ME231A Group 14 Project 2021

F1 Race Car Control using Model Predictive Control

Shravan Tangudu (3037428341), Pranav Vanjani (3037396725), Karan Jhaveri (3037396478), and Andrew Flach (3033686296)

University of California Berkeley

January 30, 2022

Abstract

Within this work, we explore and propose a method to generate the fastest lap time and optimal trajectory for Formula 1 vehicles over a mapped race course. Existing approaches solve for time and attempt to optimize this variable to find the fastest time around the track but this can be computationally heavy while yielding lackluster results. To solve this problem, we introduce an MPC controller and optimize for velocity as our dependant variable. To do this we have used a bicycle model with model parameters from the Mercedes AMG Petronas Formula one W10 Formula 1 car from 2019 Formula 1 season and tried to find the optimal trajectory and time for going around multiple racetracks and compared it with the actual lap times around the track in 2019 for validation of the model. Video: https://bit.ly/322mFFQ

Keywords
Model Predictive Controller, F1, Forward Euler Discretization, Velocity Optimization

1 Introduction

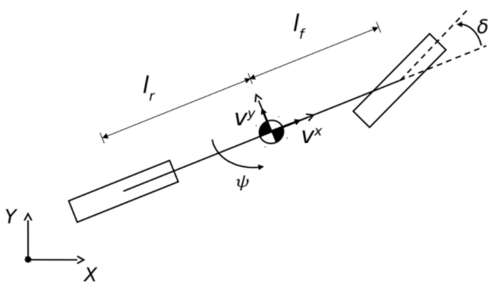
Optimizing the time around the track has historically been a popular feat that many F1 teams and other racing teams alike have been hoping to achieve in attempts to beat out the competition. New emerging engine technologies motivate the development of time-optimized track times in the entertainment world of Formula 1. One popular approach, as shown in [1], is nonlinear model predictive control where the time-optimal objective is reformulated such that it can be solved by existing efficient algorithms that build on the generalized Gauss-Newton method. The authors use real-world hardware setup of miniature race cars to numerically validate their simulations and experimentally compare different approaches. The car is modeled after the Bicycle Model in where spatial reformulation is applied to the dynamic system with time being the independent variable. The paper implements RC track conditions through their experimental setup with miniatures vehicles which might not perform well at high velocities due to slip effects not being modeled along with other factors such as tire coefficients. In this paper, we also focus on velocity-optimization using the bicycle model but with more complex F1 race tracks to compare the simulated lap times to F1 lap times.

2 Theory

2.1 Modeling

For our vehicle model we decided to go with a bicycle model for reducing the computational load for the current setup, which could be later substituted for a more complex model for higher accuracy. Road tyre interactions have been neglected in this model for computational ease, giving us a slip-free bicycle model.

Figure 1: Bicycle Model



The bicycle parameters will be defined in the

Time-Domain model.

2.1.1 Time-Domain model

A rigid body chassis is used with heading angle ψ and absolute velocity v. Equations for the model used in time domain are:

$$\dot{X} = v\cos(\psi + C_1\delta) \tag{1}$$
$$\dot{Y} = v\sin(\psi + C_1\delta) \tag{2}$$
$$\dot{\psi} = v\delta C_2 \tag{3}$$
$$\dot{v} = a \tag{4}$$

where δ and a represent the steering angle and the acceleration of the car Parameter $C_1 = l_r/(l_r + l_f)$ and $C_2 = 1/(l_r + l_f)$. If we assume that our steering angle is small, we can use the following approximations:

$$v^x = v, \qquad v^y = vC_1\delta$$

The car velocity can then be calculated as

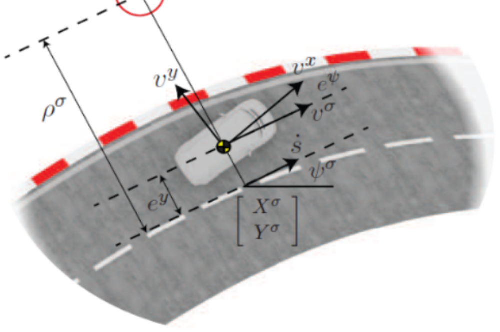
$$v^2 = (v^x)^2 + (v^y)^2$$

Equations (1)-(4) are for the kinematics of rigid body of the bicycle model. We will use this as a way to model the constraints and boundaries of the track. In the future, The maximum speed of velocity will be limited to an appropriate boundary to imitate real-world scenarios.

2.1.2 Spatial Reparameterization

Since the time-domain model is clearly a function of time while we need time to be an optimization variable, we can reparameterize the $[X, Y]^T$ coordinate system by transforming it to a function of the arc length $\sigma(s)$ on the curve σ.

Figure 2: Vehicle Dynamics Coordinates Definition of Spatial Reparameterization



Hence, the states X, Y and ψ can be replaced by

$e^y = cos(\psi^\sigma)(Y - Y^\sigma) - sin(\psi^\sigma)(X - X^\sigma), and$ $e^\psi = \psi - \psi^\sigma$, where $[X^\sigma, Y^\sigma]^T$ and ψ^σ are the position and orientation of the reference point on the path s. Assuming that the car does not stop, meaning $\dot{s} > 0$, then for the state vector $\xi = [e^y, e^\psi, v, t]$:

$$\xi' = \frac{d\xi}{ds} = \frac{d\xi}{dt}\frac{ds}{dt} = \frac{1}{\dot{s}}\dot{\xi} \tag{5}$$

Using components of the velocity along the X-Y reference, we can compute the velocity v^σ with respect to $\dot{s}$ as:

$$v^\sigma = (\rho^\sigma - e^y)\dot{\psi}^\sigma = v^x cos(e^\psi) - v^y sin(e^\psi),$$

where ρ^σ is the local radius of curvature of σ. Hence, the velocity, $\dot{s}$ is given by:

$$\dot{s} = \rho^\sigma \dot{\psi}^s = \frac{1}{1 - \frac{e^y}{\rho^\sigma}}(v^x cos(e^\psi) - v^y sin(e^\psi))$$

We hence obtain the spatial dynamic system as:

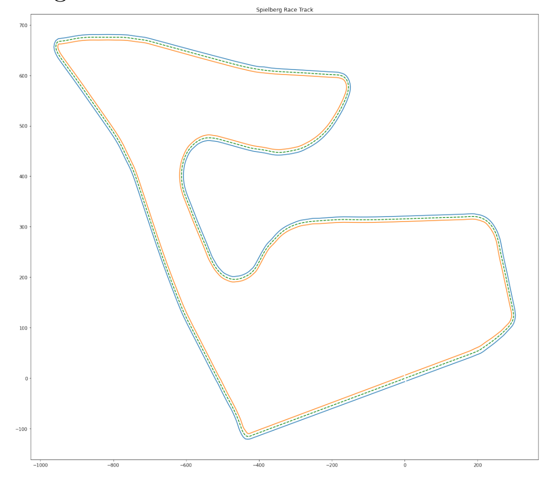
$$e^{y\prime}(s) = (vcos(e^\psi) + vsin(e^\psi))/\dot{s} \tag{6}$$
$$e^{\psi\prime}(s) = \dot{\psi}/\dot{s} - \kappa^\sigma(s) \tag{7}$$
$$v'(s) = \dot{v}/\dot{s} \tag{8}$$
$$t'(s) = 1/\dot{s}, \tag{9}$$

where $\kappa^\sigma(s) = 1/\rho^\sigma(s)$ is the local curvature.

2.1.3 Track Model

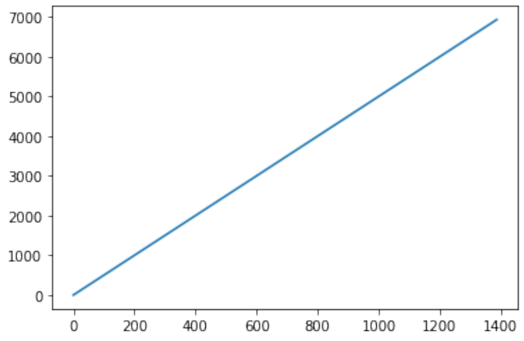
The track used, taken from [2], is imported with values of x and y coordinates for the entirety of the track. These coordinates are used as the Center line to find the track width which are converted to the left and right boundaries of the track as seen in Figure 2.

Figure 3: Track with limits and Centerline



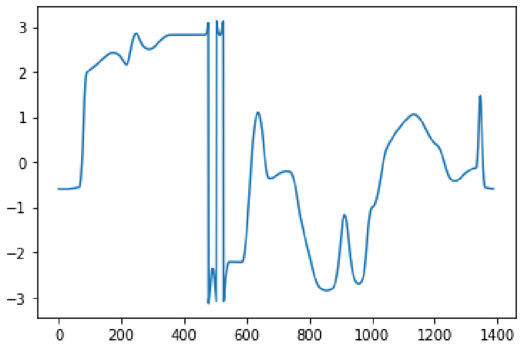
To map out the track boundaries, first we find the distance of each point on the centerline from the start of the track. The coordinates on the track are acquired using x and y while the numpy linalg operation calculates the non-linear portions of the track. These values are then added together to get the total length of the track in terms of S.

Figure 4: Distance of track in terms of S



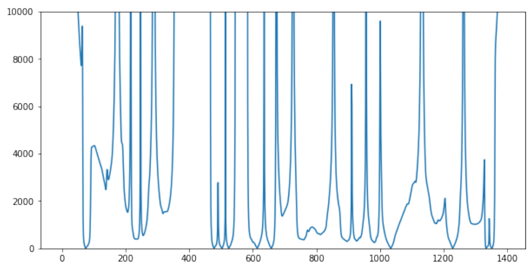
After finding the total distance of the track, the slope of each point on the centerline is calculated. These slopes are need to calculate the radius of curvature to use in our velocity optimization.

Figure 5: Slope of each point on the Centerline



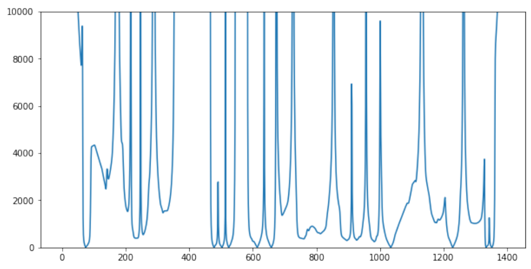
In order to find the radius of curvature, the double derivative is taken from the x and y coordinates of the track. These are then inputted into a curvature equation and plotted using python. This will then be used to calculate the maximum velocities possible at each point on the track.

Figure 6: Radius of Curvature at each point of the track



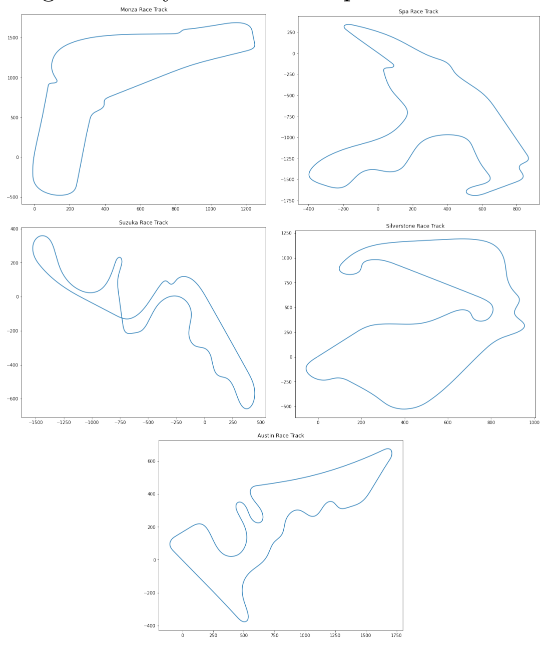
As input parameters, we limit the coefficient of friction to a value of 1.7 and the vehicles maximum velocity to 89 meters per second. This will allow us to visualize the range of velocities around every point of the track without it going to infinity. These constraints make it so we can optimize the time around the track.

Figure 7: Max Velocities for each point of the track



After computing these graphs, we then repeated the process for the remaining 5 race tracks in order to compare them to the actual F1 race times. As the tracks got longer and more complex in their radius of curvature, we noticed that the computation times were steadily increasing. In particular, the model seems to have difficulty when predicting around sharp radii.

Figure 8: Layout of Time-Optimized Tracks



2.1.4 Car Model

We decided to use the Mercedes AMG Petronas Formula one W10 Formula 1 car from 2019 Formula 1 season for setting parameters and constraints in the model which have been listed below:

Property of model	Value
mass of the car	736 Kg
steering angle	-30^o – 30^o
wheelbase	3.698 m
l_f	1.4792 m
l_r	2.2188 m
Coefficient of friction	0.9
max braking(-acc)	-49.05 m/s^2
max acceleration	10.68 m/s^2
top speed	89 m/s

Next we used the radius of curvature at every point on the track to find the maximum limiting velocity the car can take in that corner using the basic equation of centrifugal force and no slip condition.

$$m{\upsilon_{max}}^2/r = mgf \tag{10}$$

where,
m = mass of the car
υ = Limiting Velocity of the car
g = Acceleration due to gravity
f = coefficient of friction

2.2 MPC Design

For our MPC design we decided to use the standard non-linear MPC design as shown below:

$$J_o^*(x(t)) = min \quad P(x_n) + \textstyle\sum_{k=0}^{-1} q(x_k, q_k)$$

$subject\ to:$
$x_{k+1} = g(x_k, u_k),\ k = 0, ..., N-1$
$x_k \in \chi, u_k \in \mu,\ k = 0, ..., N-1$
$x_o = x(t)$

For our MPC, we chose the following tuning weights for optimizing time around the track

Q = diag([5 · 10−4, 10−10, 10−10, 10−10])

R = diag([10−3, 10−10])

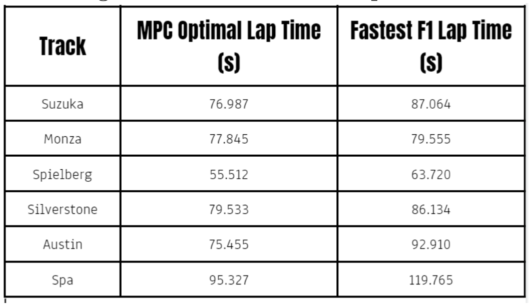
P = diag([10−10, 10−10, 10−10, 1])

keeping the weight of optimizing time as highest among all the parameters as that is what we are optimizing using this controller. you can see the weight for e^y is low so that the controller allows the car to deviate from the center line while cornering to increase speed in the corners and reduce overall time. We had to keep the horizon short because of limitations in pyomo we could not change the radius of curvature for our track in the CFTOC, it had to be a fixed parameter, but we were able to achieve reasonable accuracy by keeping the horizon short and increasing the simulation time.

3 Simulation and Results

After running the MPC model, we compared the results of lap times to the previous year's F1 fastest lap time. We noticed that the results were fairly close to each other in terms of raw time. These results are reasonable because we did not include friction parameters for the tire to road interaction, weight of driver, or fuel consumption. We expect that the times will be faster than the realistically achieved times due these imperfect conditions. The times that came the closest was the Monza track as it has very little curvature in which the driver has to slow down. This result was particularly interesting because the optimal lap time calculated was 77.845 while the fastest lap time was 79.555 which means that there is a very small margin for improvement when comparing to the model. Overall, the results seems to contain convincing results that our model was efficient in predicting the optimal track times in terms of maximum velocity.

Figure 9: Track time comparison



Track	MPC Optimal Lap Time (s)	Fastest F1 Lap Time (s)
Suzuka	76.987	87.064
Monza	77.845	79.555
Spielberg	55.512	63.720
Silverstone	79.533	86.134
Austin	75.455	92.910
Spa	95.327	119.765

Conclusions

We were able to successfully optimize the time for multiple tracks and obtain results that lined up fairly accurately with the real F1 lap times. We were also able to optimize the velocity around the entire track by utilizing spatial reformulation where we used the radius of curvature as a function. Reformulating the nonlinear bicycle model to spatial coordinates allowed us to optimize the time-tracking objective for accurate results. In the future, we plan on introducing friction models as the bicycle model has proved insufficient at high velocities due to the absence of slip effects not being modeled. Real-world track times show that there is a clear deviation without slip effects and could further be researched to include these in a non-computationally heavy manner.

Acknowledgements

Francesco Borrelli, Junseok Lee, and Edward Zhu

References

[1] R. Verschueren, S. De Bruyne, M. Zanon, J. V. Frasch and M. Diehl, "Towards time-optimal race car driving using nonlinear MPC in real-time," 53rd IEEE Conference on Decision and Control, 2014, pp. 2505-2510, doi: 10.1109/CDC.2014.7039771.

[2] Tumftm, “TUMFTM/Racetrack-database: This repository contains center lines (x- and Y-coordinates), track widths and race lines for over 20 race tracks (F1 and DTM) all over the world,” GitHub. [Online]. Available: https://github.com/TUMFTM/racetrack-database. [Accessed: 15-Dec-2021].

[3] Urosolia, “Urosolia/RacingLMPC at Devel-Ugo,” GitHub. [Online]. Available: https://github.com/urosolia/RacingLMPC/tree/devel-ugo. [Accessed: 15-Dec-2021].