Chair for Computation in Engineering and MathWorks
Department of Civil, Geo and Environmental Engineering
Technical University of Munich

MathWorks®

An Application to Optimize Lap-times for Race Cars

Software Lab Project 2019

Alaa Bashir, Ankit Khot, Máté Kelemen
Jan Janse van Rensburg and Steve Schäfer

UNREAL ENGINE

Abstract and Motivation

With the ever increasing advancements in autonomous driving and their use in motorsports, the need for a robust optimal path finding application has become crucial. The MATLAB app created in this project encapsulates every requirement to achieve this goal: importing race tracks from aerial photographs or GPS data, finding the optimal path for any combination of tracks and vehicles, and verifying the results with a high-fidelity Simulink model within a 3D simulated Unreal environment.

Path Optimization

A low-fidelity point mass vehicle model is used for the optimization, which relies on lookup tables generated by complex Simulink models, effectively combining reduced computational costs with detailed vehicle behaviour. Contrary to existing methods in the field that approximate trajectories as a collection of line segments, the proposed approach represents paths as splines with continous geometric properties, resulting in a more accurate and flexible discretization. The optimization uses a nonlinear constrained solver and can be carried out with respect to a variety of objectives.

Race Circuit Geometry and Vehicle Data

<u>Race Circuit Geometry :</u>
The race circuit geometry is generated from the Google maps, by laying the points along with the two bounds of the track (see figure 1). The GPS Exchange file containing the longitudes and latitudes are then processed MATLAB's Mapping Toolbox to get x-y coordinates.
<u>Vehicle Data :</u>
For the optimization, we require the path curvature-critical velocity, velocity-maximum acceleration, and velocity-maximum deceleration data and they are obtained by following methods respectively :
1. <u>Cornering velocity:</u> The relation path curvature- critical velocity is obtained from the dynamic force and moment equilibrium of the vehicle at a corner.
2. <u>Wide-open throttle test:</u> The Vehicle model is set to run on a straight path with maximum acceleration. The velocity and acceleration data thus obtained are stored and processed.
3. <u>Full brake test:</u> The vehicle is first allowed to reach its peak speed and then suddenly forced to brake until it stops. Velocity and deceleration data are obtained and processed.
The data obtained above will then be input to the optimization process

Fig.1: Hockenheimring

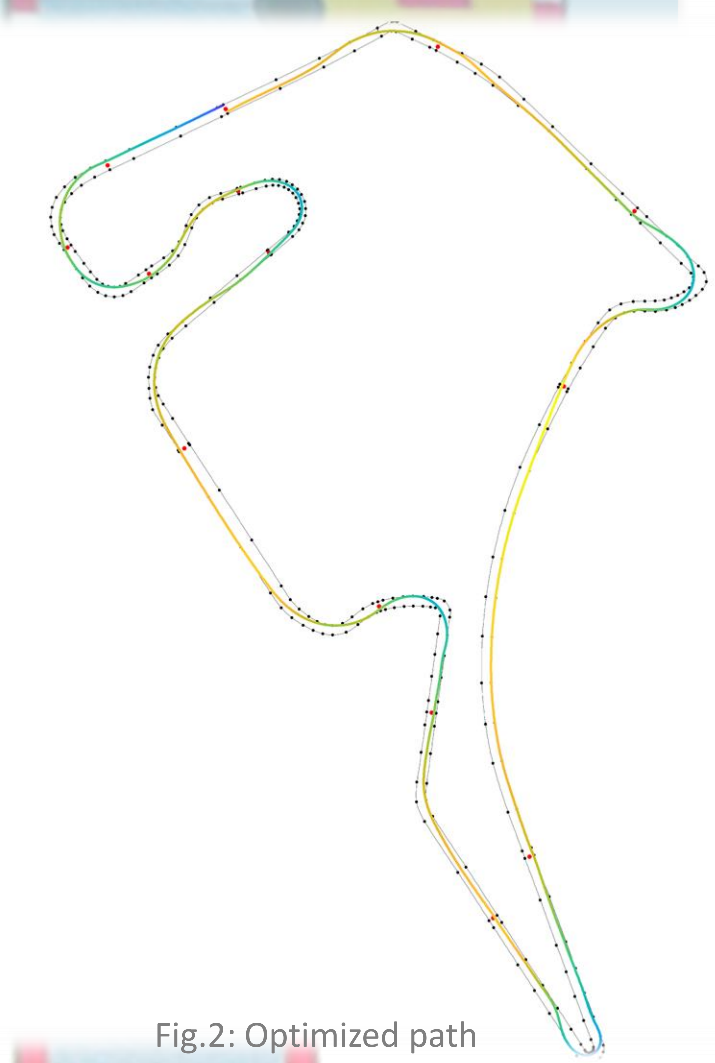

Fig.2: Optimized path

Simulation and Parameter Estimation

<u>Simulation:</u>
The optimization results are preprocessed to obtain the reference positions, yaw angles, velocity profile, path curvature and boundaries. These variables are fed to a Simulink Model that simulate the vehicle motion along the optimized path. In this model, two controllers are used: Lateral and Longitudinal Controllers which control the steering angle and velocity respectively. The resulted signals are fed to 3 DOF Vehicle and its response is simulated in XY Plotter and Unreal Engine.
<u>Parameter Estimation</u>
The optimum vehicle parameters (mass, wheelbase, COG location and axle height) are estimated using Least Square Optimization Method. This is done by considering the reference position as the desired output and changing the parameter until the simulated and desired outputs match.