**Group Too**
ME 227 - Vehicle Dynamics and Control
Spring 2021
Omar Elsewify, Dylan Lu, Nathan Poczekay, Andrew Zerbe

# Vehicle Control Project

**13th May 2021**

## Executive Summary

We implement two different controllers for guiding a VW Golf GTI two laps around a symmetrical oval track that meet the design goals listed below. We also meet additional specifications that limit maximum accelerations and unnatural steering commands. The first control method is a feedforward and feedback lookahead controller that achieves a maximum lateral error of 0.19 m. The second controller is a PID controller that provides lateral tracking within 0.05 m after settling. A discussion and comparison of the development and performance of the controllers is elaborated on below.

## Design Goals

1. Controller will produce no more than **20 cm** of lateral error a the center of gravity
2. Capable of tracking of the speed profile within **0.75 m/s**

## Description of Controllers

### Lookahead Controller

We begin with the lookahead controller framework derived in class. To calculate the steering angle, we establish a specific lookahead gain, $K_{LA}$, and lookahead distance, $x_{LA}$, that function as multipliers for the lateral error, $e$, and heading error, $\Delta\Psi$ (Figure 1). With this control input, we substitute into our state-space representation for the system (Figure 2). Using this representation, we can easily extract the poles using the eigenvalues of the state-space matrix to select an appropriate lookahead gain.

$$\Delta\psi_{ss} = \kappa\left(\frac{maU_x^2}{LC_{ar}} - b\right)$$

$$\delta_{ff} = \frac{K_{la}x_{la}}{C_{af}}\Delta\psi_{ss} + \kappa(L + KU_x^2)$$

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

**Figure 1:** Lookahead controller for steering angle

$$\frac{d}{dt}\begin{bmatrix} e \\ \dot{e} \\ \Delta\Psi \\ \Delta\dot{\Psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{K_{la}}{m} & -\frac{(C_{\alpha f}+C_{\alpha r})}{mU_x} & \frac{(C_{\alpha f}+C_{\alpha r})}{m} - \frac{K_{la}x_{la}}{m} & \frac{(-aC_{\alpha f}+bC_{\alpha r})}{mU_x} \\ 0 & 0 & 0 & 1 \\ -\frac{K_{la}a}{I_z} & \frac{(bC_{\alpha r}-aC_{\alpha f})}{I_zU_x} & \frac{(aC_{\alpha f}-bC_{\alpha r})}{I_z} - \frac{K_{la}ax_{la}}{I_z} & -\frac{\left(a^2C_{\alpha f}+b^2C_{\alpha r}\right)}{I_zU_x} \end{bmatrix}\begin{bmatrix} e \\ \dot{e} \\ \Delta\Psi \\ \Delta\dot{\Psi} \end{bmatrix}$$

**Figure 2:** State-space representation of lookahead controller

We selected and sampled a range of lookahead gain values that were appropriate from the root locus plot of the system. Holding the lookahead distance and longitudinal speed values constant, then plotting the root locus while varying the lookahead gain, we chose to tune with values with high natural frequency and damping. We determined this to be in the range of 3000-4000 N/m (Figure 3). For the lookahead distance, we sampled from a range of reasonable values (10-20 m).

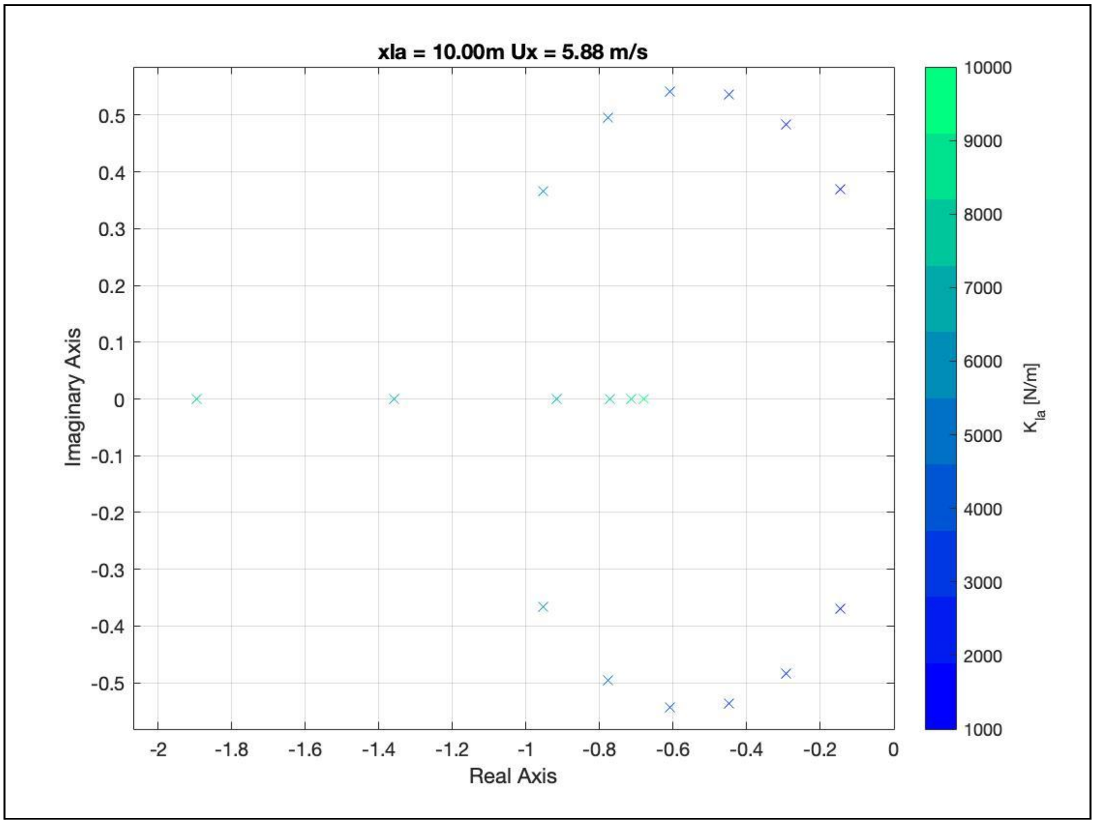



**Figure 3: Root locus plot while varying lookahead gain**

For longitudinal control, we used the feed-forward and feedback equation derived in class (Figure 4). The longitudinal gain was chosen to not exceed an “excessive” amount of longitudinal acceleration from a difference in desired and actual speeds, which was given to be below 0.25g with a 1 m/s difference in speeds. Realistically, much smaller values could be used to still track the desired speed profile quite accurately.

$$\boxed{\begin{array}{l} F_x = ma_{des} + F_{rr} + F_d + F_{fb} \\ F_{rr} = f_{rr}mg \\ F_d = \frac{1}{2}\rho C_D A U_x^2 \\ F_{fb} = K_{long}(Ux_{des} - Ux) \end{array}}$$

**Figure 4:** Feedforward and feedback longitudinal control equations

## PID Controller

We compared the lookahead controller to a PID controller, a control systems standard. PID controllers take the form:

$$U\left(t\right) = K_p e\left(t\right) + K_i \int e\left(t\right) dt + K_d \frac{de}{dt}$$

In our controller, the e(t) term is given, and the integral of e(t) is a sum to which e is added every loop (5ms). Integral anti-windup was controlled in two redundant ways to eliminate any possibility of unpredictable behavior. Specifically, the overall output from the integral term (including its gain) was limited to less than 2 degrees of steer angle, and the integral term does not begin summing until the car's forward velocity is above 1 m/s. The time derivative of e is represented by an equation derived in class:

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

It is worth noting here that while the lookahead distance in the lookahead controller has a very similar function, as a proxy for future error, this formulation of error rate is physically accurate. Gains were selected with trial and error and refined using a root-locus plot.

In order to generate a root locus plot, we had to substitute the PID controller into the state space representation of our vehicle model (Figure 5). Additionally, we had to incorporate an extra element with the integral term of the error. After substituting, we could reform the state space representation into the desirable "A" matrix, where we could extract the pole locations (Figure 6).

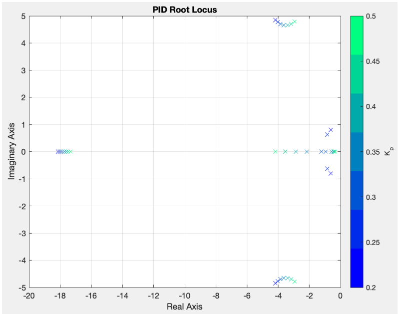
$$\frac{d}{dt}\begin{bmatrix} e \\ \dot{e} \\ \Delta\Psi \\ \Delta\dot{\Psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{(C_{\alpha f}+C_{\alpha r})}{mU_x} & \frac{(C_{\alpha f}+C_{\alpha r})}{m} & \frac{(bC_{\alpha r}-aC_{\alpha f})}{mU_x} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{(bC_{\alpha r}-aC_{\alpha f})}{I_z U_x} & \frac{(aC_{\alpha f}-bC_{\alpha r})}{I_z} & -\frac{(a^2C_{\alpha f}+b^2C_{\alpha r})}{I_z U_x} \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \\ \Delta\Psi \\ \Delta\dot{\Psi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{C_{\alpha f}}{m} \\ 0 \\ \frac{aC_{\alpha f}}{I_z} \end{bmatrix} \delta$$

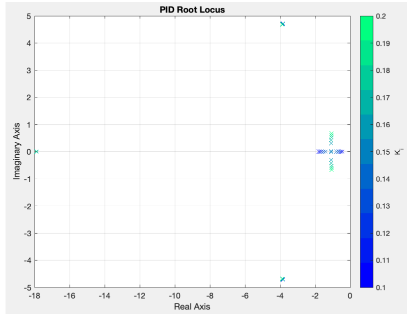
**Figure 5:** Standard state-space representation of our system

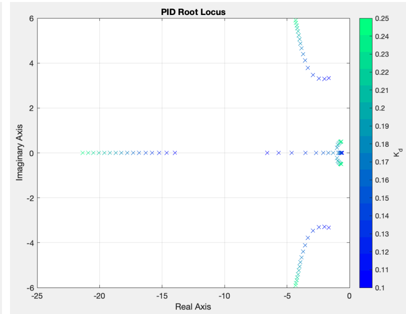
$$\frac{d}{dt}\begin{bmatrix} e \\ \dot{e} \\ \Delta\Psi \\ \Delta\dot{\Psi} \\ \int edt \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\frac{K_pC_{\alpha f}}{m} & -\frac{(C_{\alpha f}+C_{\alpha r})}{mU_x} - \frac{K_dC_{\alpha f}}{m} & \frac{(C_{\alpha f}+C_{\alpha r})}{m} & \frac{(bC_{\alpha r}-aC_{\alpha f})}{mU_x} & -\frac{K_iC_{\alpha f}}{m} \\ 0 & 0 & 0 & 1 & 0 \\ -\frac{aK_pC_{\alpha f}}{Iz} & \frac{(bC_{\alpha r}-aC_{\alpha f})}{I_zU_x} - \frac{aK_dC_{\alpha f}}{I_z} & \frac{(aC_{\alpha f}-bC_{\alpha r})}{I_z} & -\frac{(a^2C_{\alpha f}+b^2C_{\alpha r})}{I_zU_x} & -\frac{aK_iC_{\alpha f}}{Iz} \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \\ \Delta\Psi \\ \Delta\dot{\Psi} \\ \int edt \end{bmatrix}$$

**Figure 6:** State space representation with PID control

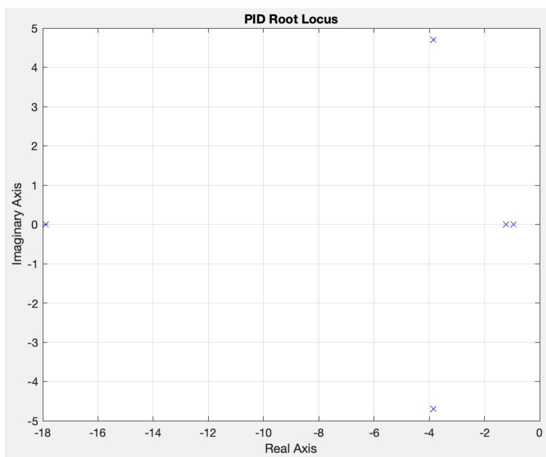
First, we tuned PID without feed-forward to perform well on straightaways, starting by increasing P and then damping with D, then increasing both of them until performance was good, and adding I to improve performance on the initial lateral error. Then, feed-forward was added, using the same equation as with the lookahead controller but without the steady-state delta-psi term (under the assumption that the PID loop would avoid a similar steady-state error). Then, keeping the other two gains at their previously selected values, a root-locus was plotted for Kp, Ki, and Kd around the selected values:








Kp was kept at its initial value, as it seemed to work well. Then, Kd was increased from its original value until it was close to the threshold where two poles would split from the real axis; in other words, these poles were selected to be critically damped (plus some safety margin that makes them slightly overdamped). This avoids oscillatory behavior, especially when delay and noise are introduced. Ki was kept at its original value as well. Increasing Kd and keeping Ki the same made sense because in our trial and error, higher Kd values had slightly better performance on curvature, but higher Ki values did not make much of a difference. Keeping poles within the LHP was a mostly trivial task, since the gains selected with trial-and-error were already stable. The final values correspond to the poles below:




We think that these poles are a good balance of responsiveness and damping and minimize tendencies for oscillation as much as possible. Trial and error, and the subjectivity of prioritizing different criteria, mean that we cannot claim objectively that our gains are perfectly tuned; however, the results are reliable and satisfactory.

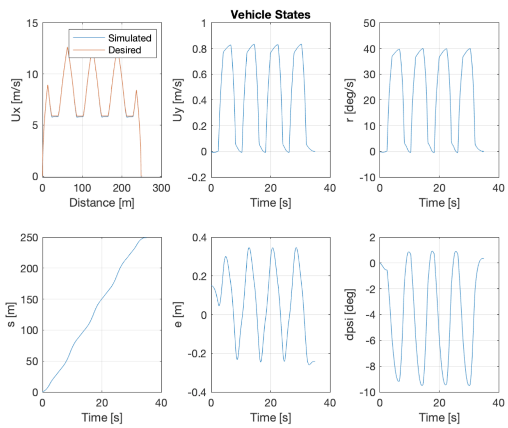
## Comparison of various simulations

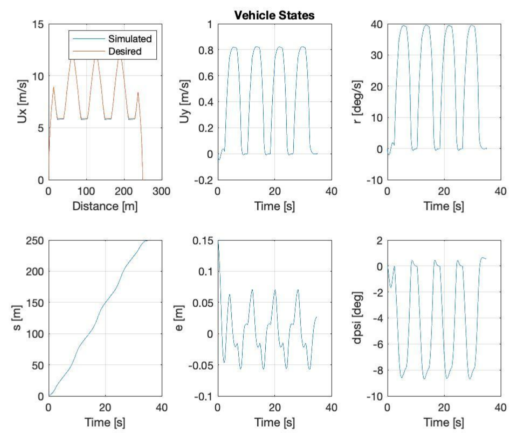
### Simulation results

We deployed 4 different simulation modes to show the effects of increasing complexity on the results of the PID and Lookahead controllers:

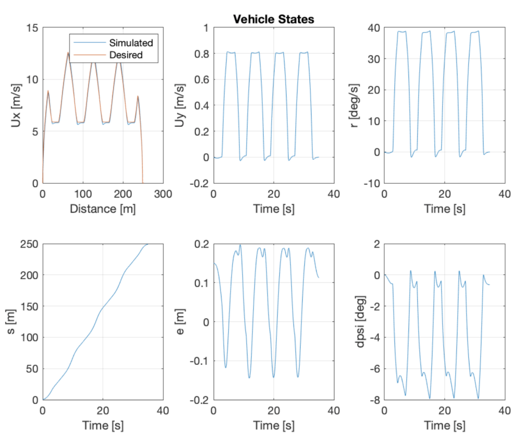
**Lookahead on the left.** **PID on the right.**

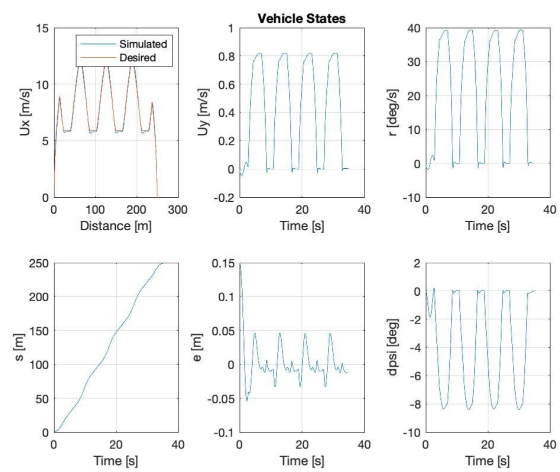
Nonlinear Results (sim_mode = 0).



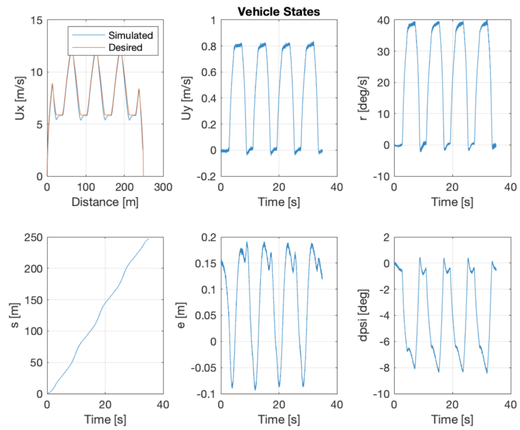



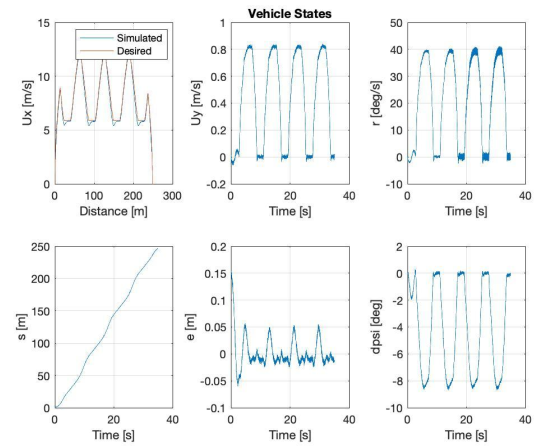

Actuator Dynamics (sim mode =  1)



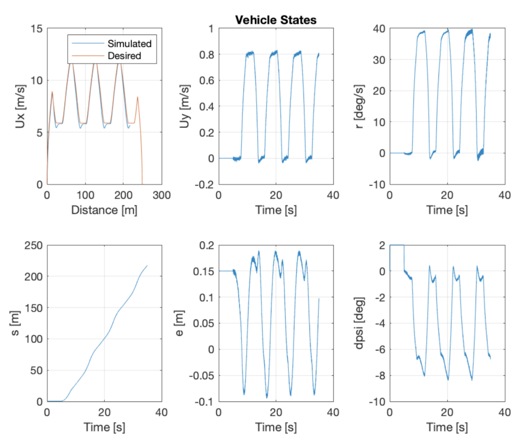



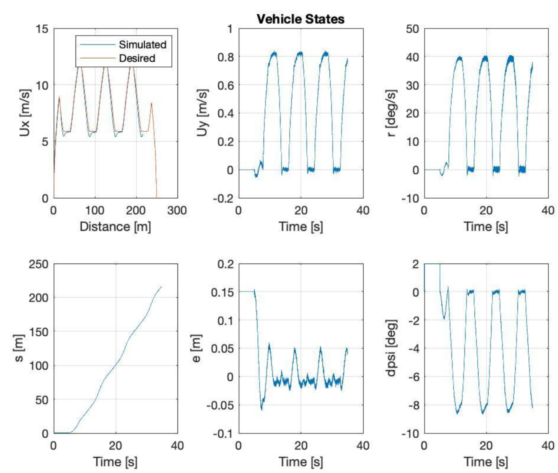

Actuator Dynamics and Noise (sim mode = 2)




## Actuator Dynamics, Noise, and Hold (sim mode = 3)




The results that most accurately describe the real life vehicle would be those from sim_mode 2 and 3, that show some noise added to the system. Note the identical performance of the PID controller when the hold period is added, due to anti-windup protection.

## Impacts on our design

With the lookahead controller we found that setting gains in the simplest model (sim_mode 0) had a very poor translation to the more complicated models. When tuning our lookahead design we started with estimates of the gains from equations and then developed root locus plots showing the effects of $x_{LA}$ and $K_{LA}$ on the system to narrow down our range. We took a similar approach to this when finding the gains of the PID controller.

More complicated simulations certainly introduced complications in PID control. In particular, noise tended to excite natural frequencies in controllers with high gains that were stable on paper, and with actuator delay. The solution was typically to reduce overall gains, especially the proportional gain, pushing poles to the left (as we can see in the root-locus plots).

## Discussion

### Tracking performance

The PID controller performs quite a bit better in lateral error than the lookahead controller. When looking at sim_mode 3 results for both, after the initial system response (after about 10s), the PID lateral error stays within 0.05m to -0.03m, compared to 0.2m to -0.05m with the lookahead.

This is mostly due to the formulation of de/dt; even without an integral component, the PID loop outperforms the lookahead to a very similar extent. The I term serves to buffer out steady-state error, and with the curves already taken care of by the feed-forward term, it has very little function. At least, in the world of simulation; in the real world, it could compensate for road tilt or wind or other similar factors.

In terms of achieving the goal of staying within 0.75m/s of the target longitudinal velocity, both controllers do this very well.

### Robustness of design

PID is not without its disadvantages. It seems more susceptible to noise than the lookahead controller, even with low gains; noise gets amplified to about ±0.2° of steering angle, which is very small, but larger than the lookahead. The steering curve is generally less smooth. This makes sense, because the poles are quite a bit further down on the LHP in the PID controller compared to the lookahead, corresponding to a higher Wn and a faster, potentially more erratic response. Future work could address methods to limit the response to noise.