Optimal Racing Line Using Trajectory Optimization

Dalton Richardson

***Abstract*— Proposal for a Controller to model and determine optimal racing line using trajectory optimization methods**

I. Introduction

In motorsports, such as Formula 1 and IndyCar, one of the most important aspects is the racing line you take around a track. This racing line determines when you should break, when you should turn, and how fast you should be going. An example of such a racing line is shown below in Fig. 1. Just being off by a few meters or milliseconds can make or break a race for a driver so it is incredibly important to know exactly what the optimal line is. For my project, I propose using trajectory optimization to determine this optimal line on any given track.

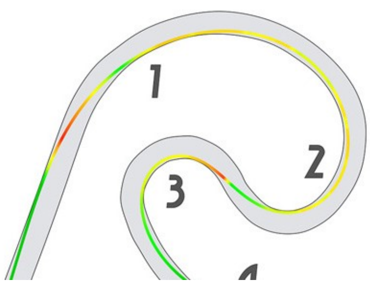

Fig. 1. Example of an Optimal Racing Line

II. Simulation

Simulating a system such as this will be the first challenge to tackle. My plan for the simulation, as well as most of the project, is to keep it simple at first and just get a working implementation. From there, as time permits, I can add onto the work and improve from there. The initial concept is to simulate a simple 2D track of constant width with a static environment with no air resistance or temperature constraints. Because of the constant width and 2D dimensions, I can model a track as a line from which I can calculate the distance from the car to determine if the car is on the track or not. Possible extensions could include making the track 3D by including inclines and such or having a changing width along different points of the track.

III. Hyper Parameters and Control Inputs

There are a few hyperparameters for this system that must be chosen before modeling can take place. The most obvious of which is the track itself, including its width. These other parameters have to do with the car dynamics themselves. This includes the max acceleration, max deceleration, turn radius, and coefficient of friction.

As for the control inputs, it will consist of a vector of (X(t), Y(t)) coordinates indicating car position along the track.

IV. The Control Task

A. Objectives and Constraints

The objective of the problem will be to minimize the time it takes to drive a lap around the track. Constraints for this would include max and min speeds and accelerations, the car not leaving the track, turn radius, and the friction keeping the car on the track. This friction and turn radius constraints will be especially important to prevent the car from just speeding through corners.

B. Design Vector

The design vector will consist of a spline of points the car will cross along the track in a format similar to: $[X_1(t) \;\; X_2(t) \ldots X_n(t) \;\; Y_1(t)\; Y_2(t) \ldots Y_n(t)]^T$. From a vector such as this we can determine the optimal path, along with the optimal speed, to take.

V. Controller Design and Solver

A. Previous Works and Inspiration

An excellent source of information for me has been a paper published by MIT on this subject of optimization of racing lines [1]. It lays out in a fair amount of detail the formulas they used and methods. While I can't obviously take exactly from their work, it is an excellent starting point from which I can start the design of my own controller.

B. Proposed Design

My proposed controller would take use of the Multiple Shooting and Direct Collocation Trajectory Optimization techniques discussed in class to find an open loop solution. With a possible extension of adding Model-predictive Control if time permits as suggested by Dr. Hubicki. I am planning on using MATLAB to implement both the simulation and controller but want to investigate the benefits of using Python further before making this a concrete decision. I do not believe I will need any additional computational requirements for this project as of now.

References

[1] Y. Xiong, *Racing Line Optimization* , Massachusetts Institute of Technology, 2010