

CS/ECE/ISyE 524 — Introduction to Optimization — Spring 2023

Trajectory Optimization for Autonomous Vehicles

NEVINDU BATAGODA (batagoda@wis.edu)

ISHAAN MAHAJAN (imahajan@wisc.edu)

Please insert your answers in the cells below. Use only Markdown notation, not Julia code

(A) Issue being addressed in this project

This project is inspired by our work as members of Wisconsin Autonomous student organization. We are part of a university team competing in the AutoDrive Challenge II competition organized by SAE International and General Motors. The goal of this competition is to build a self-driving car capable of navigating urban environments.

As part of the control stack of the self driving car there are three planners: 1) The High Level planner which creates a set of target waypoints to follow, 2) The Mid Level planner which generates a smooth trajectory from the waypoints and 3) A Low Level planner, which uses a feedback control algorithm to generate a target velocity and heading which is then passed into the vehicle as a throttle and a steering input.

In this project we plan to address the trajectory optimization problem of the Mid Level planner of the Wisconsin Autonomous control stack. The theoretical foundation for our work will be based on the work done by University of Toronto during the previous iteration of the competition (AutoDrive Challenge I) (<https://onlinelibrary.wiley.com/doi/full/10.1002/rob.21958>).

The goal is to generate a path $P = [\mathbf{X}_1 \dots \mathbf{X}_N]$, from the way points generated from the High Level planner, $x_i = [x_i \ y_i \ \theta_i]$, where θ_i is the heading angle at way point i .

The optimal path \mathbf{P}^* , minimizes the cost of the path by solving the following optimization problem,

$$\min_{E_{i,cur}, E_{i,dev}, E_{i,obs}} \sum_{i=1}^N E_{i,cur} + E_{i,dev} + E_{i,obs}$$

$E_{i,cur} = w_{cur} \left(\frac{\Delta \theta_i}{|\Delta x_i|} \right)$ is the curvature term that minimizes the abrupt steering actions. $\Delta \theta_i$ is the change in heading at x_i , Δx_i is the change in position and w_{cur} is the weight of the curvature term.

$E_{i,dev} = w_{dev} |x_i - c_i|$ is the deviation term which ensures the final path is closer to the center of the road. c_i is the centerline point closest to x_i and w_{dev} is the weight of the deviation term.

$$E_{i,obs} = \begin{cases} 0, & |x_i - o_i| \geq d \\ w_{obs} \left(\frac{d_{obs}}{|x_i - o_i|} - 1 \right)^2, & |x_i - o_i| < d \end{cases}$$
 is the obstacle term which keeps the path away from obstacles. o_i is the distance to the obstacle closest to x_i , d is some tolerance threshold and w_{obs} is weight of the observation term.

(B) What kind of data will you need, where will it come from, and how will you obtain it?

We will need the current state of the vehicle that includes the current position, velocity, and orientation of the car, along with lane line information from the HD map. All this data will come from the Localization subteam for the competition purpose, however, for this project, we will be simulating our own input data in Julia.

(C) What is the optimization problem underlying this project? (LP, QP, SOCP, MIP, NLP, other?)

- Non-convex optimization problem

(D) What will you deliver in terms of code, results, outcomes?

- We will write a JuMP program using Julia to solve the optimization problem and then test the program using different scenarios to generate a trajectory. Some scenarios include,
 - Creating straight line trajectory.
 - Creating a smoothed curved trajectory.
 - Creating an obstacle avoidance trajectory which has a smooth lane line change.
- For each scenario we will create a plot of Cost Vs Way point position.

(E) Other points to be considered.

- N/A

In []: