Final Report ECE346

Claira Fucetola, Samuel Kleiner, Vani Pandian, and Lachlan Toovey

Professors: Jaime Fisac

Princeton University

May 2024

1 Task 1: Obstacle Course Racing

1.1 Problem Statement

Task 1, Obstacle Course Racing, is designed to test the ability of a truck to autonomously drive through a series of waypoints on a roadmap while avoiding static obstacles and maximizing speed. This task aims to use onboard perception to detect static obstacles. This is performed to add an element of variability in the environment. One of the leading problems in robotics is uncertainty and the ability to account for uncertainty. Driving an obstacle course with obstacles that are not previously known and must be detected on the road adds multiple levels of uncertainty to the basic problem of autonomously navigating a trajectory, including access to the location of the obstacles on the track, reliability in the detection of the obstacles, and robustness in avoiding them.

As per the assignment specifications, the truck and the algorithm needs to account for the possibilities of up to 20 static obstacles in various positions, emphasizing the importance for the truck to be fine tuned, so that it may account for sudden swerving and stopping. Additionally, the truck needs to have an appropriate response to boundary lines, as even in avoiding obstacles it should not cross over solid lines or drive on the wrong side of the road.

Implementation of this task requires application of the techniques learned in class and in earlier labs to real-world simulations that an engineer configuring autonomous vehicles would have to consider. An in-depth understanding of the drawbacks and benefits of ILQR was required for implementation, along with innovative solutions to supplement ILQR when it failed to produce reliable results across trials. Therefore, creative ways of smoothing our reference path and trajectories for scenarios not expected or encountered in previous labs were imperative to reliable successful runs on the track.

While this task can seem like a straightforward implementation of ILQR and an extension of previous labs by solely redefining the reference points, this task is more involved and fails for such naive solutions. This is because although fine tuning ILQR can lead to acceptable results, uncertainty is

introduced when navigating around obstacles. The ILQR may attempt to avoid an obstacle and in doing so run into a boundary or drive on the wrong side in a lane, creating dangerous situations when driving autonomously on a road. Tuning the parameters is helpful, however it is ideal to provide more guidance to the truck when making a decision on how to avoid an obstacle to ensure the truck behaves as expected and follows the rules of the road. Ultimately, ILQR is effective when fine tuned to a smooth reference path that avoids obstacles, through lane changing and smoothing techniques.

1.2 Implementation

a. Stage One

Stage One consisted of designing waypoints and adding goals to the existing ILQR implementation, checking that the truck could follow the race course waypoints well *without* the presence of static obstacles. This proved successful, with both the simulation and the physical truck following the trajectory fairly well (with the exception for the weak points that exist for all the trucks in the bottom right and top right corner, previously discussed with the TAs). While using the ILQR code implemented for Lab 1, there were additional inconsistencies along the one-lane slow top portion of the track. Therefore, the ILQR function was switched to use ILQR_jax, provided in the “good ILQR” , and an increase in consistency of and reliability in pinch points of the track was observed.

b. Stage Two: With Obstacles

Stage Two consisted of adding in a few static obstacles at a time and adjusting the weights of the lane boundary cost and the obstacle cost to get the truck to swerve around the static obstacles *without* changes to the algorithm or the reference path. Stage Two was performed solely on simulation to observe the behavior of the ILQR when avoiding static obstacles on its own. Although we altered the values many times, the truck continued to swerve outside of straight lanes, or make turns that were too wide to maintain the goal trajectory, proving that significant changes to the reference path needed to be made.

c. Stage Three: With Obstacles

Stage Three focused on making alterations to the algorithm so that the reference path would be updated to avoid obstacles. This was implemented by keeping an updated dictionary of static obstacles, detecting when points in the reference path intersected an obstacle, and saving these locations. For each location where the reference path intersected the obstacle, the lanelet map and lanelet class were used to detect which lane the truck was currently in and update the x,y position of the truck to be in the center of the neighboring lane. However, this technique produced sharp changes, which required abrupt turns close to the obstacle. Due to the cost of collision and a path consisting of actions infeasible for the truck, this ended up resulting in either collisions with the static obstacle that broke the trajectory, or large and unpredictable swerves out of the track as the ILQR attempted to keep up with the reference path but was limited by the trucks dynamics. Example reference path shown in **Figure 1** in the Appendix.

d. Stage Four: With Obstacles

Therefore, Stage Four was decided to implement a low pass Gaussian convolution filter on the reference path to smooth out the curves. Initially, we intended to write out and test the Gaussian filter ourselves, but our method made the reference path go off the track at points as well as intersect itself in loops. After some input, we imported the scipy.ndimage.gaussian_filter1d and applied this function to the current x,y positions along the path message poses to retrieve updated x,y positions representative of the smoothed curve. Finally, sigma was tuned to change the smoothness of the curves and find the best combination of path around the obstacles and ILQR obstacle costs (**Figure 2**, Appendix).

1.3 Performance

When testing on Stage Four of the implementation, high performance was achieved on the simulation and reasonably high performance was achieved when running directly on the truck. In simulation, it was evident that the more crowded the lanes, by adding more obstacles, the more difficult the task as the reference path became very curvy. On simulation, one of the unexpected hurdles encountered due to the curvy path was figuring out how to limit lateral drifting around turns. It was difficult to tell if this was

something encountered due to lag on the computer or due to a need to more closely fine tune an ILQR parameter. This drifting resulted in unpredicted collisions with obstacles as the simulation deviated from the reference path. However, an unexpected strength on the truck was the absence of this drifting that occurred in the simulation. The truck was able to navigate around obstacles quite consistently without dramatically falling off the reference trajectory.

In terms of reliably staying within the boundaries of the track, the simulation performed significantly better than the truck. The truck displayed difficulty around sharp turns (especially in the inner loop of the track) and would often cut off a portion of the turn, therefore going over the solid line. Additional tuning of the ILQR costs on the truck were performed and did lead to increased performance, but deviations are still present at times. Additionally, an unexpected hurdle was found with some detection failures of obstacles when creating the reference path, or the location of the obstacle such that it blocks the localization of the truck. The truck's path for two successful runs shown in **Figure 3**, Appendix.

1.4 Takeaways

Practical challenges remain with observing static obstacles when going around a bend for the first time. When calculating the given path between the current waypoint and the next waypoint to account for obstacles, if the truck cannot see an obstacle at the current way point, the truck has a high probability of not calculating the reference path around the obstacle. In this situation the truck relies on ILQR to avoid the obstacle. One place this occurs is if obstacles are set up between waypoints 9 and 7 or 8 on the first pass through the loop of waypoints. To account for this in the future, the reference path could be dynamically recalculated each time there is new data available. However, this would have increased computation overhead and the computer was already experiencing lag, so improved hardware might be required for this to be implemented.

2 Task 2: Autonomous Driving in Traffic

2.1 Problem Statement

The goal for Task 2a, Autonomous Driving in Traffic, was to implement a robust urban autonomous driving policy designed to replicate a more realistic and dynamic environment. The truck was tasked to complete the same goalpoints as Task 1 with the additional caveat that the truck must remain in the right hand lane at all times except when passing other trucks.

This task replicates the real-world application of driving an autonomous vehicle with other vehicles, and the uncertainty that exists in any model with respect to the decisions the other vehicles may make at any time. The track in the lab most closely resembles a highway driving scenario. It is complex enough that naive approaches to creating autonomous driving policy, like behavioral cloning, fail since they do not rigorously account for all possible multi-agent interactions on the track. For example, the truck might have learned how to avoid a dynamic obstacle, and stay within the boundaries of the road, but to do both at once at any given point on the track might cause the truck to violate one of those safety critical tasks, creating a dangerous situation in a real world scenario. One naive approach is to treat a dynamic obstacle as a static obstacle and dynamically update the autonomous vehicle's reference path to move around the obstacle (i.e. directly extending task1 onto task2). This approach leads to failure as the forward reachable set of the truck and obstacles given their current state and control input or potential controls is not considered. This could result in the autonomous vehicle planning a path to pass the traffic and the traffic moving in a way to intersect the planned path, causing a collision.

2.2 Implementation

To implement this task a combination of the Forward Reachable Set (FRS) code from Lab2 and the dynamic building of trajectory through the given waypoints, adjusted to keep to the right lane, was used. Time restraints prevented us from implementing additional safety measures for a more robust approach.

We worked in a separate trajectory planner file for this task, namely trajectory planar traffic, to build the environment, but we still implemented the full reference path by checking when the truck hits a goal point (within reasonable error) and creating a new reference path from the truck's current location to the next goal point. Additionally, the waypoints were adjusted to force the truck on the right hand lane at all times to keep to the assignment specification. Finally, a forward reachable set on the dynamics obstacles was calculated, and the obstacle list was updated and sent to the ILQR to add a penalty to paths that intersect with one of the obstacles.

Since the implementation of this assignment relies heavily on the ILQR to understand and apply the necessary cost to the boundary and obstacles, to enforce that neither are hit, it was necessary that the parameters in the yaml file the ILQR was configured with were finely tuned. While this implementation accounts for all possible next states for the dynamic obstacles, which is key when interpreting uncertainty of other moving vehicles, it lacks additional reinforcement for navigating around the other dynamic obstacles (similar to what is done in task 1 with updating the reference path to explicitly avoid collisions).

2.3 Performance

As there has not been access to physical trucks for testing the solution of this task, the performance of the simulation was surprisingly high, given the limited time to implement and semi-naive approach of closely tuning the ILQR. The implementation of the Forward Reachable Set provided enough feedback to the ILQR to enforce the truck to reduce speed when tailing behind a car in a single lane and encourage the truck to pass in the left lane when available. Through the simulation runs, the truck was highly unlikely to attempt to cross solid lines when stuck behind a truck in a single lane. Additionally, the truck accounted for other dynamic obstacles getting too close to cause collision through rear-ending the truck. Given this case, that one of the dynamic obstacles would attempt to rear-end the truck, the truck would adjust its speed to remain outside the forward reachable set of the dynamic obstacles.

While this solution performed well, weakness was found in the truck's response to merging traffic. The truck prioritized avoiding collision with the dynamic obstacle and remaining outside of its forward reachable set. Therefore, if one of the dynamic obstacles unexpectedly merge into a lane we were currently in, our truck would attempt to avoid collision which, depending on location on the track, sometimes resulted in running off the track. This was the highest piece of uncertainty, as sometimes the truck would respond by speeding up to get in front of the traffic attempting to merge into our lane and other times avoid by leaving its current lane. This does seem to be a reasonable response to aggressive merging traffic in the real-world, so it is unfortunate, but not surprising.

An additional hurdle encountered was the case of "missing" a waypoint if it occurred along the route while in the process of passing a vehicle in the left lane. Since the waypoints were set to the right lane to force the truck to remain on the right side except for passing, which worked reliably, if the truck completed a waypoint in the left lane due to passing traffic, then the truck would pass the "range" of the way point and cease to plan a new reference path to the next waypoint. However, this hurdle was able to be accounted for by increasing the error around completing a way point to accommodate when the truck is in the left lane. Based on the simulation runs, this solution proved successful.

See Appendix **Figure 4** for three cases, from different trials, where the simulation was able to successfully avoid collision by either slowing down or passing in the left lane.

2.4 Takeaways

Remaining practical challenges are found in the unexpected hurdles that were not possible to accommodate given the time restraint. To solve this problem a reference path that accounted for the dynamic obstacles, such as in Task 1, could have been implemented. However, this approach would need to be tweaked such that at every time of replan, the reference path was updated considering the new locations of the dynamic obstacles, rather than having the reference path calculated once for every goal

point. This would look like a dynamic reference path that shifts and curves with the dynamic obstacles and their forward reachable set as they move.

This paper is a representation of our own work in accordance with university regulations.

/s/ Claira Fucetola

/s/ Vani Pandian

/s/ Lachlan Toovey

/s/ Samuel Kleiner

3 Appendix

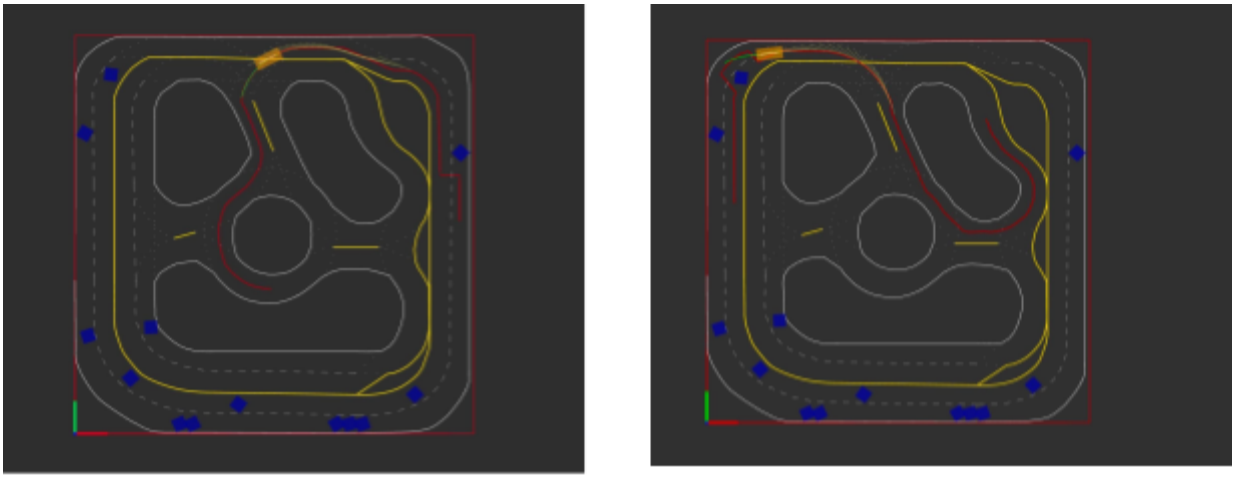


Figure 1: First Implementation of adjusting for obstacles in the reference path displaying the abrupt lane changes and sharp turns.

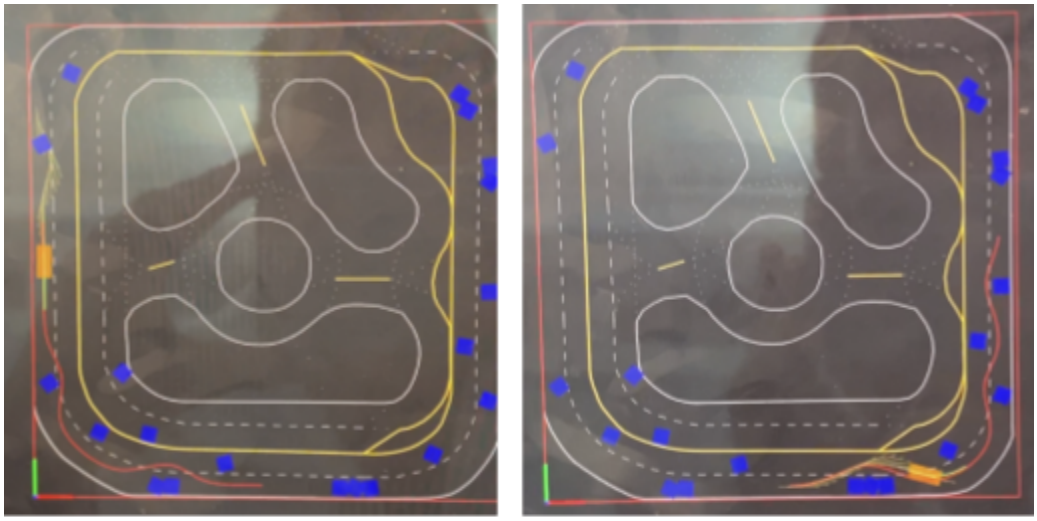


Figure 2: Adjusting for obstacles in the reference path after applying the gaussian filter to smooth the path.

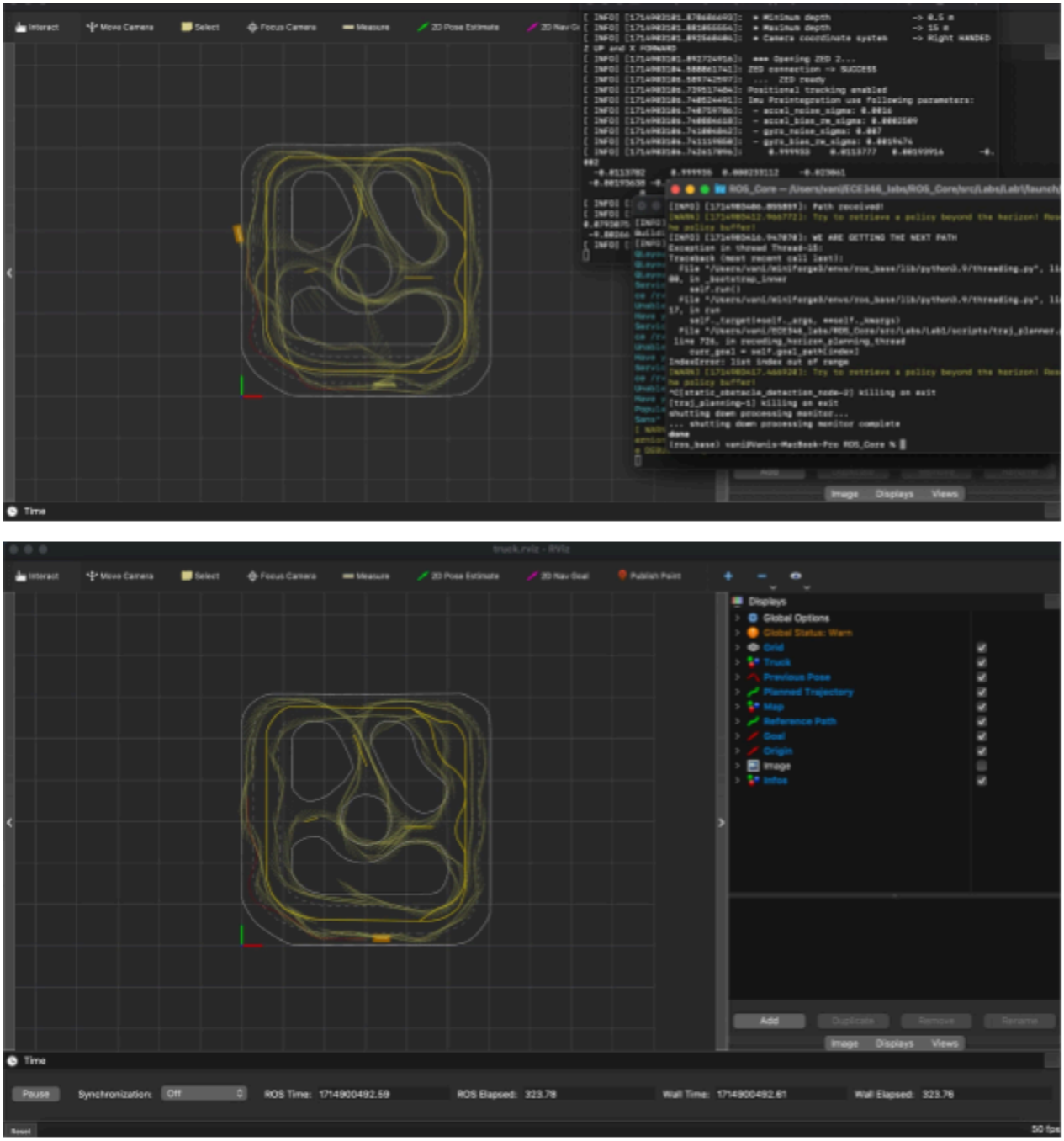


Figure 3: Trajectory of Truck after completing a full pass over the waypoints and updating the referencing path the account for obstacle avoidance..

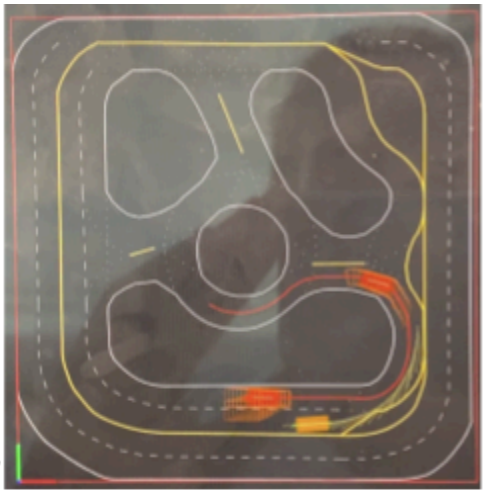
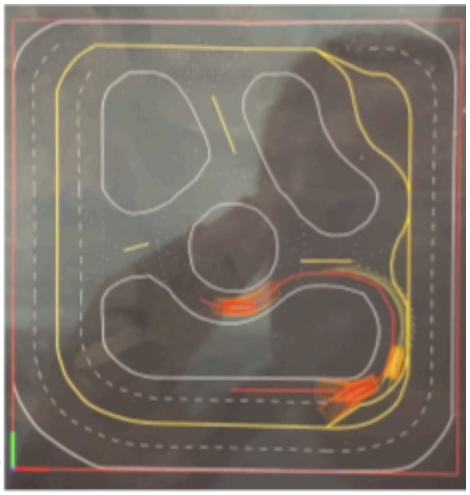
a.  b.  c. 

Figure 4: Task 2a Traffic Simulation, **(a)** avoiding the dynamic obstacles by slowing down when passing is unavailable due to the forward reachable set **(b)** using the left lane to pass an obstacle when available **(c)** slowing down when passing is unavailable due to being in a single lane.