Professor Fearing EECS192/Assignment #3 Learning Control v 1.00 Spring 2020

**Due by 330 pm Tues. April 28 on BCourses in .pdf.**

For this assignment you will be using `V-rep` for dynamic simulation of a car, similar to HW#2, except now learning will be used to improve car performance after an initial learning run. The track is the same as for round 1, except the car has increased torque so that it is capable of 12 m/s in straight line, and a set of cones have been added to the track. The steering slew rate limit is 60 degrees in 160 ms. The car is green to distinguish it from the previous car. (This car model will be used for round 2.):
`https://github.com/ucb-ee192/simulator-pub/blob/master/tracks/cory-track-fastcar.ttt`
You should not modify the car interface:
`https://github.com/ucb-ee192/simulator-pub/blob/master/carInterface.py`
The car control code will need to be modified, both to put in a good line tracker, and to add learning control:
`https://github.com/ucb-ee192/simulator-pub/blob/master/controller.py`

As mentioned in lecture and discussion, it is not too difficult to build a crude map of the track from a slow run, and then use that information to make subsequent runs faster. For example, the track can be segmented by crossings or changes in curvature while using odometry to roughly note track position as shown in the figure below.

For this assignment, you will use learned track information for either (1) steering control or (2) velocity control, your choice. (You do not need to use both, but are free to do so for Round 2.) The goal of this assignment is to see if you can use learning control to get a significant speed improvement (say 10%) over a decent standard PID control (as used in HW#2.) You are enouraged to use your own mapping and feedforward strategy, but here are some examples of each to get started.

*Steering Control.* For feedforward steering (referring to Lec 6 slide 24), a reference signal $r(s)$ is fed into the steering controller, where $s$ is path length from odometry. Learning can be used to compensate for the step between 17 and 18. 1 meter before the step, the reference can be set to 15 cm to the left of the track. At the step, the reference is set to zero. In more advanced versions, curves might be cut (but watch out for cones).

*Velocity Control.* For velocity control, knowing that segments 27 and 28 are straight, the velocity can be set much higher than for the curves. From odometry (not time!), the speed can be reduced before the curve segments.

(25 pts) 1. Base line (memorization run)
(20) a. Using your favorite control method (e.g. PID) find a reasonable fixed speed that allows the car to smoothly complete the track with small error and without hitting any cone(s). Report $k_p$, $k_d$, and $V$ or describe control strategy used. (This is your *reference run*). Plot on 1 page: actual $x$ vs $y$ position of car, waterfall plot, lateral error $y_a$ as function of time, steering angle $\delta$ as function of time, and odometry from integrated wheel velocity.
(3) b. Specify worst-case overshoot (cm), and note on plots where this occurs.
(2) c. Note on plots where wheel slip is occuring (a difference between wheel velocity and body velocity).

(25 pts) 2. High speed run without memorization
repeat question 1abc, but try to maximize speed without hitting cones. This is your *control run.*

(25 pts). 3. Control with learned track
Using either learned *Steering Control* or *Velocity Control*, run the simulation, and plot on 1 page: actual $x$ vs $y$ position of car, waterfall plot, lateral error $y_a$ as function of time, steering angle $\delta$ as function of time, and odometry from integrated wheel velocity. This is your *learning enhanced run.* You should try to get a 10% time improvement over your *control run.*

(25 pts) 4. Algorithm Description and Comparison
(5). a. Describe the learning algorithm used.
(15). b. Using plots of steering and velocity vs odometry for *control run* and *learning enhanced run*, explain specifically where the learned track inputs improved performance.
(5). c. Upload python code used to bcourses along with single pdf of all plots.