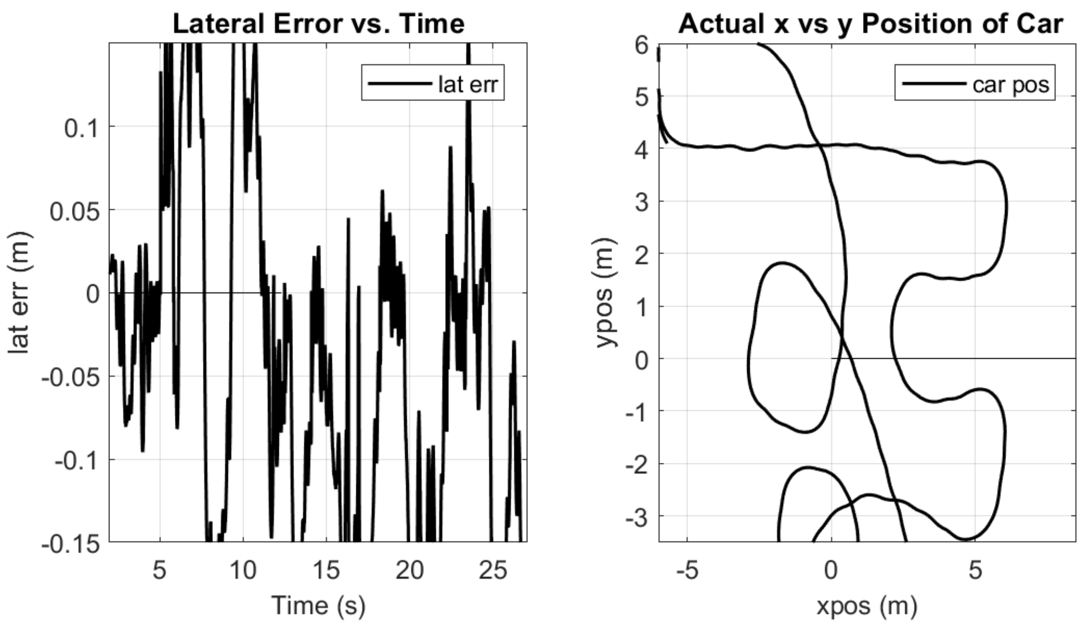


Figure 9: Lateral Error and Actual Position Plots for Comparison

3 Steering Servo Speed Limit

With PD control and a more reasonable steering servo, we chose a fixed speed $V = 3.15$ m/s (almost double our earlier speed), a $k_p = 500.0$, and a $k_d = 150.0$ which allowed the car to successfully complete the track without hitting any of the cones. The worst-case overshoot was 0.1766 m or 1.766 cm, and is indicated as a point in Fig. 10 and a red-highlighted path in Fig. 11. A comparison between the lateral error and the actual position of the car is shown in Fig. 12. In this case, the steering servo itself responded much slower to commands to turn than in previous runs, which caused our car to "sway" more and required a higher k_d in order to compensate.

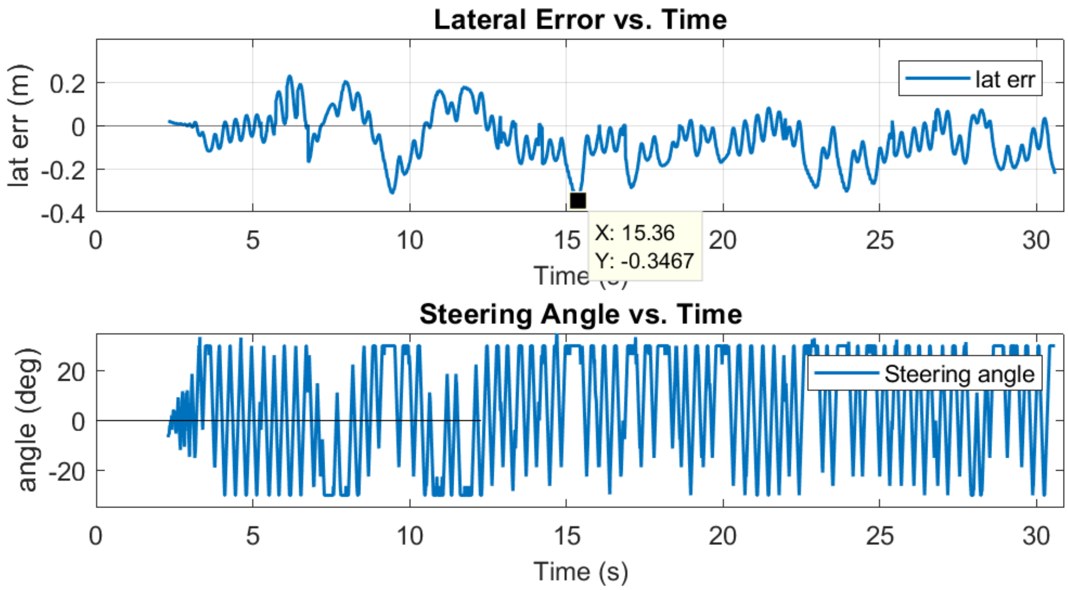


Figure 10: Time Plots of the Lateral Error and Steering Angle

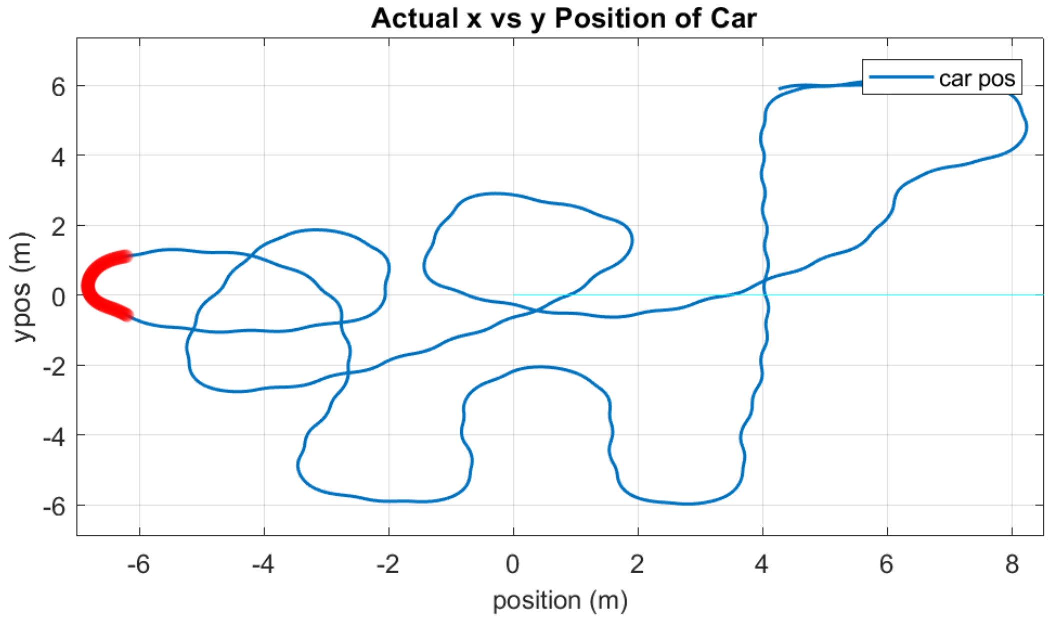
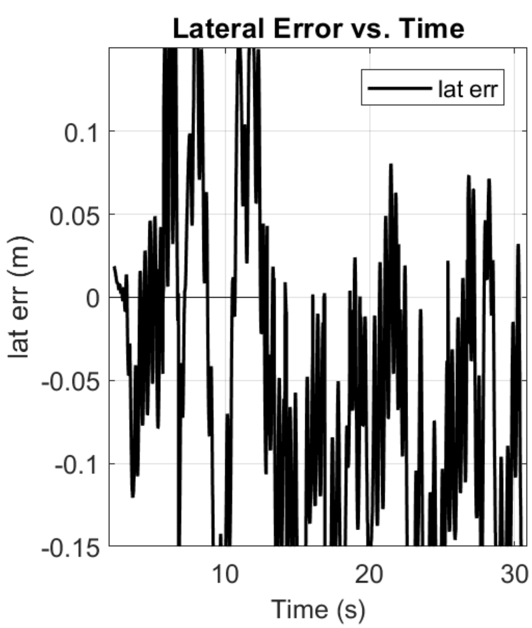


Figure 11: Actual x vs. y Position of Car



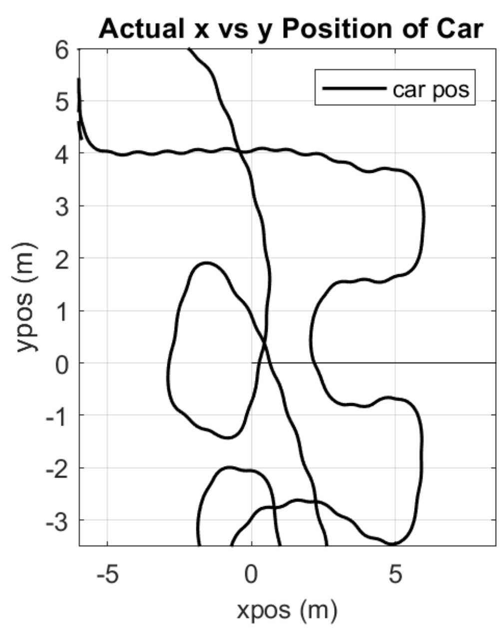


Figure 12: Lateral Error and Actual Position Plots for Comparison

4 PC Control, Steering Servo Limit, & Speed Control

With PD control, a more reasonable steering servo and a simple if statement shown below for speed control (where, if the lateral error was greater than 0.15 the car would slow down to $V = 2.5$, otherwise it would maintain a decent speed of $V = 3.0$), a $k_p = 500.0$, and a $k_d = 100.0$, the car to successfully complete the track without hitting any of the cones. The worst-case overshoot was 0.31 m or 3.1 cm, and is indicated as a point in Fig. 13 and a red-highlighted path in Fig. 14. A comparison between the lateral error and the actual position of the car is shown in Fig. 15.

It's not as clean as we'd like it to be (the easiest way to achieve that would be to tune a bit more carefully, or just slow down the car), but it did avoid the cones. The velocity controller helped a bit with the swaying and some of the larger turns, but the car still struggles to make tight turns without a lot of overshoot. The laptime was 31.42 seconds, which was pretty similar to when we had just a fixed car speed. Most likely, we need to implement a more advanced controller (probably PID or just PD control) for speed as well.

```
# Constant speed for now. You can tune this and/or implement advanced
# controllers.
#car.set_speed(ve)
if (np.absolute(lat_err) > 0.15):
  car.set_speed(2.5) #slow down
else:
  car.set_speed(3.0) #setpoint
```

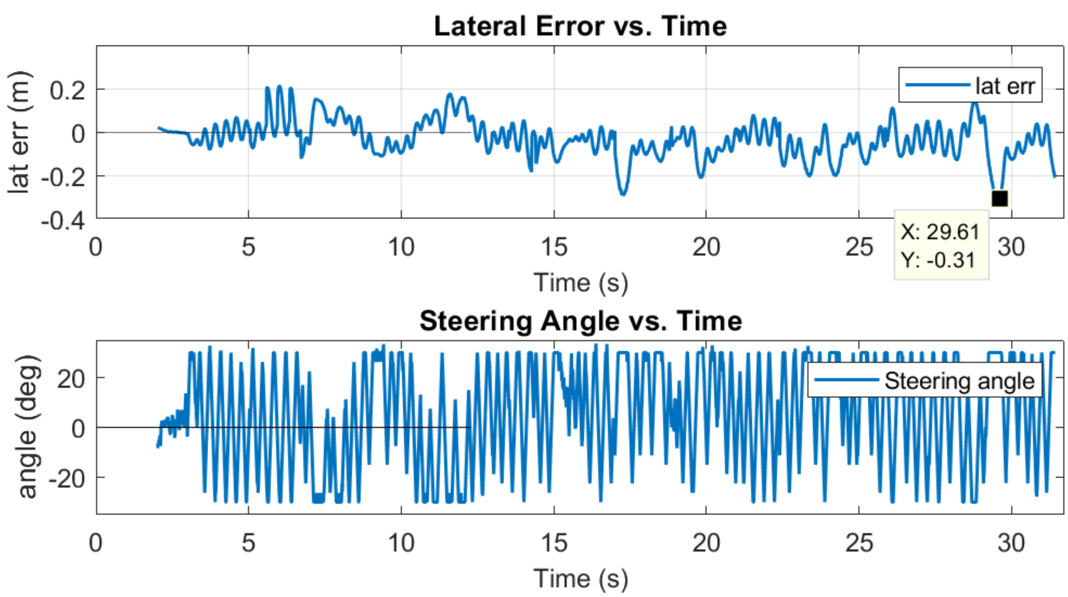


Figure 13: Time Plots of the Lateral Error and Steering Angle

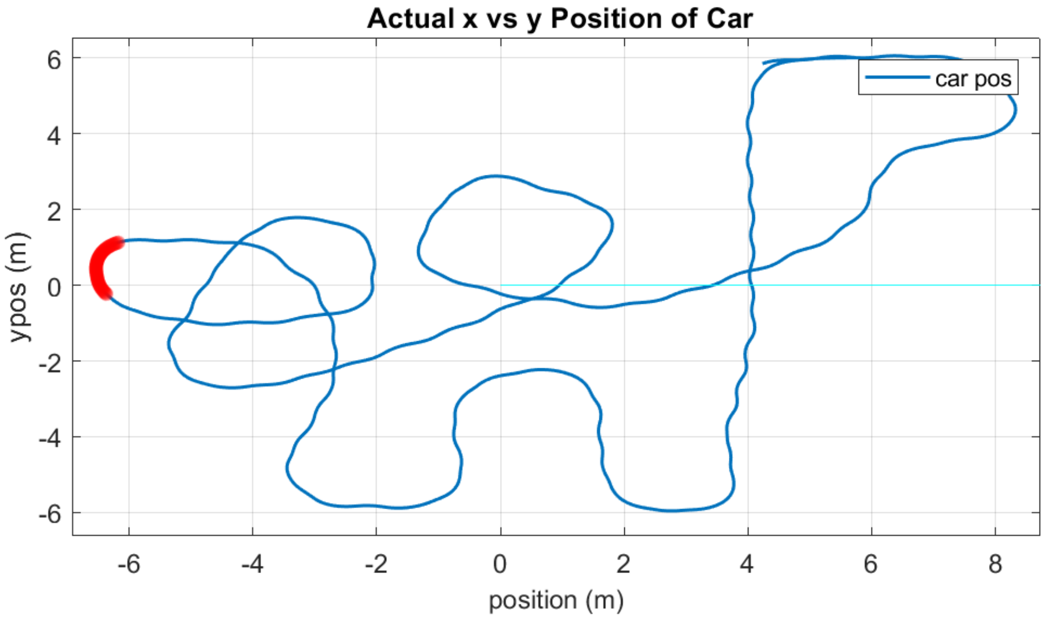


Figure 14: Actual x vs. y Position of Car

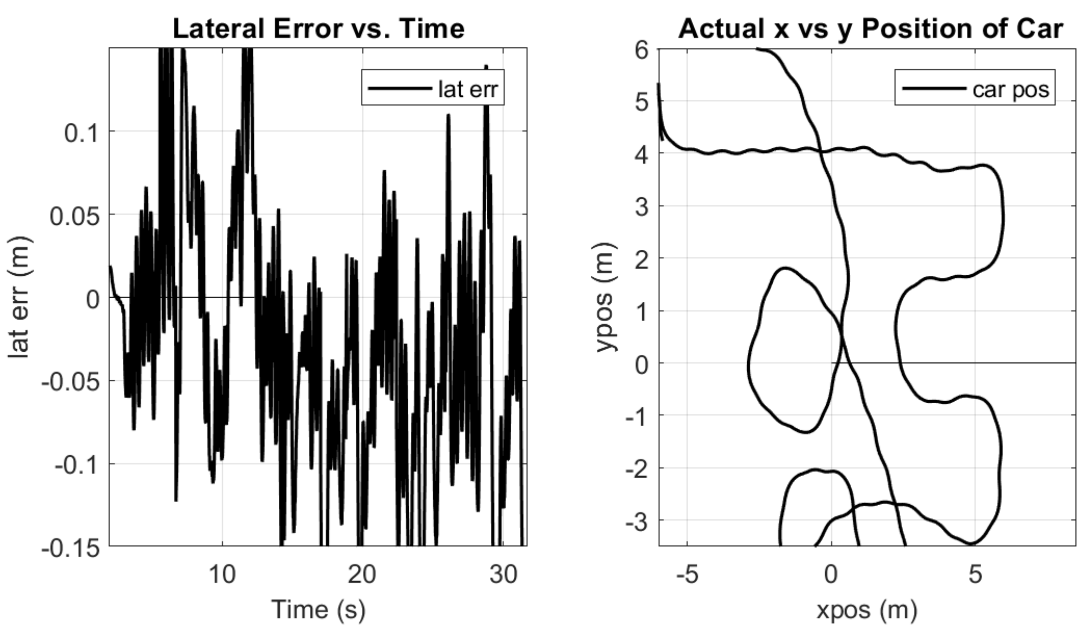


Figure 15: Lateral Error and Actual Position Plots for Comparison