

Group 7 – ME 227 - Team Oh Niki You're So Fine

Names (Alphabetical) - Andrew (arbaird4@stanford.edu), Courtney (moranc@stanford.edu), Satyan (satyanc@stanford.edu), Sergio (sergio16@stanford.edu)

Table of Contents

1. Executive Summary
2. Gain Selection
 - a. Lookahead
 - b. PID
3. Sim-Mode Comparison & Controller Performance
 - a. Sim-Mode 0 (Lookahead & PID)
 - b. Sim-Mode 1 (Lookahead & PID)
 - c. Sim-Mode 2 (Lookahead & PID)
 - d. Sim-Mode 3 (Lookahead & PID)
4. Learnings
5. Sources of Noise & Other Observations

Notes - (1) only some plots are shared below. More plots are uploaded in the code section of gradescope for plot results for simulations and gain selections.(2) Also, initially, the runtime was changed to 40 seconds to ensure we capture end of simulation dynamics. Some plots may show a 40 sec runtime, but the code is at 35 secs.

1. EXECUTIVE SUMMARY

Our team developed two controllers - one, as the project outlined, a feedback/ feedforward lookahead, and two, a PID controller. At a high-level, we aimed to meet project requirements while maximizing user/ driver comfort i.e. minimize the following - lateral and longitudinal accelerations, steering and position oscillations, overshoot (and subsequent correction) and control instabilities (jerk, high initial steering offset).

Our lookahead controller was able to achieve a delta U_x of approx 0.5 m/s, total peak acceleration magnitude of 4 m/s², and lateral error of 0.17 m with minimal to no parasitic steering oscillations and acceptable lateral dynamics (U_y , yaw). Our PID controller was able to achieve a maximum error of 0.37 m/s in U_x , lateral error of 0.15m, and the acceleration magnitude stayed within 4 m/s², apart from noise and a small spike at the first turn. In both cases, while controller performance could have been improved (for tighter response, less error, better speed tracking) through more aggressive gain selection, we modulated our gains in meeting our high-level goals of driver/ user comfort. Our results allow margin for unmodeled effects or other errors/ non-linearities not captured.

While working on the project, we also developed an appreciation for controls, and our vastly limited (yet growing) knowledge in vehicle dynamics.

The following report first discusses notable differences in the sim-modes, dives into gain selection, and then compares our controllers. At the end, we include our micro and macroscopic learnings, other observations, and areas we still need to develop better intuition and reasoning.

2. GAIN SELECTION

Gains for the lookahead controller (k_{la} , x_{la} , and k_{lo}) and PID controller (k_p , k_d , k_i) were selected in 3 steps.

LOOKAHEAD

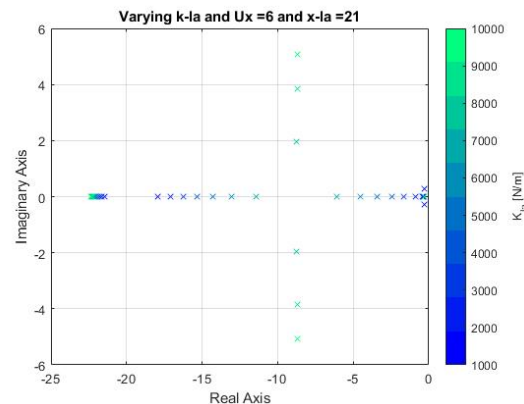
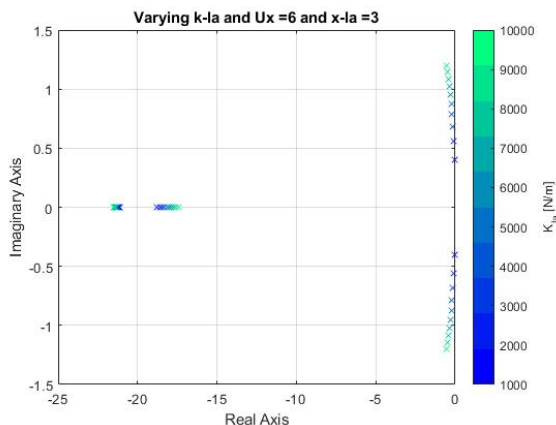
Step 1 - wide spans of k_{la} and x_{la} were evaluated for pole position, natural frequency and damping response, and pole stability using state space derivatives derived from laplace representations of key transfer functions (for lateral and heading errors). k_{la} of 1000 to 15000 N/m was accessed with x_{la} of 3 m to 30 m in 3 meter increments to establish bounds. k_{lo} was estimated initially at $\frac{1}{4}$, $\frac{1}{2}$, and $2x$ the feedforward F_x required to maintain the speed profile. Two speed levels were also used of $U_x = 6$ m/s and $U_x = 15$ m/s to capture system responses at the low and high bounds of speed, and tune gains for good performance in the corners as well as in the straights.

Step 2 - wide bounds showed clear trends to narrow down pole/ zero selection

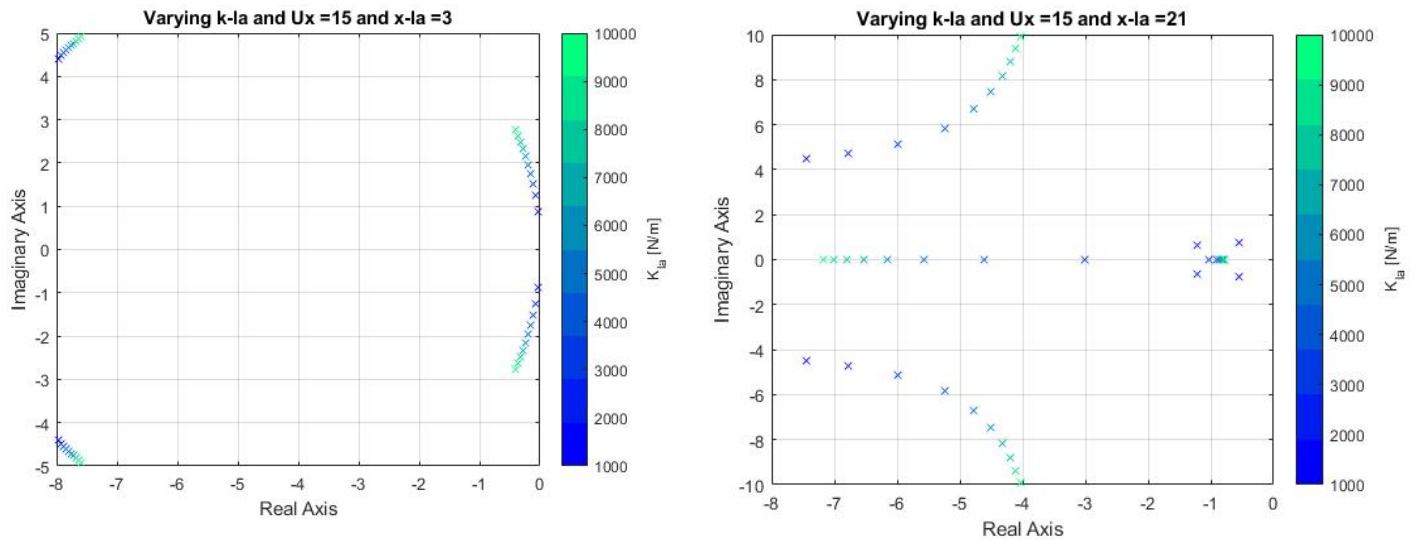
- At x_{la} below 4 meters, poles shifted into the right half s-plane for low k_{la} gain values, rendering an unstable system.
- At low x_{la} and high k_{la} values, though the dominant poles were in the negative half s-plane, damping was very low.
- At high x_{la} and high k_{la} values, the response became oscillatory (and oscillations were evident in lateral error plots).
- At high x_{la} and low k_{la} values, though the system showed stable response at $U_x = 15$, at $U_x = 6$, dominant poles were too close to the right half s-plane.

The pole zero and root locus plots (some pasted below, others submitted in Gradescope) ultimately showed that the system was most stable (fast response, minimal oscillations, tracking speed, acceleration, and lateral position adequately) between an x_{la} of 6.5 to 8 m, a k_{la} of 8000 to 9000 N/m, and a k_{lo} of between 2000 and 2500 N.s/m (proportional to approx $\frac{1}{2}$ the feedforward term).

(Figures below at $U_x = 6$, $3 \text{ m} \leq x_{la} \leq 21$, $1000 < k_{la} < 10,000$)



(Figures below at $U_x = 15$, $3 \text{ m} \leq x_{la} \leq 21$, $1000 < k_{la} < 10,000$)

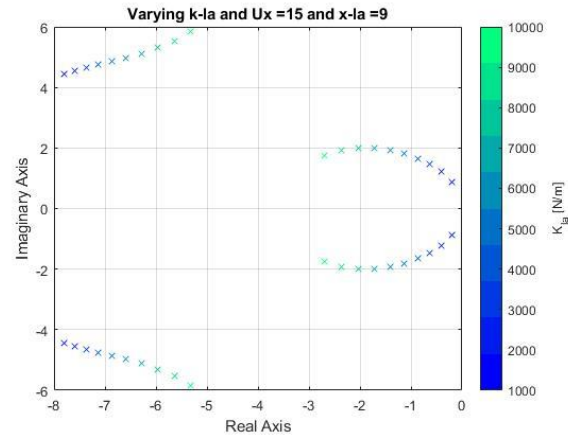
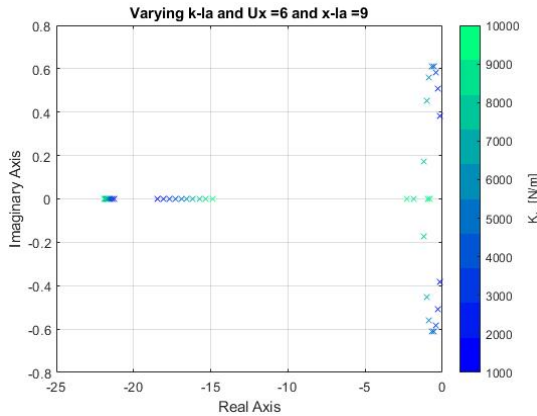


Step 3 - final gains of x_{la} of 7.8 m, k_{la} of 8600 N/m, and k_{lo} of 2308 N were selected. Pole-zero plots below show final range selection for x_{la} and k_{la} that satisfied both $U_x = 6 \text{ m/s}$ and $U_x = 15 \text{ m/s}$.

From step 2, once the gain bounds were tightly windowed, some manual tuning and iteration was needed for x_{la} between 6.5 and 8 m, k_{la} between 8000 and 9000 N/m, and k_{lo} between 2000 and 2500 N.s/m. For example, minor changes to k_{la} led to changes in lateral acceleration, above bounds. Similarly, minor adjustments to x_{la} affected lateral error, yaw, and lateral velocity non-trivially. Step 3 consisted of final tuning to achieve the right x_{la} , k_{la} , k_{lo} balance. These gain values we ended up with satisfied both 6 m/s and 15 m/s for being closest to the negative real axis, minimizing oscillations with as much damping as was feasible, and achieving moderate frequency response (so as to achieve response requirements but not cause control jerk or excessively fast response).

Managing insignificant poles was also considered (far left of negative half s-plane) to minimize their Imag. axis position and achieve marginally more stable response.

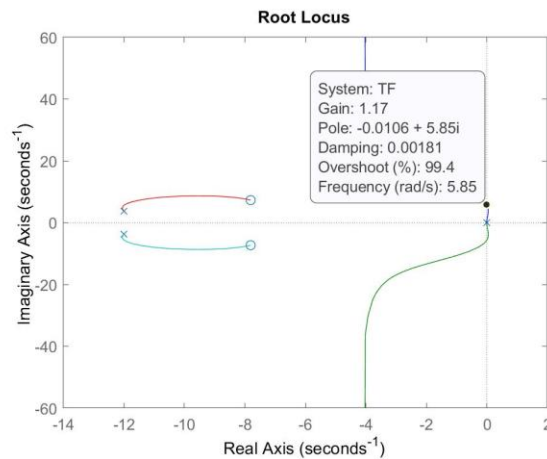
(Figures below show x_{la} at 9 m for U_x of 6 m/s and U_x of 15 m/s, showing the tightened window of gains we ended up on)



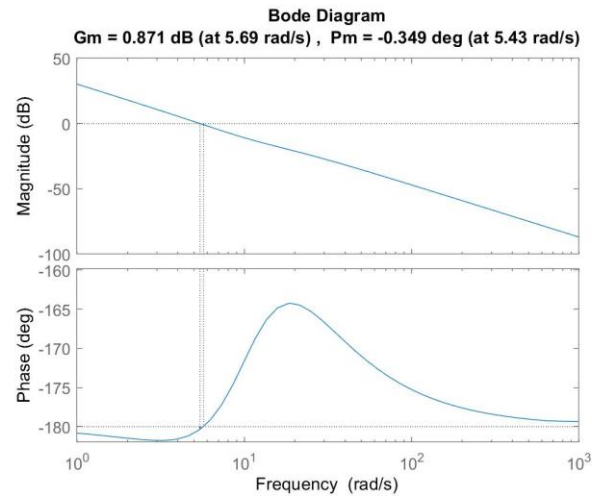
PID

Gains for the PID controller were first obtained using the Ziegler Nichols tuning method but were then adjusted with a few iterations with the help of root locus and pole location plots.

Step 1 - Using the transfer function of the lateral error to the steering angle, we plotted the root locus with a simple proportional gain to find the ultimate gain and period, the gain that makes the system marginally stable and the period at that point. This is the first step in Ziegler Nichols.

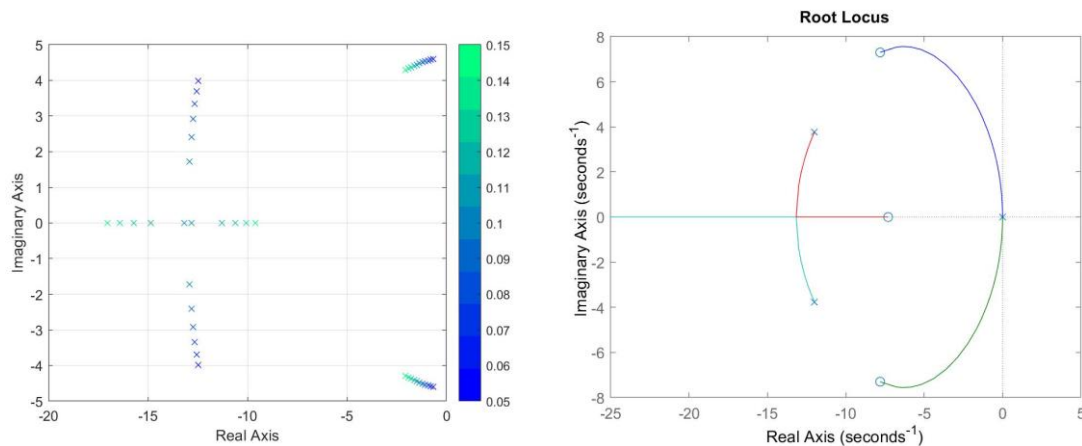


Using only the root locus made it hard to pinpoint this value and plotting the step response of the closed loop system was hard to read to gather the ultimate period so we made use of a bode plot of the transfer function!



The Bode diagram more easily displays the point where the transfer function goes marginally stable; this is the gain margin. Converting from dB we can find a more exact value for the ultimate gain and by converting the phase to radians we can find the ultimate period! At this point we turned to a Ziegler Nichols table to put us in a ballpark range for our PID gains.

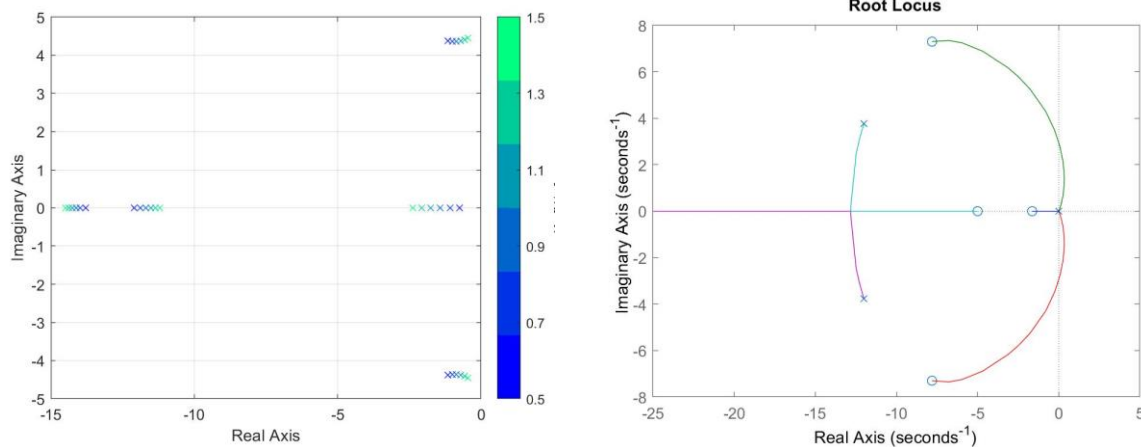
Step 2 - Even after getting values from the Ziegler Nichols table, our readings of steering angle were very oscillatory and lateral error was out of spec, so there was need for some more tuning. We decided to trust the Kp gain for now, using intuition for its units of rad/m and making sure that its value was not an absurdly large input to the car that would oscillate dramatically or insanely small that it would keep lateral error out of spec. We then built up the Kd gain by using the state-space matrix for a PD controller ($\delta = -(K_p * e + K_d * \dot{e})$) and varied the gains of Kd using a fixed Kp from earlier and $U_x = 10\text{m/s}$.



We also used the root locus of the transfer function with a PD controller, fixing Kd at a value in this range, just to get a sense of the direction of the pole locations because in some instances it was not very clear. Multiple ranges of Kd were attempted using this method until the dominant poles either landed on the real axis, or if they could not, they were as close to the real axis as possible (local minima) so as to reduce oscillations in the system. The values on the caxis are not indicative of our final Kd value.

Step 3 - After obtaining a rough Kp and Kd value, it was now time to add the Ki value! We expanded our state-space matrix from before to include a PID controller ($\delta = -(k_p * e + K_d * \dot{e} + K_i * \text{integral}(e))$)

and varied the value of K_i , plotting the pole locations and root locus to get a sense of location and direction of the poles.



We used the same method as before with the K_d in that we varied K_i over a certain range, using the root locus to guide the direction, and landed at a K_i value that plotted our dominant poles on or as close to the real axis as possible. The values on the axes are not indicative of our final K_i value.

Step 4 - Do it all again! The original batch of tuning doing this method got us better performance than the Ziegler Nichols tuning method, but there were still some major oscillations in steering angle and we were running close on the lateral error spec, so we iterated a few times on these values using the same methods with varying K_p values to try and balance good tracking with minor oscillations in the steering angle for a comfortable ride.

Note - We recently went to office hours and found that our term to numerically integrate over the lateral error for the K_i term was incorrect and needed to be fixed. This required our K_i value to need to be amplified by a few orders of magnitude and seemingly necessitated the amplification of its fellow gains. Once this adjustment was made we went through all of the steps highlighted above again, checking on the root locus and pole locations as we varied each value one-by-one to arrive at a solution with good tracking and a comfortable ride (we hope). A figure in a later section displays that the poles of our controller are stable but just a bit oscillatory.

3. SIM_MODE COMPARISON & CONTROLLER PERFORMANCE

Note - We began directly using the Hard Simulation since we had already spent time with HW 4's ME227 Simulator following a straight, curved, and undulating path.

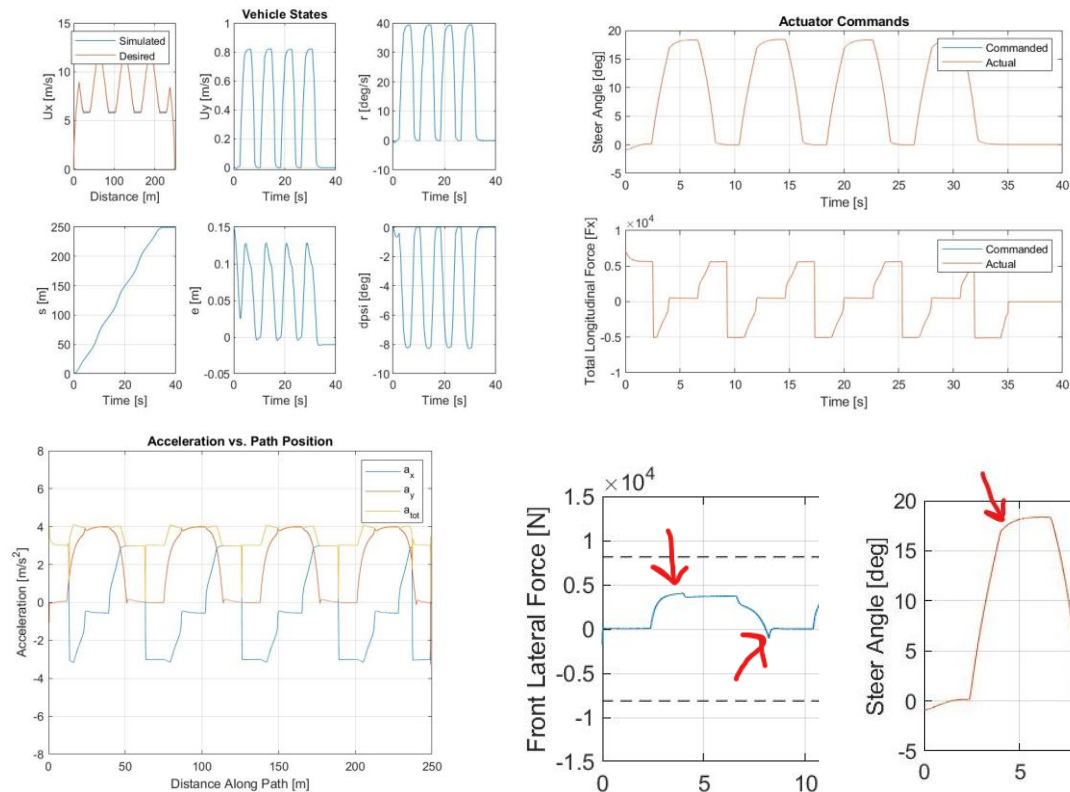
SIM-MODE 0

Sim-mode 0 started with an idealistic, impractical control environment (i.e. noiseless). A noiseless (or near no noise) environment meant that with our gain selection, controller tracking was superior on error, velocity, and actuator commands. This sim-mode results in the vehicle behaving exactly how it is commanded, giving us a general idea of how the vehicle will perform. No major external disturbances

were present on the sensory inputs, and our nonlinear models tracked the path, speed, and acceleration profiles we set as inputs.

Lookahead Performance

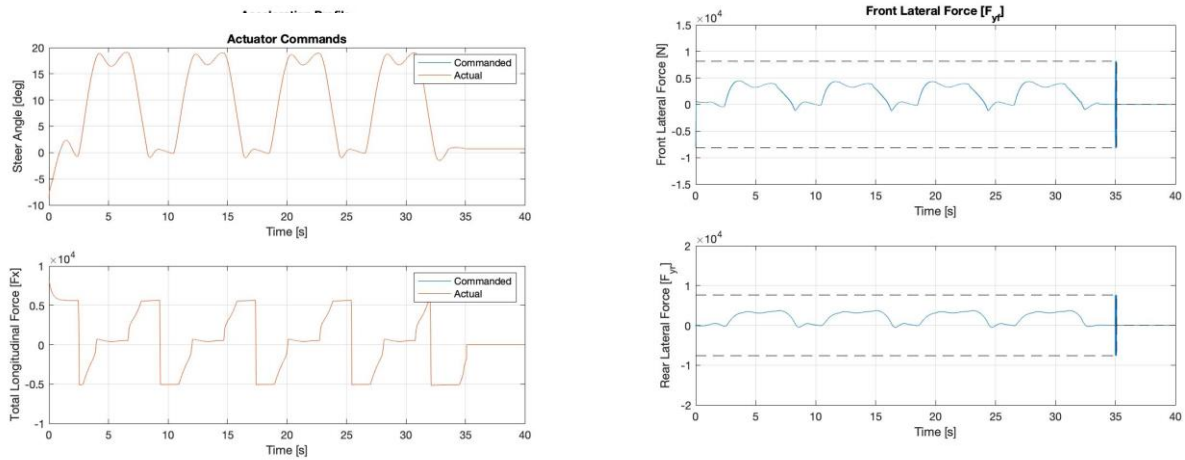
1. As expected, we noticed that $\mathbf{a_x}$ during braking was instantaneous; and that $\mathbf{a_y}$ only presented itself at the beginning of the clothoid, and persisted through the corner exit
2. Lateral dynamics were stable - error tracking had very minor overshoot at mid-corner but $\mathbf{U_y}$ and \mathbf{r} experienced no parasitic low or high frequency oscillations, overshoot, unsettling behavior
3. $\mathbf{U_y}$, \mathbf{r} , and \mathbf{dpsi} were highest in magnitude mid-corner as expected.
4. $\mathbf{F_y}$ had a slight dip at around 4 seconds after clothoid exit and constant R entry from corresponding reduction in steer angle slew; and an overshoot after corner exit.
5. Gain selection for sim-mode 0 was satisfactory as gain selection was primarily dictated by sim-mode 3.



PID Performance

1. A key difference seen in the PID control is that the steering forms more prominent peaks when the car enters/leaves the curves. This is seen in both the acceleration and steering actuation plots.
2. Another key difference is that the steering overshoots a little when the car reaches the straights (again seen in the steering actuation plot).

3. The lateral error, e , oscillates at a higher frequency, and has smaller peaks between the dominant ones.



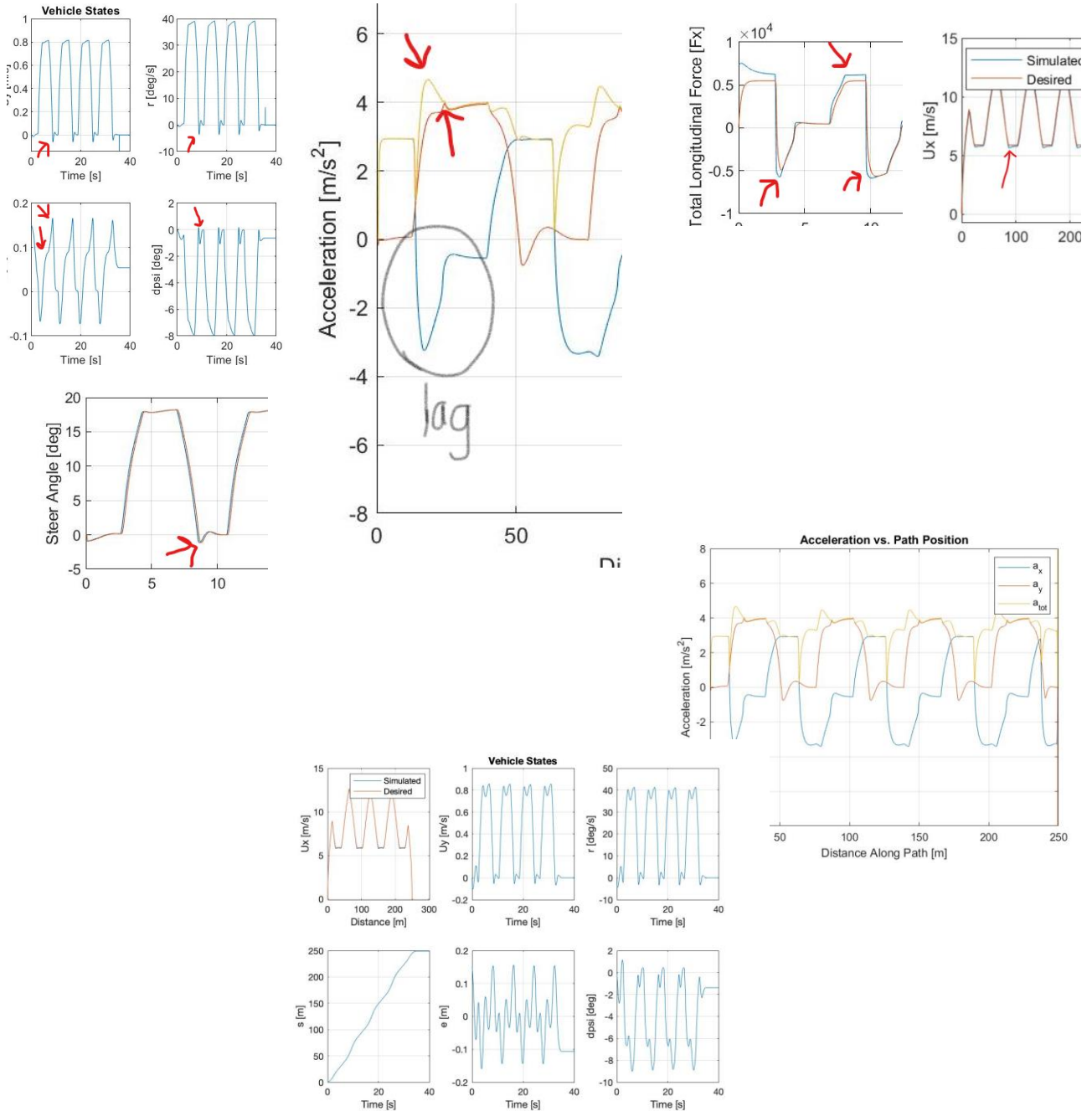
SIM-MODE 1

Sim-mode 1 added actuator dynamics between steering (and lateral forces) and longitudinal forces, effectively fighting for traction and path control. In this case, the actuator outputs are lower in magnitude than the commands output from the controller for the longitudinal force. This results in the vehicle being less responsive when it comes to tracking as compared to sim-mode 0. System response overall weakened, overshoot grew, and some non-threatening oscillatory dynamics emerged.

Lookahead Performance

1. **ax** and **ay** saw both latency and overshoot - **ax** had latency in deceleration/ braking (as expected) with no overshoot, but **ay** had minor overshoot oscillations after clothoid exit and at constant R entry. **a-total** had overshoot behavior because of the latency/ delay in **ax** which superimposed with **ay** meaning that we were braking into the corner/ clothoid, and **were not** just braking at the straights (like sim-mode 0).
2. **ax** also saw minor overshoot after corner exit at straightway entry, as cruise control tracked and maintained speed
3. Steering angle **delta** saw overshoot at corner exit, recurringly, with quick settling and recovery. This can be attributed to CG and front axle having different positional lateral errors i.e. steering angle will keep correcting (even if it's on path) until CG's not on path.
4. **Fx** commanded vs actual difference grew at clothoid entry, and corner exit, and actual **Fx** lagged commanded **Fx** i.e. vehicle decelerated before command. Behavior at corner-exit may be attributed to steering angle **delta** overshoot, and development of **Fy** which opposes **Fx** with non-zero steer **delta**, lowering **Fx** actual. Behavior at clothoid entrance is attributed to perfect vs imperfect system response i.e. errors with cruise/ speed (**Ux**) control tracking on Niki vs in a simulation - these errors may be stacked up from GPS measurements for example, leading to **Ux** deviation at the same time **Fx** deviations occur between command and actual

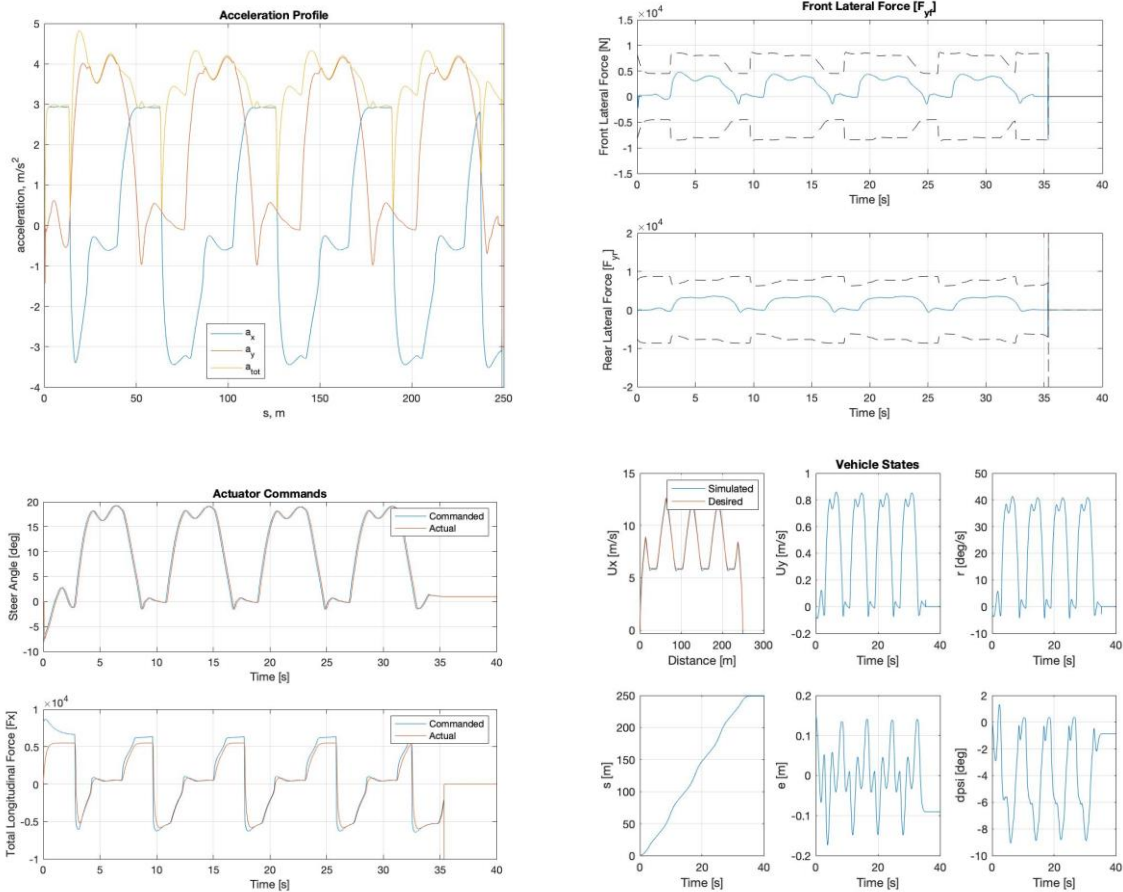
- Oscillations in Sim-Mode 1 grew for **Uy**, **e**, **dpsi**, and **r**, but always (and interestingly) at corner exit. This is attributable to, as before explained, the difference between lateral error **e** at CG and Steer Axle. With that, overshoot grew in **Uy**, **dpsi**, and **r**, but response and settling from gain selection was rapid so overshoot correction oscillations didn't materialize in **e**.
- We see the same correction in **e** right after clothoid exit and before corner entry with a reduced de/dt slope in the states plot for **e**. This correction isn't present in Sim-Mode 0



PID Performance

Similarly to the lookahead controller, the actuator outputs from our PID controller are not quite able to match the command magnitudes, particularly for longitudinal commands. The acceleration profile for PID control looks nearly identical in sim-mode 1 as in sim-mode 0, with a few exceptions. Similar to the effect described in the look-ahead controller, we see a higher overshoot in **a-total** as we enter the first turn. We also see deeper and more rounded dips in longitudinal acceleration (**ax**) throughout the simulation during braking, resulting in corresponding peaks in the a-total. Whereas in sim-mode 0 there were only two primary peaks for each turning maneuver, this effect adds a smaller, third acceleration peak as we enter each turn.

Despite the presence of additional longitudinal and actuator dynamics in the model, the PID controller was able to track with an almost identical error for sim-mode 1 as in sim-mode 0 for both lateral and heading error, with error margins in a range of $(-0.15\text{m}, 0.15\text{m})$ and $(-9^\circ, 1^\circ)$, respectively. It also closely tracks the desired speed profile. Overall, the PID controller is highly effective on this model.



SIM-MODE 2

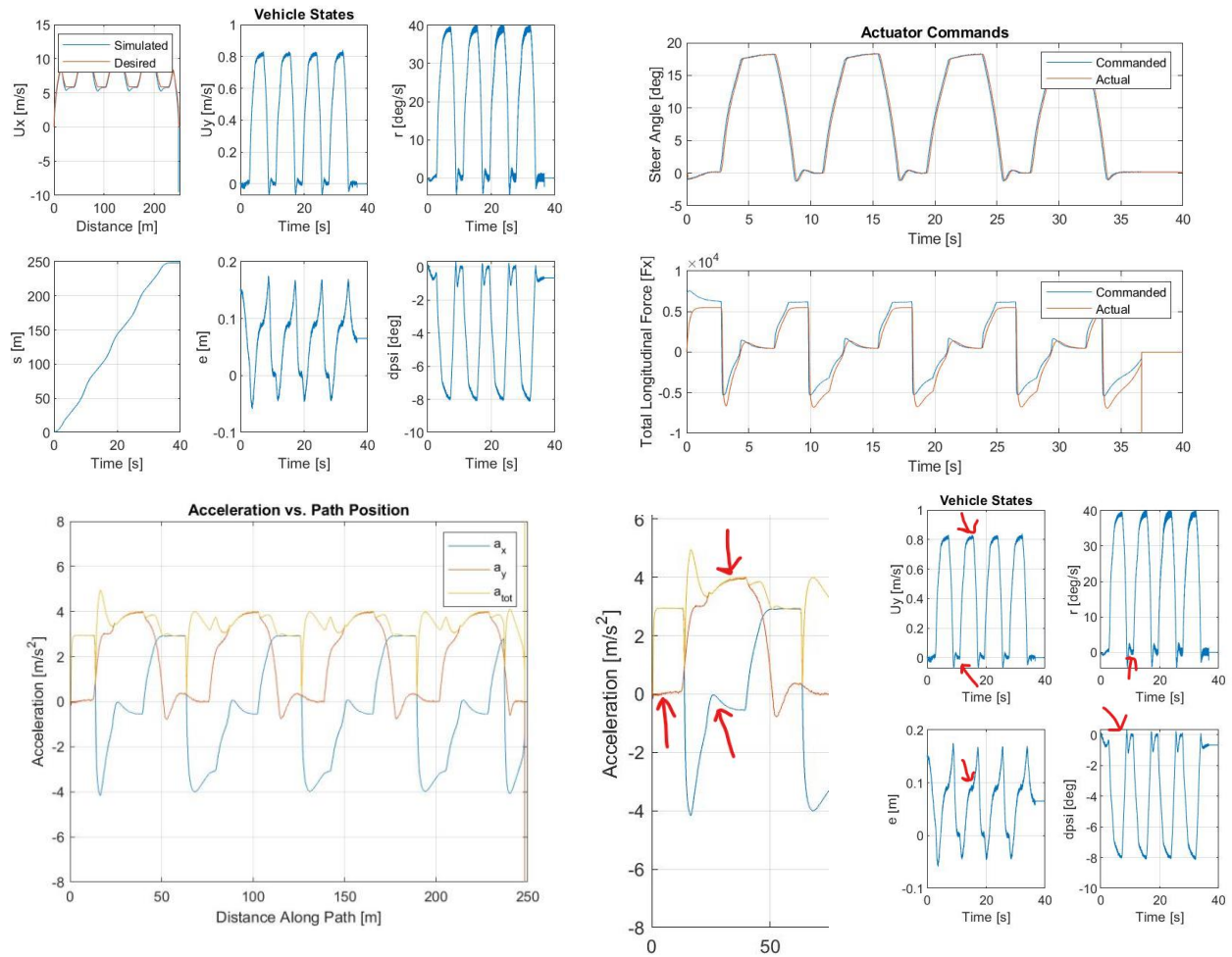
Sim-Mode 2 added noise, and response weakened further with gains reacting to noise (and amplifying inputs and outputs). This noise primarily impacts the steering. The result is some small jitters in the

steering commands from the controller. The output from the actuator smooths out these commands a little bit, however. Another difference in simulation mode 2 is that the actual longitudinal force applied is always lower than the commanded. This indicates that the car brakes more than is expected when it slows down. This results in a little bit more error in the speed profile tracking.

Lookahead Performance -

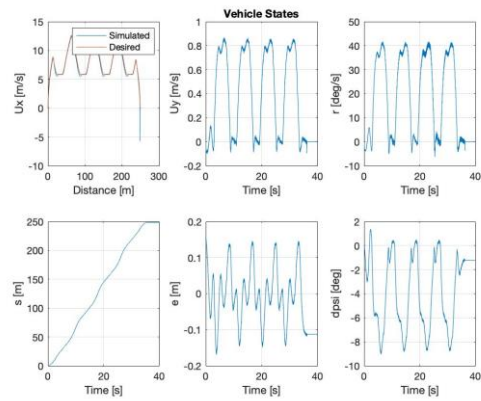
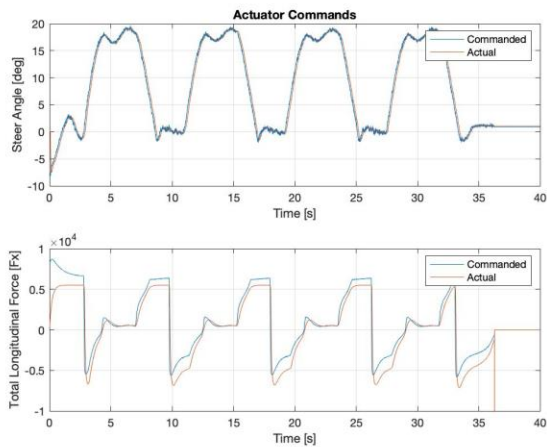
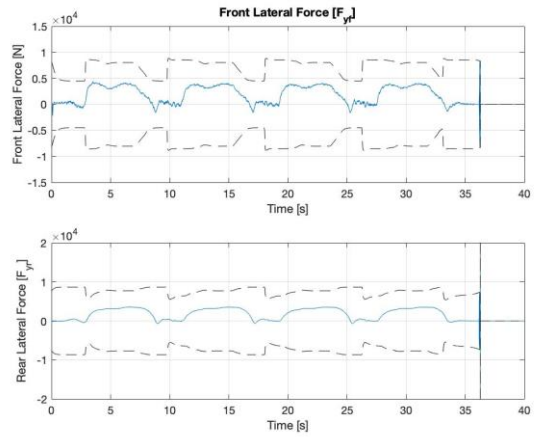
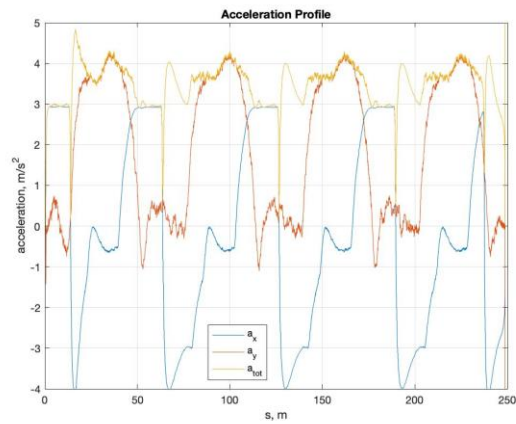
1. Overall, there was more noise, parasitic oscillations, and heightened response to noise in Sim-Mode 2. **A_x** saw more overshoot in recovery near corner exit and **A_y** had superimposed oscillations (of minor scale). We also observed a spike in **A_y** at the end of the simulation.
2. **F_x** and **U_x** had interesting trends. We observed that Niki deviated, while braking in both ends of the corners, from the commanded longitudinal force i.e. Niki braked later in actuality. This can be attributed to brake response lag, hydraulic latencies, actuation latencies (on brake booster, pedal travel, caliper piston travel), and stuck-unstuck state of brakes. This **F_x** lag led to a corresponding **U_x** deviation.
3. Similar to Sim-Mode 2, steering **delta** also witnessed a response to noise injection (mostly through marginal increase in overshoot and settling dynamics) but to a much lesser extent than we expected.
4. Noise in **U_y**, **r**, **e**, and **dpsi** also rose and oscillations were detected at mid-corners. Excessive gain values would've scaled the noise response higher, so managing our response to injected noise was important. Interestingly, the noise, which is always present in reality, only presented itself mid-corner during direction reversal. This is because the lateral dynamics (yaw, velocity, heading, lateral error) trajectories coincided with the noise frequencies and only when the rate of change of these lateral dynamics lowered (mid-corner) did the noise present itself in the graphical data. In reality, noise will always be present, and you wouldn't necessarily see noise only during certain maneuvers.

Plots on next page.



PID Performance

Going into sim-mode 2 the system became a lot more oscillatory due to noise now being injected into the system. Despite the noise, our controller was still doing a fair job of tracking the lateral error, and our acceleration spikes, although a bit high, looked the same. One of the biggest concerns though was the introduction of small oscillations in the steering angle. We became concerned for the comfort of the ride since these oscillations could jerk the car if left too large. It then became a balance of trying to flatten the divot at the top of the steer angle curve but also trying to avoid the oscillations at the trough of the curve since they were pretty heavily tied together. We got these oscillations to what we feel might be a comfortable place but are looking to still improve between now and Saturday. The primary suspect for these oscillations is the difference between our commanded and actual total longitudinal force which is related to noise being plugged into the braking system of the vehicle as mentioned above. We are no longer in the realm of previous sim-modes where braking is instantaneous and exact, and as such we see some offsets in what we want and what the brakes could possibly deliver.

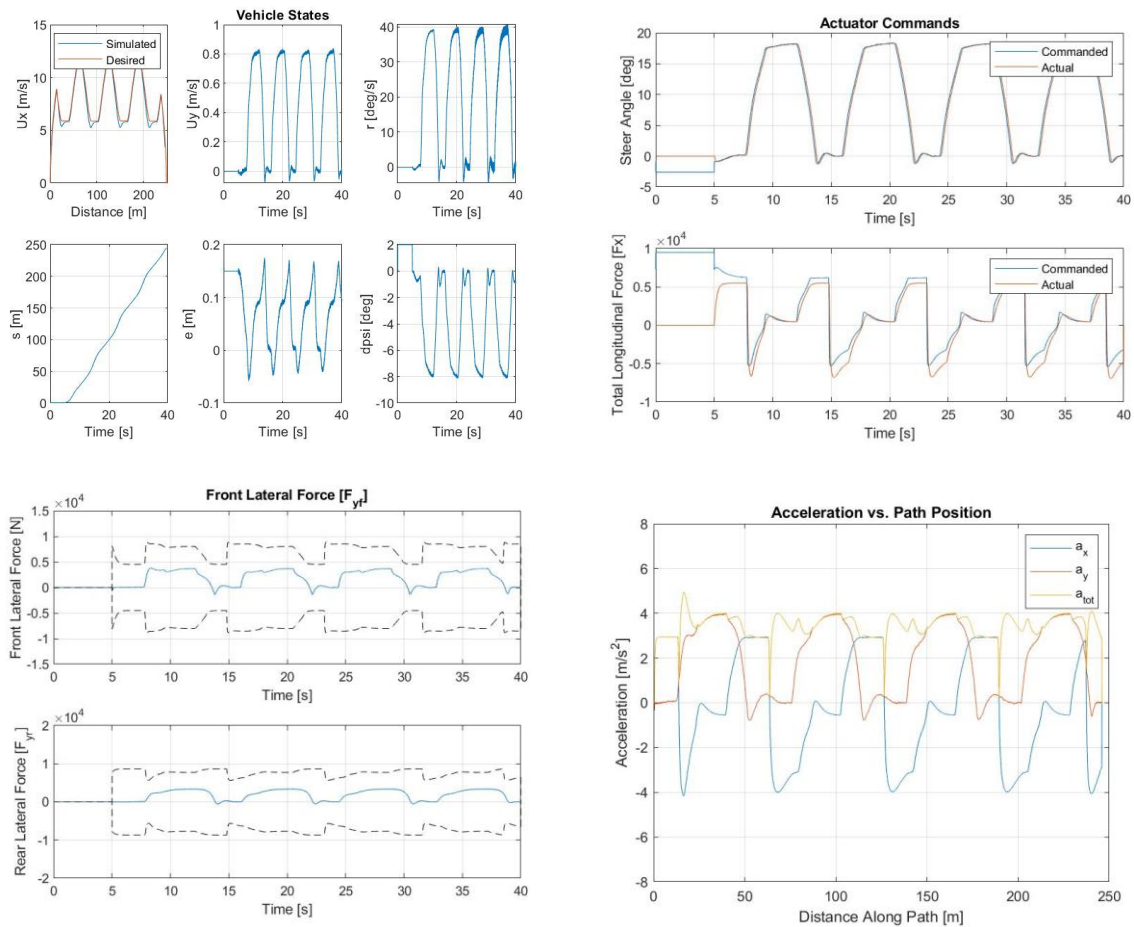


SIM-MODE 3

Sim-Mode 3 primarily differed from Sim-Mode 2 in the addition of a 5 sec hold period.

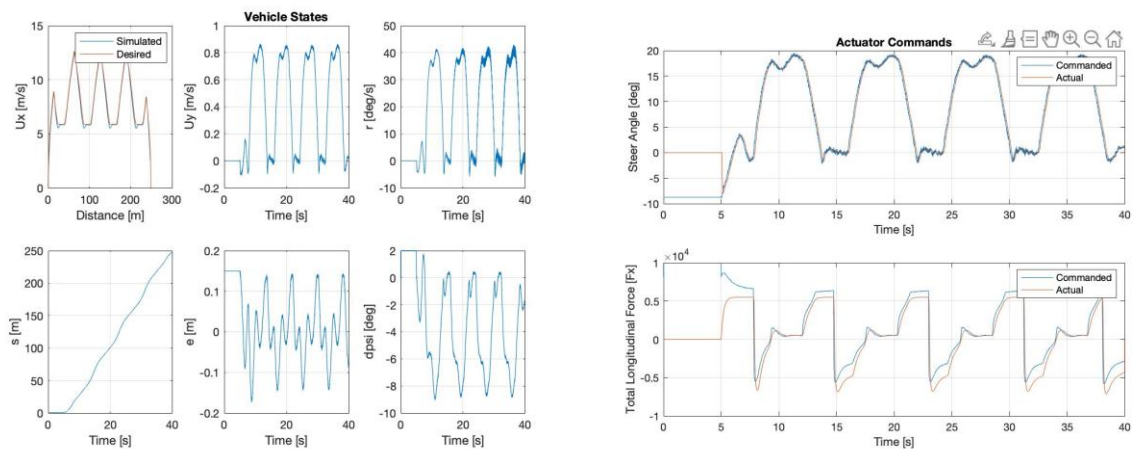
Lookahead Performance

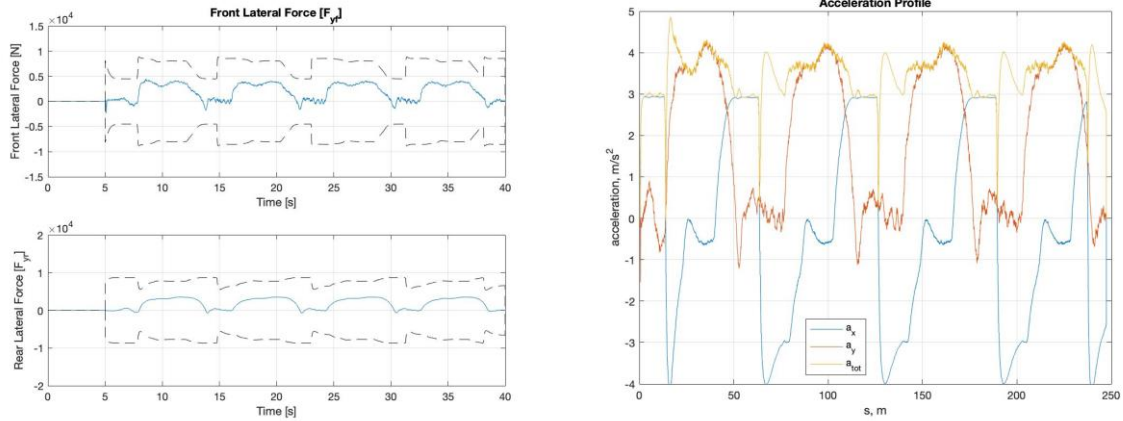
Unlike the PID controller, Sim-Mode 3 had minimal effect on the lookahead controller given absence of windup and that the lookahead controller evaluated absolute error and absolute error derivatives at each timestamp (adjusting actuation accordingly) - there was no error accumulation.



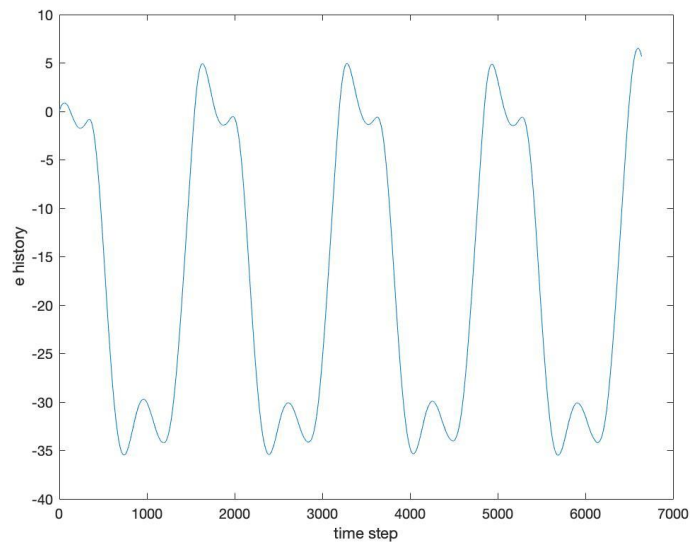
PID Performance

In sim-mode 3 we saw a lot of the same behavior as that noted in sim-mode 2 with some noisier responses, attributed to similar but noisier inputs, but the introduction of the 5 second delay created the biggest problem for our PID controller, specifically the integral term.





The 5 second delay doesn't impact the lookahead controller too much, but since the PID controller has an integral term which accumulates error, this causes windup. Our integral gain was applied to simply sum all previous error terms, and as such, sitting at an offset for 5 seconds would cause a large buildup of error that the vehicle tries to rectify immediately after starting. To mitigate this issue, we included an anti-windup statement, which resets the accumulated error term when it exceeds a certain threshold. To choose an appropriate threshold, we created a plot of the accumulated error in the system without the 5 second hold.



Accumulated error vs Time

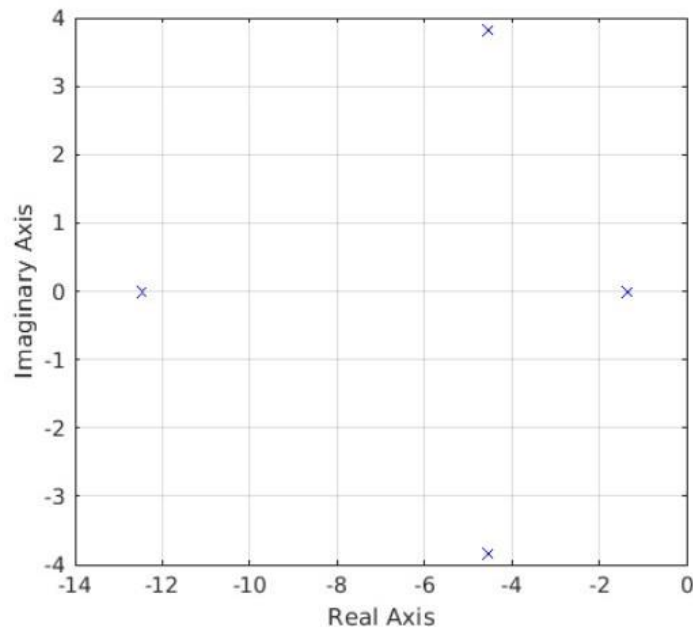
From this plot, we see that while going around the track, the accumulated error has a maximum magnitude of around 35. From this, we decided to use a threshold of 40. After implementing this anti-windup scheme, the system became stable again, but the wheels would turn back and forth pretty sharply while the car was sitting still. To get around this, we included another statement such that the integral term would only sum error once the car started moving. At first we set the condition as $U_x > 0$, but after considering noise in the system/measurements, and that the error could be high initially, we changed this threshold to 4m/s. This means the integral term of the controller doesn't kick in until the car is moving and approaching the first turn

4. Learnings

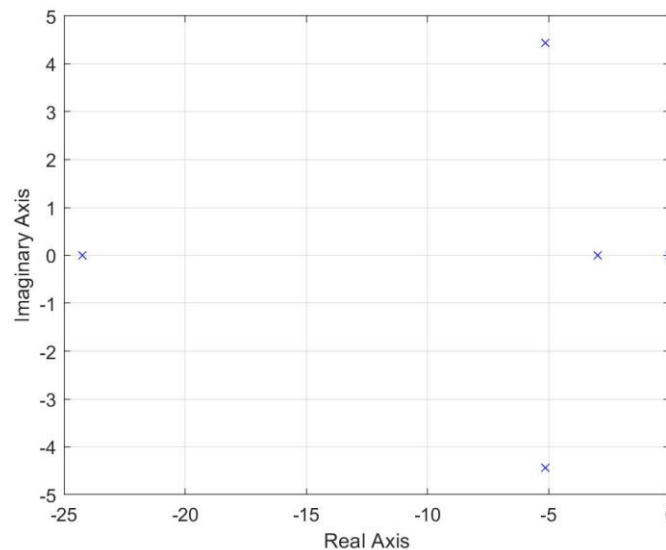
With regards to most categories the lookahead controller performs better than the PID controller. Looking at the lateral tracking error the lookahead experienced less oscillations throughout the path and had smaller amplitudes as well. Taking a look at the acceleration values and steering commands through the path we also noticed that the lookahead controller is a lot smoother and more robust compared to the PID controller which introduced a lot of micro-oscillations in the same values in response to the injected noise of sim-modes 2 and 3.

The one aspect that the PID controller performs better on is the speed tracking. In this regard the PID follows the speed very closely with small offsets at the curves as a result of the braking system noise, while the lookahead controller exhibits similar behavior but is less quick to correct itself to the desired speed. In summary, the lookahead controller is much more robust than the PID controller and will result in a smoother ride. Such results would beg the question if a lookahead component addition to the PID would help with the oscillations in acceleration and steering angle.

LOOKAHEAD CONTROLLER POLES



PID CONTROLLER POLES



We also explored dynamic gains on the lookahead controller where gain values for the straights (at $U_x = 15$ m/s) would be lower than that mid-corner ($U_x = \sim 6$ m/s), realizing that autonomous vehicles may rely on continuously varying control dynamics. For the LO and HI gains at the 2 speed settings, better response was observed **after** the switch from LO to HI gains, but at the transfer point from LO to HI gain (at HI and LO speeds), high transients were observed with an un-ramped, step change in gains.

If we look at the poles of the PID controller, the two that are not on the real axis are further away from it than the ones in the lookahead controller. This explains why we saw more oscillations. The system is less damped. The PID controller also has a pole on the real axis that is much further away from the imaginary axis. This means it has a minimal effect on the control of the system.

In the process of tuning the PID controller, we observed that the system was sensitive to changes in K_d . The system would work reasonably well for a relatively small window of K_d values, and then either go unstable, or have unreasonable oscillations outside of that window. This made tuning tricky for minimizing steering oscillations on the straights. The vehicle would overshoot the turn when it entered the straight again, and oscillate a little until it settled back at a heading of 0. In theory, this should be fine because the system is within the bounds of the error specifications, but in reality these oscillations can be problematic as it makes the drive quite jittery for the passenger.

5. SOURCES OF NOISE & OTHER OBSERVATIONS

NOISE

We wanted to also share some insights, in a separate section, on the sources of noise, where they may come from, how they impacted controller performance, and what we may do further to improve performance

First, noise sources are many in vehicles and automotive - inductive and capacitive coupling, radiated emissions from switching power electronics, low frequency noise from mosfets/ power supplies, antennas, FM radio, high frequency wave matter (GPS, LTE, XM radio) and so on. The list is very long.

Often, signals that are in close proximity to noise sources (either voltage or current modulated noise), pickup noise that leads to poor SNR (signal to noise ratios), causing high transients in the signal (whether that's GPS measurements, or speed, temperature, pressure, force, torque sensors).

The lookahead controller was overall more robust to noise. At selected gains, we saw system response worsen in terms of lateral dynamics, speed overshoot, and longitudinal force tracking, which led to longer settling times. This oscillation and overshoot may be uncomfortable for an onboard passenger or driver. Further experimentation would be necessary to close loop on more gain adjustment vs user experience.

The PID controller was more susceptible to noise primarily because of windup and our windup adjustment. Any built up error, as much as mitigated, leads to unstable initial response which propagates to some extent as PID control kicks in. In PID, the primary noise consequence was steering angle oscillations which ultimately led to oscillations in acceleration and lateral forces; and since lateral force oscillations affect longitudinal force demand, oscillation propagation from noise was throughout.

Last, on how we may address these in simulation and on the car. On simulation, filtering noise inputs would be the best course of action. Depending on the modal frequencies, and noise harmonics, low pass, high pass, or band pass filters can be added to smoothen, average out, or filter out band noise.

On the car, adequate electrical shielding, sufficient termination, minimal to no impedance mismatch are some ways to reduce the effect of parasitic noise.

OTHER OBSERVATIONS

Our primary concern for implementation is with the steering oscillation in the PID controller. With our current version, in simulation the steering overshoots by about 1° when it enters the straights, then steers back by a little more than 0.5° in the opposite direction (about 1.5° turn). The steering then turns back and forth in diminishing oscillations with a second peak of about 0.5° . In the animation, these turns don't appear very drastic, but as we learned from the teaching staff, this could still lead to very noticeable wobbling of the steering wheel given steering wheel ratios in common rack and pinion systems. To account for this, we plan to explore a few other gain combinations as alternatives if our controller is too jittery on Niki.