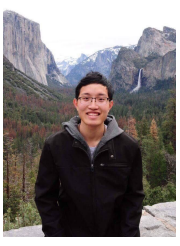
Autonomous Car Path Tracking with Feedback Control

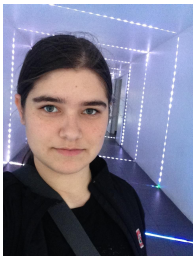
ME227 Vehicle Dynamics and Control
Spring Quarter 2018
Team Pole Position

Jackie Chan
Biochemistry
Chem ‘18 | MSME ‘20

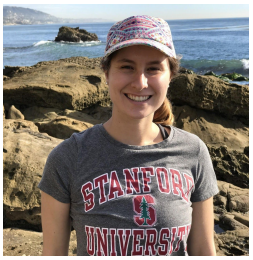


Ryan Lee
Mechanical Engineering
ME ‘18 | MSME ‘19

Tatiana Mowry
Mechanical Engineering
ME ‘17 | MSME ‘19



Megan Rowe
Mechanical Engineering
BSME ‘19



1 Overview of Control Scheme

1A Desired Longitudinal Speed and Acceleration Profile as a Function of Distance

To ensure an autonomous Audi TT-S could follow a given path within 30 cm of lateral error and below acceleration thresholds, we defined our vehicle's desired velocity and acceleration profiles as a function of distance along the path. To move the car around the track as quickly as possible, we maximized accelerations around the three unique track geometries of the oval path: clothoids, straights and arcs. Figure 1 describes the designated path in terms of track geometries. As a general strategy, our group first defined the desired accelerations needed to maximize car speed along the track and updated velocities based on these accelerations.

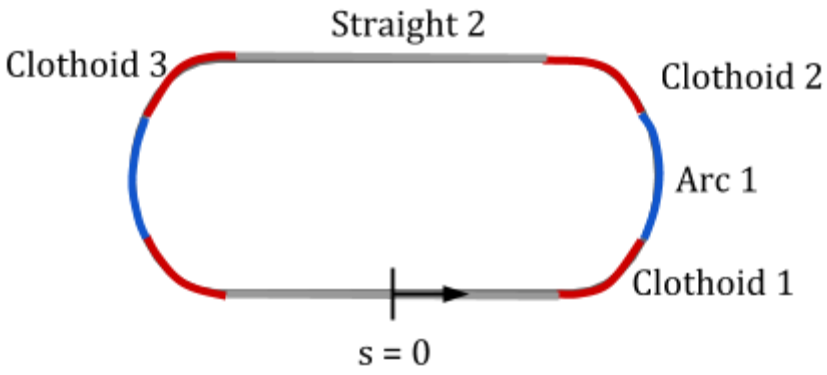


Figure 1: *Designated Path and Unique Path Geometries*

We found that the longitudinal velocity of the car along all points of the arc to be a single critical velocity where all acceleration comes from lateral acceleration. Given the equation of a point mass lateral acceleration, or $a_y = V^2$ /R where ay is equal to our maximum allowable lateral acceleration, our velocities at clothoids were calculated to be $V_{critical} = \sqrt{a_{max} * R}$ where R is the radius of the curve. The arc's critical longitudinal speed for this path was approximately 5.9 m/sec, and since this speed did not change, the arc's acceleration profile was set to zero for all distances along the arcs.

On clothoids, equations for speed along a clothoid were calculated by approximating the car as a point mass moving along the path, where lateral or centripetal acceleration around a radius of length R is given by: $a_y = V^2$ /R or $a_y = K_{curvature} * Ux^2$.

$$dUx/ds = 1/Ux * \sqrt{a_{max}^2 - a_y^2} = 1/Ux * \sqrt{a_{max}^2 - (K_{curvature} * Ux^2)^2}$$

Equation 1: *Change in Speed Profile Along a Clothoid*

By multiplying dUx/ds by Ux where Ux = ds/dt, longitudinal acceleration at a given distance along the path was obtained. It is important to note that this calculation assumes

that the vehicle longitudinal speed V_x is identical to the speed along the path, i.e. that the vehicle stays on the path. Given that we are utilizing a lateral controller, this is reasonable to assume, given we have minimized lateral error with a given tolerance. For clothoids, speeds were determined by Euler integrating from known velocities, where the next distance step (ds) velocity was given by:

$$Ux(s + ds) = Ux(s) + dUx/dx * ds$$

Equation 2: *Clothoid Velocities*

When entering clothoids from straights, such as clothoid 1 in Figure 1, the sign of dUx/ds is negative, as the vehicle decelerates from high speed on a straight to a slower critical velocity (5.9 m/s) around the arc. Velocities for entering a clothoid from a straight were back-calculated using the arc speed as the final velocity of the clothoid then solving for the entrance velocity into the clothoid. When exiting clothoids onto straights, such as clothoid 2, the sign of dUx/ds is positive to denote acceleration from the slow critical speed of 5.9 m/s to a higher speed on the clothoid. Since speeds were known at the arcs, velocities for exiting clothoids onto straights were forward calculated using the arc speed as the initial velocity of the clothoid.

In general, the kinematic equation for final velocity given an initial velocity and acceleration used to obtain the velocities on the straights is given by equation 3 and 4.

$$U_{x,final} = \sqrt{U_{x,initial}^2 + 2 * a_{x,initial} * ds}$$

Equation 3: Kinematic Equation for Velocity

$$U(s + 1) = \sqrt{U(s)^2 + 2 * a_x(s) * ds}$$

Equation 4: Kinematic Equation for Velocity for Euler Integration

Equation 4 describes the kinematic equations in a single step for forward integration for longitudinal velocities, with a given step size, ds, of 0.25 meters and a longitudinal acceleration as defined by the specific path geometry.

On straights (not including the starting and finishing one), we selected our velocity profiles (calculated using equation 4) based on constant acceleration out of a clothoid for half the distance of the straight followed by constant deceleration at the same magnitude until the next clothoid was entered. For example, when the car would exit clothoid 2, the calculated (positive) exit acceleration at that point would be maintained until the car passed the midpoint of straight 2. By design, this acceleration profile results in a desired velocity profile in which after the straight, the car returns to the initial velocity exiting the clothoid. Given that the clothoid geometry is symmetric, calculated entry and exit velocities

of the clothoids on the track are identical, and our chosen speed profile allows the car to accelerate from a starting velocity and then return to that same velocity.

Regarding the initial straight, we constructed our velocity profile starting from rest and accelerating at a constant rate up to the first clothoid, which was implemented using equation 4. For the final straight, we calculated a braking acceleration using equation 3 such that the final velocity was -0.1 m/s (to resist idling forward) and that the vehicle would stop three meters before the path ended. Using the exit velocity of the final clothoid as the starting point, the velocity profile for the last stretch was built using equation 4 with this braking acceleration.

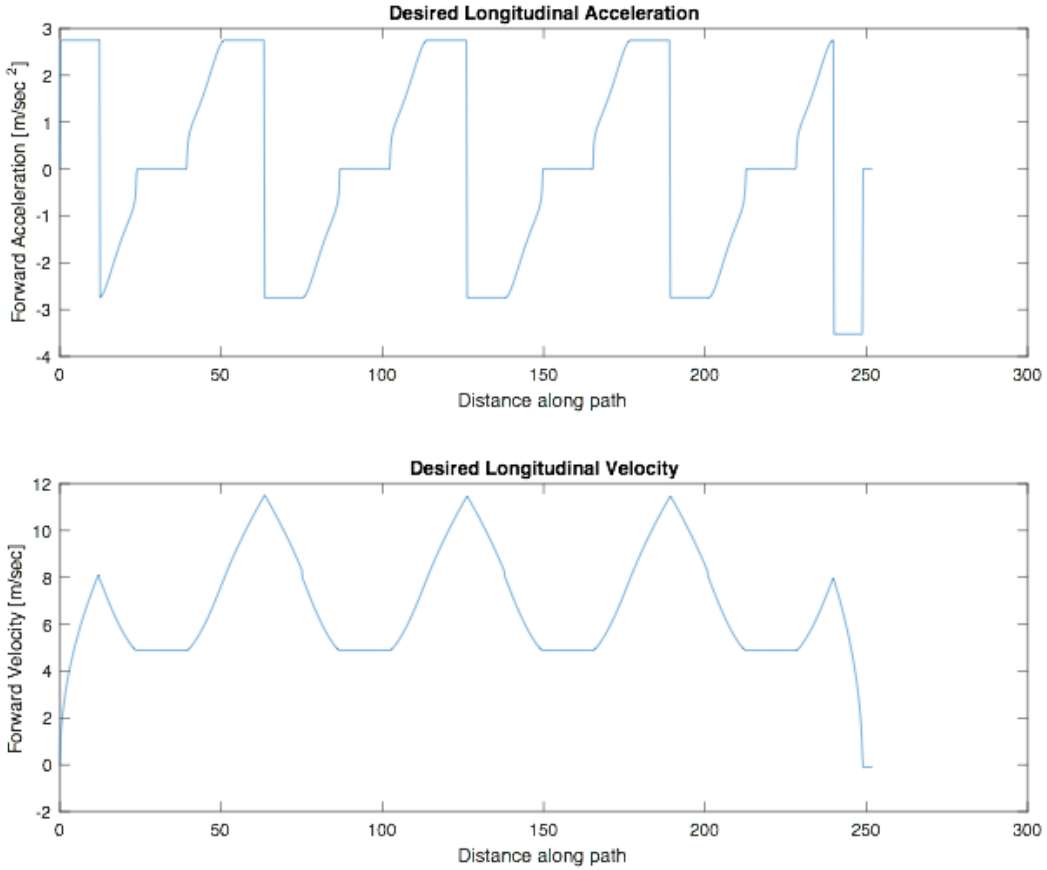


Figure 2: *Desired Acceleration & Velocity Profiles Relative to Path*

1B Longitudinal & Steering Controllers

Our longitudinal speed controller consisted of a simple feedforward and feedback structure designed to drive the the error between the desired and actual longitudinal speed to zero. Our feedforward term accounted for car and road information that we were given such as the drag coefficient and rolling resistance. We also included our acceleration profile since this is predetermined information that the car could use to track longitudinal speed. In this controller, the longitudinal drive/brake force was given by

$$F_x = k_{drive} * (U_{x,des} - U_x) + F_{drag} + F_{rolling\ resistance} + F_{grade} + m * a_{x,\ des}$$

We determined our gain k_{drive} by choosing an acceleration of 0.15g's for a 1m/s velocity error as shown below:

$$k_{drive} = \frac{F_{x,des}}{U_{x,diff}} = \frac{0.15mg}{1m/s}$$

Overall, we had very good tracking performance in both simulation and experiment with this choice of controller and gains as shown below in Figure 3.

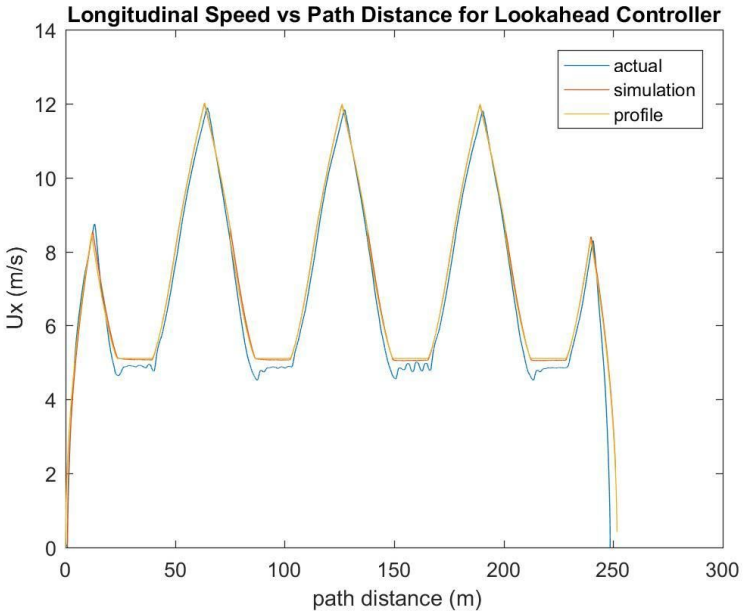


Figure 3: *Longitudinal speed for our desired velocity profile , simulation, and experimental results.*

Our first steering controller was a lookahead controller with a feed-forward term. In this first controller, steer angle was given by

$$\delta = \frac{-K_{la}(e + x_{la} * \Delta\psi)}{C_{\alpha f}} + \delta_{ff}$$

$$\delta_{ff} = \frac{K_{la} * x_{la} * \Delta\psi_{ss}}{C_{\alpha f}} + \kappa * (L + U_x{}^2 * k)$$

$$\Delta\psi_{ss} = \kappa * \left(\frac{m * a * U_x^2}{L * C_{\alpha r}} - b \right)$$

We used a lookahead gain K_{la} = 7000 and lookahead distance x_{la} = 25m. The lookahead gain was determined by simplifying the steer angle equation by setting x_{la} = 0m and δ_{ff} = 0. From here, we chose an initial gain that gave a reasonable steer angle given a lateral error e

= 1m. We refined this gain by running our lookahead controller in simulation and adjusted the gain to give us the minimum lateral error given a lookahead distance x_{la} = 25m.

Our second steering controller utilized PID control. The proportional control used a lookahead controller with a feedforward term, similar to our first controller. The integral term saved a persistent variable that added up the error for each time step. This sum would get reset along the straight paths to prevent integral windup since there would be significant error buildup coming out of curves. The derivative term was a gain multiplied by the time derivative of the error. The error derivative is given in equation 5. In this second controller, steer angle was given by

$$\delta = \frac{-K_{la}(e + x_{la} * \Delta\psi)}{C_{\alpha f}} + \delta_{ff} - K_d * \frac{de}{dt} - K_i \int e dt$$

Since we used the lookahead controller in our PID, we are double counting the derivative term. To reduce the effect of the derivative term in the lookahead terms, we reduced the lookahead distance x_{la} to be 5m instead of the original 25m. We kept the original proportional gain of 7000 and used our simulation to iterate on gains for the derivative and integral terms to minimize our lateral error along the path.

2 Lookahead Controller Performance in Simulation & Experiment

The lateral error we observed in experiment was a bit larger than the value predicted by our simulation as seen in Figure 4 below. Our experimental results produced a peak lateral error hovering between 25cm to just over 30cm, while our simulation predicted a peak lateral error of around 16cm. In areas of peak lateral error, our simulation predictions were off by almost a factor of 2, which is not ideal. However, we still stayed very close to our desired error spec. Our greatest deviation occurred in our first turn, where we barely went over our desired error spec of 30 cm, maxing out at around 30.9cm.

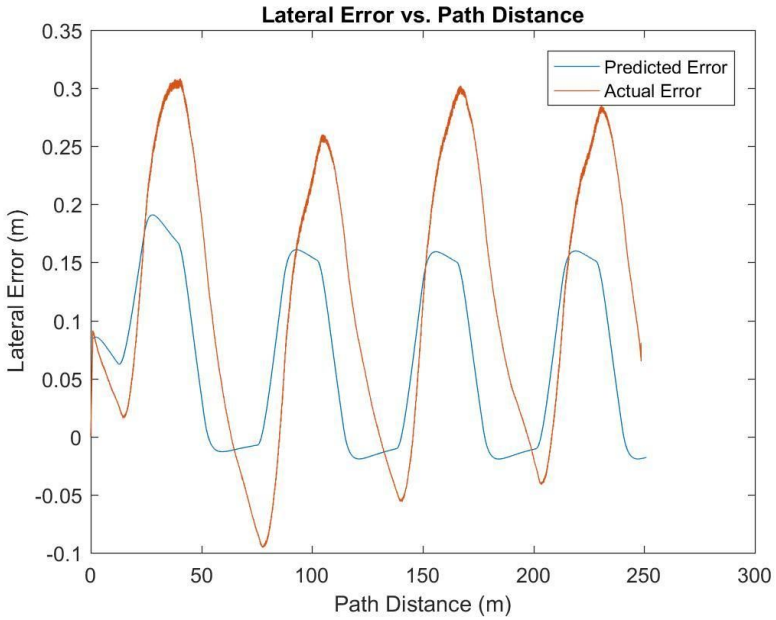


Figure 4: *Lateral Error vs Path Distance for uncoupled tire model*

To investigate whether our tire model might have been a factor in the difference between our simulation results and experimental results, we updated our simulation with a coupled tire force model. We see the results below in Figure 5.

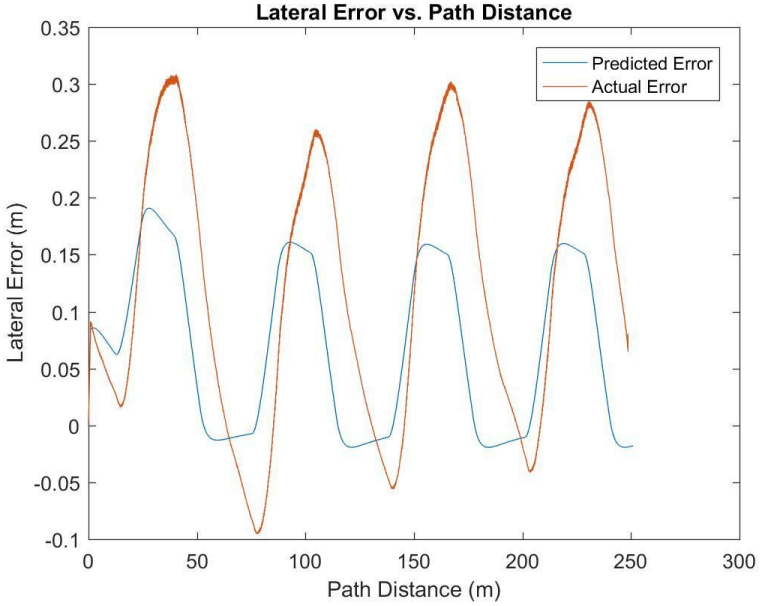


Figure 5: *Lateral Error vs Path Distance for coupled tire model*

Comparing the predicted error in the simulation in Figure 4 and Figure 5, we see there is almost no difference in our predicted error when we implement a coupled tire force model.

This is expected since the coupled tire model only diverges from our uncoupled tire model when the tires are operating close to the limits. At the limits, there is only so much force that can be generated at the contact patch. What is used for longitudinal force cannot be used for lateral force. Since our tires were not operating close to their limits at the turns, there was no competition between the lateral tire force and longitudinal tire force since there was enough force generated at the contact patch for both tires. Therefore, very little discrepancy between the simulation and experiment is explained by the impact of longitudinal forces. We determined that our differences between simulation and experiment were due to a variety of other factors including unmodeled system dynamics, delays in the steering and engine systems, and sensor noise. These factors are described in further detail in the next section.

3 Comparison of Controller Performance

One of the biggest issues we ran into was double-counting derivative control in our PID controller. Because we were including additional derivative control terms, our gains were larger than anticipated, driving our closed loop poles to instability. More specifically, we did not identify that the lookahead controller functioned as a proportional derivative controller. Assuming small angles and zero lateral velocity, heading error is directly related to $\dot{e}$ by equation 5 as described below.

$$\dot{e} = U_y cos\Delta\psi + U_x sin\Delta\psi$$

Equation 5: Rate of Change of Error

Using small angle approximations in heading angle and for small lateral velocities, the derivative of lateral error can be approximated by $\dot{e} = U_x\Delta\psi$. Thus magnitude of gain on our $\dot{e}$ term would be (kla*xla/Ux) from our lookahead controller in addition to our kd from our derivative term, resulting in a combined gain of $K_{la} * x_{la}/(U_x) + K_d$. Because the magnitude of our lookahead gain and lookahead distance are roughly 1000 times greater than our K_d term, the K_d term did not significantly contribute to reduction of lateral error on our straightaways, where small angle approximations and the no lateral velocity assumption were appropriate.

Another issue was that we implemented an alternative form of $\dot{e}$ in our controller. Rather than use the full form of $\dot{e}$ developed for tracking paths given by equation 5, which utilized the vehicle's parameters of motion, we estimated it using the difference in error for each timestep. This process of finding $\dot{e}$ amplified the noise in our lateral error calculation. Although our lookahead with feedforward controller performed reasonably well with this scheme, our PID controller did not.

Finally, despite the tuning we had performed using our simulation data, our gains proved to be too high on the actual vehicle. These large gains shifted our poles outward,

amplifying the noise in our system. We did not model some aspects of our system, such as delays in response and road noise, because those poles typically have less bearing on our system than our modeled dynamics like lateral and heading error. These unmodeled poles become more apparent as our modeled poles become less dominant at higher gains. Additionally, we did not model human comfort when adjusting our gains. Our gains tuned in simulation could account for a fairly rough ride before we turned the gains down.

Looking at our PID data from our physical runs, we found that the plots for lateral error had a large initial peak. We would typically set our initial lateral error to zero in the simulations, but the car had a nonzero lateral error during the tests. We hypothesized that matching these initial conditions would improve the discrepancies between our simulation and experimental data. After adjusting the initial values in simulation, we found that this only marginally improved the simulation results (Figure 6). Therefore, this peak in lateral error was likely caused by another factor, namely the integral term.

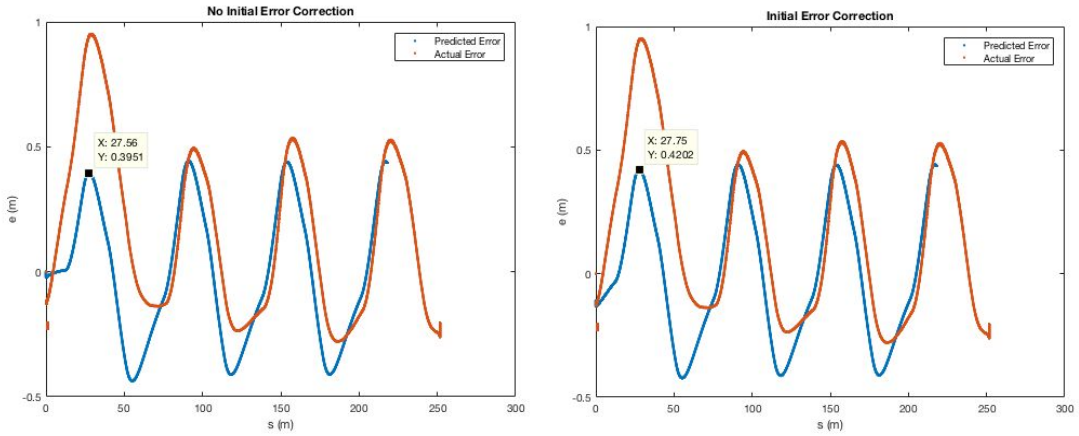
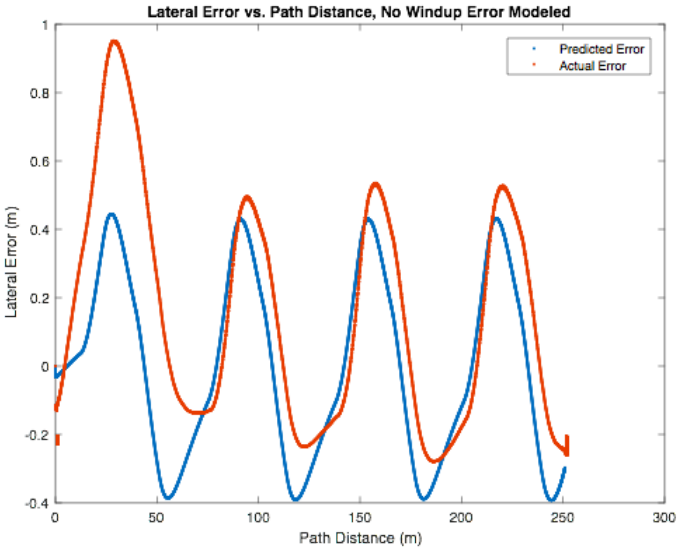


Figure 6: *Initial Error Correction*

Our PID controller for steer angle had more lateral error than our simple lookahead controller, and had a large error not modeled by simulation, as shown in Figure 7. We observed that for the first curve we had a large unmodeled error for the first turn. After examining our data and consulting with the teaching team, we identified that prior to activating our controller and moving forward, our car would idle and accumulate error for the first few seconds. We theorized that this integrator windup in the start condition was responsible for the large lateral error observed for the first turn, as we would be commanding an exaggerated steer angle for the first turn due to this windup. Since we did implement an anti-windup method involving zeroing the accumulation of error entering straight, the initial windup did not impact the following turns.



-

Figure 7: *Experimental Lateral Error vs. Simulated Lateral Error*

To understand how initial windup error impacted our vehicle performance, we modeled the effect of wind-up error in simulation by finding and adding an initial integral error term from our experimental data. By summing the error we accumulate before we started moving forward at 0.1 mph (0.05m/sec), we found the initial integral error sum to be -12.5567 ms. To simulate this error, we changed our initial integral error from zero to this offset value, and graphed the resulting lateral error, shown in Figure 8.

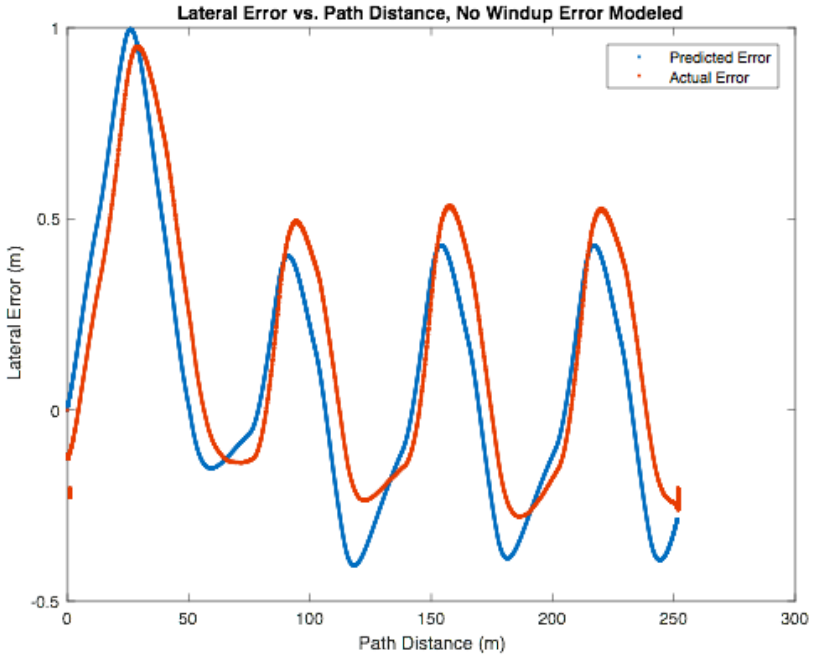


Figure 8: *Experimental vs. Simulation Given Integral Windup*

Although we still observe a phase lag due to time delay in response of command to steering angle, the simulation more accurately represents the initial experimental error for the first turn, confirming our hypothesis that integral windup was responsible for the initial spike in lateral error for the first turn.

While we were testing the vehicles, we felt oscillations in the longitudinal speed, particularly around the turns. We attributed these oscillations to two reasons - delay and not accounting for lateral force. Regarding the former, we realized that our longitudinal speed controller did not take into consideration the effects of delay (specifically steering and engine delay) and of unmodeled system dynamics. Regarding the latter, we did not model the centripetal force that acted laterally on the car through the turns (i.e. we did not have a yaw term in our controller). Since this lateral force reduced the vehicle's speed, the longitudinal speed controller needed to output a greater force at each step to maintain a constant speed through the curve; this resulted in a feeling somewhat jerky motion in the car. This oscillation in longitudinal speed can be seen in Figure 3 in the previous section. Since there were no lateral forces acting on the straightaways, they felt fairly smooth in contrast.

Ultimately, we preferred just the simple lookahead controller over our PID controller. The PID controller seemed to amplify noise in our system instead of minimizing it. If we had tuned the gains in the PID controller perfectly, it might have further minimized the lateral error from the original lookahead. However, by adding integral and derivative terms to the lookahead, we are adding more states, and more complexity to our model, making it harder to tune the gains.