AA203 Final Project:
Re-implementing *Real-Time Trajectory Replanning for Autonomous Driving*

David Gonzalez
Stanford University
Mechanical Engineering
dxg2@stanford.edu

Nefeli Ioannou
Stanford University
Mechanical Engineering
nefeli@stanford.edu

Aubrey Kingston
Stanford University
Aeronautics & Astronautics
aubreyk1@stanford.edu

June 12th, 2020

1 Introduction

Trajectory planning for vehicles at the limit of friction is a problem of particular interest to autonomous vehicle design. Being able to quickly plan a new trajectory in an emergency scenario has the potential to save lives. For our project, we chose to re-implement the algorithm used in "Real-Time Trajectory Replanning for Autonomous Driving"[SG19]. In this paper, the authors outline an algorithm that allows a racing vehicle to avoid an obstacle while the vehicle is at the limits of its handling abilities. A key appeal of this paper was that it allowed our group to reuse the low-level control architecture and vehicle dynamics simulator we had previously built for Professor Chris Gerdes' Vehicle Dynamics and Controls class (ME227) and incorporate the tools we learned in AA203. This allowed us to solve the optimal control problem using Sequential Convex Programming (SCP), and more specifically, the CVX framework in Matlab. Our preliminary results show our adapted algorithm is able to replan a trajectory around an obstacle which is then tracked using our low level controller. In testing, the replanning stage is very sensitive to the initial conditions and can fail to generate a safe trajectory under some constraint scenarios.

2 Summary of Baseline Paper

The authors cast the optimal replanning problem as a quadratically constrained quadratic program (QCQP) [SG19]. The motivation for this is that non-linear optimization problems that have been used in the past for trajectory replanning are slow. This is particularly undesirable when used for online trajectory replanning for autonomous vehicles.

The authors in this paper assume that we have a nominal trajectory (generated by other means) but, for some reason or another, is no longer safe of physically possible, for example, when weather conditions leave a patch of road slippery or when an obstacle obscures the original nominal path. The authors use a simplified dynamics model, modeling the vehicle as a point mass particle with acceleration limits, in order to calculate a new trajectory. This new trajectory is then executed by a low level controller like the one seen in [KG15]. The paper claims a 10 second planning horizon can be calculated in under 20 milliseconds with their convex solver. While the point mass simplification may seem large, accurate modeling of the longitudinal weight transfer and road topography are included in the constraints. The authors demonstrate in this paper that the point mass simplification can accurately capture the vehicle dynamics paramount to trajectory generation at the limits of grip.

Their cost function takes the form:

$$\begin{aligned} \min_{z} \quad & \sum_{i=1}^{N} \left(\frac{1}{2} z_i^\top H_i z_i + f_i^\top z_i \right) \\ \text{subject to} \quad & C_{i-1} z_{i-1} + D_i z_i = c_i \\ & \underline{z}_i \le z_i \le \overline{z}_i \\ & A_i z_i \le b_i \\ & z_i^\top Q_{i,k} z_i + L_{i,k}^\top z_i \le r_{i,k} \end{aligned}$$

The z_i vector in their formulation is a combination of state, control and slack variables for the ith discretization point. These variables are deviations from a nominal trajectory [SG19]. The first constraint resolved the vehicle dynamics given by the point mass model. It also ensured that the planned trajectory starts from the vehicle's current state and returns to the nominal path at the end of the planning horizon. The second constraint on minimum and maximum values (or box constraints) for each state encodes information about road edge and attainable accelerations limits and ensures the vehicle velocity at the end of the horizon is not too great for the vehicle to then traverse the ensuing nominal path. The third constraint encodes engine power constraints and the last constraint incorporates the tire friction model.

A key contribution of this paper was shifting the independent variable from time to distance. This allows the authors to precisely encode the obstacle avoidance constraints for a stationary obstacle like the one we are testing in this problem. We see the transformation of the state matrix $x(t) = [s(t), e, V, \sigma]^\top$ to $x(s) = [t(s), e, V, \sigma]^\top$. In this scenario, setting the upper and lower limits on error e sets the outer bounds of the road, or in the case of an obstacle, the bound of the obstacle as it restricts the size of the available path. Section 3.4 outlines how the obstacle is incorporated into the path via the box constraints.

The authors of this paper give us models for the linearized dynamics matrices A and B which take into account the road topography with variables i, representing local twisting of the road surface and j, the component of the path's total curvature in the plane of the road, as well as unit vectors $\hat{p}_x$, $\hat{p}_y$, $\hat{p}_z$, which correspond to the orientation of the road at that path step.

$$A(s) = \begin{bmatrix} 0 & \frac{-k}{V} & \frac{-1}{V^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-k\Sigma F_x}{mV} - Vij & \frac{-\Sigma F_x}{mV^2} - \frac{\rho A C_d}{m} & \frac{g \cdot \hat{p}_y}{V} \\ 0 & \frac{-k(a_y + g \cdot \hat{p}_y)}{V^2} + i^2 & \frac{-2(a_y + g \cdot \hat{p}_y)}{V^3} & -\frac{g \cdot \hat{p}_x}{V^2} \end{bmatrix} \qquad B(s) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{V} & 0 \\ 0 & \frac{1}{V^2} \end{bmatrix} \tag{1}$$

Finally, the authors validate their algorithm using an Audi TTS autonomous test vehicle at Thunderhill West Raceway. Prior to the beginning of their testing, the exact location of the obstacle is given to the vehicle to avoid potential vision errors while testing. Testing reveals their vehicle operates at the friction limits while tracking an minimum time optimal trajectory around the obstacle.

3 Our Approach

3.1 Nominal Controller Incorporation

The low level control of the vehicle is handled by an LQR controller for lateral control and a simple feedforward controller for longitudinal control. The longitudinal controller we implemented took into account rolling resistance, drag, and grade of the road. Therefore, the longitudinal control took the following form: $F_x = m\,\alpha_{x,desired} + F_{rr} + F_{drag} + K_{long}(U_{x,des} - U_x) + F_{grade}$, where K_{long} is the chosen longitudinal gain. The lateral control is based upon the non-linear bicycle model with states $x := [e, \psi, \dot{e}, \dot{\psi}]^\top$. Here, we see that e is the lateral error as measured from the nominal path, ψ is the heading error as measured from the nominal path, and $\dot{e}$ and $\dot{\psi}$ are their respective derivatives. The two control inputs to the bicycle model are steering angle and longitudinal force (from the engine). The steering angle δ is therefore calculated from $\delta = Lx$ where L is the time invariant LQR gains and x is the current lateral state.

3.2 Choice of Optimal Control Method - Sequential Convex Programming

After considering a few alternatives, we selected Sequential Convex Programming (SCP) as our control optimization method to minimize cost under a slightly simplified version (see Section 3.3) of the constraints outlined above. We chose this approach because of its relative simplicity and its ability to efficiently handle constraints on state and control variables. In our implementation of SCP, we used the Matlab CVX framework. We repeated the SCP optimization problem under our simplified set of dynamics and box constraints, until convergence of our state and control vectors. In most cases, convergence was achieved within 2 to 5 iterations and the cost was minimized to ≈ 70. We believe that this could be performed with fewer passes had we had a more accurate initialization of the nominal control for these dynamics. Since we do not, we initialize to a random series of controls and run SCP multiple times to converge to the optimal values.

3.3 Assumptions and Simplification for Replanning

As was described in Section 2, the reference paper allowed for a detailed road topography. For our implementation, we chose to treat this as a 2D problem and eliminate the provisions for road orientation (off camber turns, hills, etc.). With this simplification, our point mass dynamics matrices change to those presented below.

$$A^*(s) = \begin{bmatrix} 0 & \frac{-k}{V} & \frac{-1}{V^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-k\Sigma F_x}{mV} & \frac{-\Sigma F_x}{mV^2} - \frac{\rho A C_d}{m} & 0 \\ 0 & \frac{-k a_y}{V^2} & \frac{-2a_y}{V^3} & 0 \end{bmatrix} \qquad B^*(s) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{V} & 0 \\ 0 & \frac{1}{V^2} \end{bmatrix} \tag{2}$$

We made this decision in part because we had previously crafted a nominal planar path. Creating a new path in three dimensions from scratch is outside of the scope of this class and we would not be able to easily ensure feasibility of a nominal path.

In our implementation of the objective function, we simplify the constraints. We include the constraints for the vehicle dynamics which are modelled off of a point mass, but roll the friction constraints and vehicle power constraints into the box constraint on the combined state z. This allowed us to craft z_{min} and z_{max} constraints and encode the presence of an obstacle into the path of the vehicle.

Another major change we made was modifying the combined weight matrix H. In the paper,a modified Hessian was used that was not tractable for our CVX implementation as it was not in a symmetric positive definite form as is required by our solver. We instead formed this weight matrix based on LQR theory. More specifically, we utilized Bryson's rule and set the weights of our cost matrix by following the $\frac{1}{(\text{maximum error})^2}$ convention. We decided upon our maximum allowable errors for all state variables based on literature and design considerations for an average urban road. We decided on our maximum F_x and F_y controls based on the amount of available engine power of the test vehicle used by our baseline paper and the available friction parameters corresponding to a flat road. This procedure for H formation yielded favorable results compared to performing the formation of the H matrix using a symmetric eigenvalue decomposition as alluded to in the paper.

3.4 Collision Avoidance

A key criterion for the robustness of our algorithm was its performance when tasked with an abrupt change in its nominal path. In order to create an obstacle along the path, we modified the maximum allowable error value to be reduced by the width of the obstacle, while keeping e_{min} the same. This corresponded to reducing the overall available width of the road. Furthermore, the placement of the obstacle was such that the vehicle was forced to deviate from its nominal path, since the obstacle width was larger than one-half of the road width (*obstacle width* $> e_{max}$), and the obstacle extended from the edge of the road. This placement guaranteed that the vehicle could only circumnavigate it from one direction, i.e. either the left or the right, depending on the obstacle's location. For our implementation, we placed both a "minimum obstacle" and a "maximum obstacle" in the replan episode: the minimum obstacle extends from the minimum lateral error boundary and the maximum obstacle extends from the maximum lateral error boundary towards the center of the path.

3.5 Modification to Nominal Path

After using CVX to find the combined z vector we then parse out the state x for each path step along the replanned trajectory. We then use this to rebuild a modified path. In order to modify the path struct so that the low level controller can follow it. We use the lateral error, heading error and are able to recalculate the path. The path structure combines the following states, discretized along a path:

$$\text{Path} = P = [s, k, \psi, E, N, V_x, a_x, a_y, V_{max}]^\top$$

Where s is the path step, k is the curvature of the path, E and N are the global East and North coordinates of the nominal path respectively, V_x is the nominal velocity, a_x and a_y are the nominal longitudinal and longitudinal accelerations respectively, and finally V_{max} is the maximum velocity for that section of the path.

4 Results

Overall, our results were promising and showed that Sequential Convex Programming is a viable algorithm for collision avoidance trajectory replanning in automated vehicles. Figure 1 depicts all the state and control variables during the course of the replanned trajectory. More specifically, it shows that the vehicle is able to circumvent both obstacles, staying within the desirable bounds for the lateral error as well the heading error and all the other state and control variables. Figure 2 is a close-up of the lateral error plot during the replan episode with visualization of the obstacles. The car itself is treated as a point mass throughout the baseline paper as well as in our model and therefore has no physical extent.

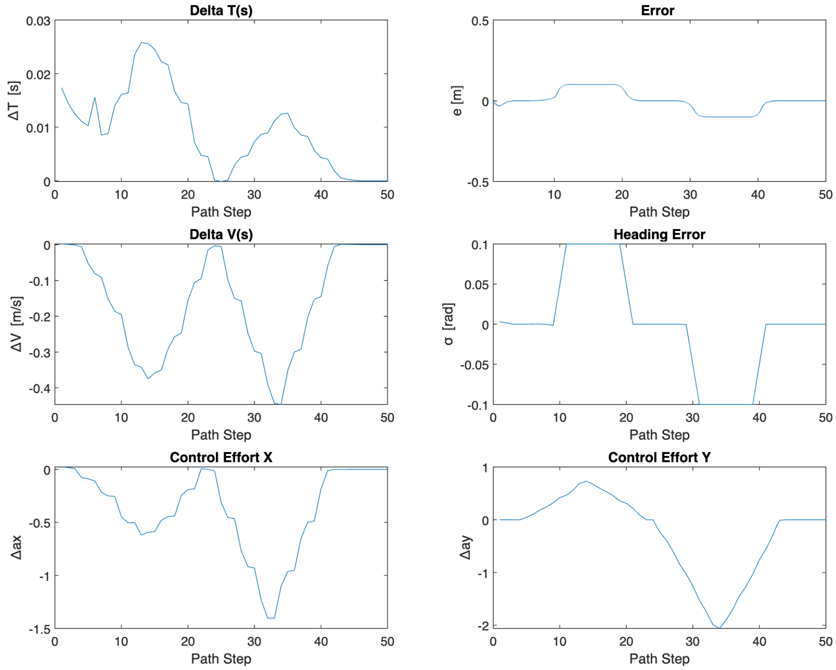

Figure 1: Trajectories of state and control over the replan episode.

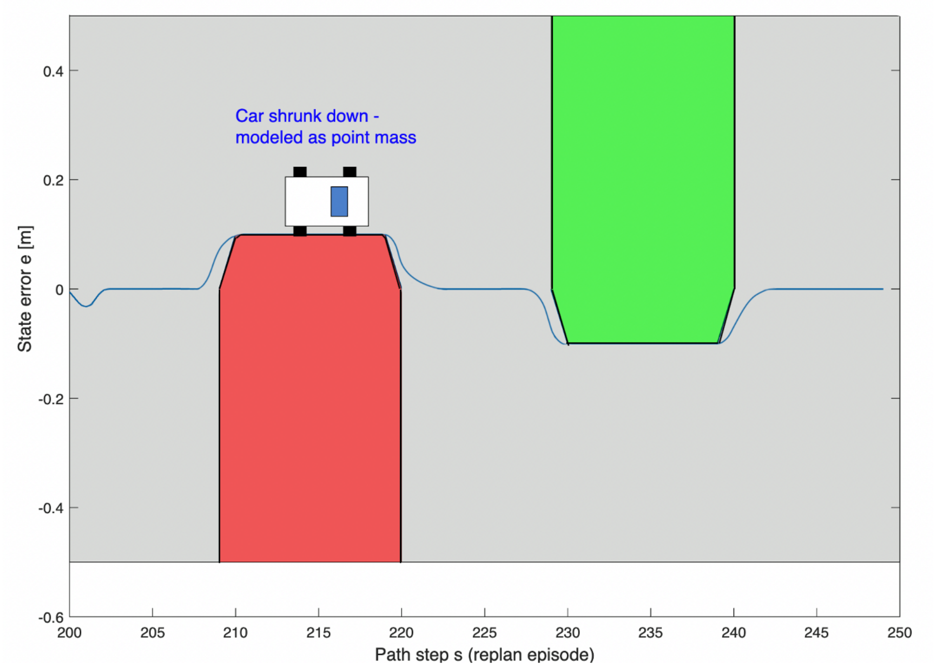

Figure 2: Close up of the replan episode. The part of the car that is closest to the nominal path follows the lateral error trajectory as the car maneuvers the obstacles

The box constraints for this particular iteration are $x_{max} = [2\text{s}, 0.5\text{m}, 2\frac{\text{m}}{\text{s}}, \frac{\pi}{4}]$, using a planning horizon of 50 steps. Additionally, the objective cost was ≈ 70.2. Although the performance of the car handles the presence of obstacles relatively well over the short replanning horizon, when assessing the robustness of our CVX replanning, we were unable to retrieve a solution when the replan episode was longer than ≈ 50 steps. We also noticed that if the end of the first obstacle was "close" (< 5 steps) to the start of the second obstacle, again, we were unable to retrieve a solution. However, this makes logical sense in that, while adhering to state and control box constraints, then it is logical that there are some scenarios that the vehicle cannot physically outmaneuver.

Figure 3 shows the replanning of a vehicle traveling around a 2-dimensional oval track. At the replanning point after rounding the first corner, the vehicle must avoid two obstacles encoded into the error constraint. After avoiding the obstacle, the vehicle recovers the nominal trajectory and is able to match the desired velocity at the end of the path We see control effort around 0.1 - 0.2 g in both the lateral and longitudinal direction which is well within the attainable limits for our test vehicle while moving at approximately 10 m/s.

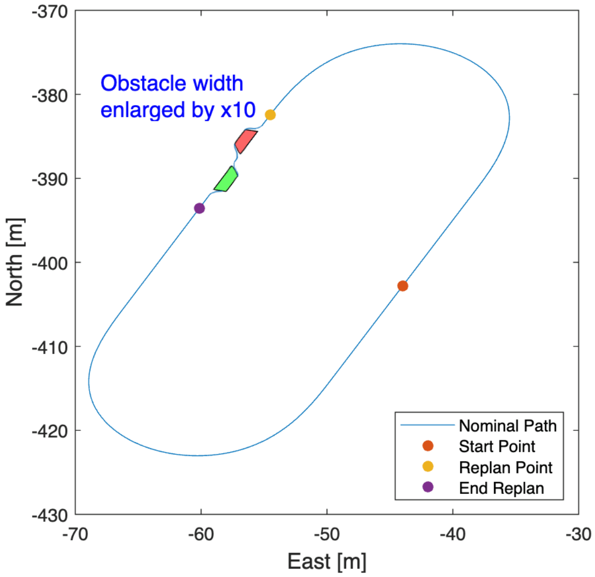

Figure 3: The oval path is the original nominal path. $s = 0$ is the starting point and is indicated by the red circle. The replan point occurs at $s = 200$.

5 Conclusions and Future Work

In conclusion, our results demonstrate that collision avoidance trajectory replanning is achievable with sequential convex programming for automated vehicles. If we had more time on this project (and with a more sane world around us), we would like to try to test different paths and longer replanning episodes. We would also be interested in looking at how noise could affect how well the algorithm is able to converge on the optimal trajectory. In our testing our initialization for the nominal control had a great affect on how fast and how reliably CVX converged on a new trajectory. It would be interesting to understand why the replan episode initialization is sensitive to where the replan point is placed and to other factors, such as the box constraints on state and control. We have linked to a drive to reference our Matlab Code: AA203 Final Project Code or https://drive.google.com/open?id=1lD_jyiSFrIk09L_Ga1S8Vzwh0cs86l_I.

Thanks to Professor Pavone and the TAs for a challenging but rewarding quarter.

References

[KG15] Nitin R. Kapania and J. Christian Gerdes. "Design of a feedback-feedforward steering controller for accurate path tracking and stability at the limits of handling". In: *Vehicle System Dynamics* 53.12 (2015), pp. 1687–1704. DOI: 10.1080/00423114.2015.1055279.

[SG19] J. K. Subosits and J. C. Gerdes. "From the Racetrack to the Road: Real-Time Trajectory Replanning for Autonomous Driving". In: *IEEE Transactions on Intelligent Vehicles* 4.2 (2019), pp. 309–320.