Racing Trajectory and Control Optimization with Model Predictive Control

Preliminary Report

MAE 598: Design Optimization

Kenneth Lozes and Tanner Bitz

November 4, 2018

Abstract

This paper explores the use of model predictive control(MPC) in a race car environment. A race environment poses problems for a controls engineer in that the car must optimally plan and execute a path around the track while taking into account numerous nonlinear models and constraints. Path planning is also made more difficult in this simulation by adding static and dynamic obstacles. To optimize the lap time, a measure of the parametric distance is maximized which also minimizes heading error. The optimization is constrained by factors such as vehicle dynamics, tire slip models, corridor planning, and input limits like maximal steering angle and torque from the engine.

Nomenclature

α	slip angle
β	turning angle
$\underset{.}{m}$	longitudinal load balance
δ	steering angle
κ	trajectory
ν	width
ω_b	yaw rate of body
ω_w	tire rotational speed
ρ_s	track curvature
σ	tire slip
φ	angle relative to inertial frame
C	tire stiffness
d	duty cycle of motor
e_y	lateral error to reference path
e_ψ	heading error to reference path
F	force
g	gravitational acceleration
h_{cg}	height of center of gravity above axel
I	moment of inertia
l	length
M	moment
m	mass
s	track progress arc length
T	Torque
T_s	sampling period
u	control input vector
v_x	x velocity in body coordinates
v_y	y velocity in body coordinates
X	x position in inertial coordinates
x	state vector
Y	y position in inertial coordinates

Contents

List of Figures

1 Problem Statement

Many papers have been published in recent years about the autonomous control of vehicles. One context of interest is a race environment as it allows for the simulation of control up to the mechanical limits of the vehicle. This research project will investigate several different race environments with the use of model predictive control (MPC). The environments with which this paper is focused is a car racing around a track with:

- no obstacles
- static obstacles such as an object blocking part of the track path
- dynamic obstacles such as an additional car:
  - moving along the center line of the track
  - with an equivalent racing controller, but with different kinematic parameters
  - that races and also attempts to block passing

Given this paper in based in a race context, the first and most important goal is to maximize the car's progress along a path. This problem can be broken down into two separate problems. The first problem is a path planning problem which tries to optimize the path of the vehicle due to some objective function and constraints. The second part of the problem deals with how well the vehicle adheres to the planned path. The MPC framework was chosen because it provides a flexible environment to formulate these problems and address their many constraints. The next section will focus on how MPC allows us to formulate each of these race scenarios.

Note: We intend to investigate these items in the order they are listed. As a matter of scope, we recognize that the items relating to the dynamic obstacles are significantly more difficult and it's possible we will not have the time to finish one or two of them.

1.1 Model Predictive Control Framework

Model predictive control solves an optimization problem to determine optimal control inputs such that the system follows a set of desired state trajectories. MPC does this by using a model of the system, which allows it to predict it's state response to given inputs at many sampling times in the future. The number of future sample times used in the optimization problem is called the prediction horizon. Ideally, the optimization would be predicted infinitely into the future as this would allow for the optimal set of the control inputs to be found for all time. However, the longer the prediction horizon, the more computationally expensive the optimization problem becomes; therefore the prediction horizon will be limited by the computational power of the controller.

Equation 1, originally from [1], shows a useful general formulation for this problem.

$$\begin{aligned} \underset{u}{\text{maximize}} \quad & \text{vehicle progress} \\ \text{subject to} \quad & \text{vehicle dynamics} \\ & \text{road constraints} \\ & \text{input constraints} \end{aligned} \tag{1}$$

In each of our cases, a vehicle progress measure must be determined. To do this, a track will be made and mathematically defined using parametric equations in the form of b-splines or track segments made of connected primitive curves. Therefore, the equations that describe the x and y positions of the track in a global frame will be function of a parametric variable, s, which will be scaled 0 to the arc length of the track centerline. The vehicle progress measurement will be the projection of the predicted vehicle path with respect to the parametric curve describing the track.

Vehicle dynamics constraints are the set of constraints that describe the motion of the car due to the physical nature of the vehicle. These physical constraints define the equations of motion of the vehicle body in the global and body frames. Included in this set of constraints is a tire model that defines the forces each tire experiences and the corresponding slip, if there is any. These nonlinear dynamics equations will form our state equations, $\dot{(x)} = f(x, u)$. MPC is computationally expensive, so in an effort to reduce computation, these equations will be linearized such that $\dot{x} = Ax + Bu$. Given this controller will be implemented in a discrete environment, the differential equations will be converted to difference equations $x_{k+1} = A_d x_k + B_d u_k$.

Road constraints are the next set of constraints to discuss. Road constraints will differ depending on the setting of simulation. In the most trivial case, the simulation without any obstacles, the road conditions will include equations that define a boundary that our car cannot cross (or will pay a high cost in the objective function). These can be defined in several ways, the simplest of which is to define two side boundaries by half-spaces some distance away from the center of the track. When there are obstacles, the road constraints get more complicated. One way to deal with this is to define each side of the obstacle as a half-space that the car is not allowed to enter into. This creates two paths around the obstacle that the car can take. Road constraints will be discussed in further detail in the model analysis section.

The last set of constraints is input constraints. These ensure that the input on the system is a reasonable size. The two main inputs for our system will be thrust and steering angle. Both of these input have limits in reality; the engine cannot output infinite torque to the wheels and the suspension can only be turned to a certain steering angle. For example, the steering angle constraint would be set with the equations $\delta \leq \delta_{max}$ and $delta \geq \delta_{min}$.

2 Mathematical Model

In this section we describe the mathematical models used in the paper. We require models for the car dynamics, tire/motor forces as well as the race track. Our experiments are done entirely in a simulated environment, so we must distinguish between the models used for the simulation and those used by the model predictive controller. We want our simulation to accurately reflect reality, so we choose the highest fidelity models for this purpose. Model predictive control also requires accurate models, however, increased model complexity can lead to impractical computation time in real applications. So we must consider this balance when choosing a prediction model. We test 3 different car models in our MPC, each with increasing fidelity, and analyze how they affect its performance.

2.1 Kinematic One Track Model

The first is a simple kinematic once track model, which is commonly called a bicycle model. Being a kinematic it does not account any affects due to forces/moments on the body of the vehicle. Because of this, constraints preventing infeasible motion must be explicitly added to our optimization problem. The equations of motion are shown below in 2.

$$\begin{aligned}
\dot{X} &= v_x \cos(\varphi) - v_y \sin(\varphi) \\
\dot{Y} &= v_x \sin(\varphi) + v_y \cos(\varphi) \\
\dot{\varphi} &= \frac{v_y}{l_r} \\
\dot{v}_x &= a \cos(\beta) \\
\dot{v}_y &= a \sin(\beta) \\
where &: \\
\beta &= \tan^{-1}\left(\frac{l_r}{l_f + l_r}\tan(\delta)\right)
\end{aligned} \tag{2}$$

2.2 Kinetic One Track Model

The second is a kinetic one track model. Following [2] we account for lateral forces, but not longitudinal forces on the tires. This allows us to ignore the rotational motion of the wheels, which should prevent small timestep restriction due to fast dynamics of the wheels. We also assume that the each wheel carries half the weight of the vehicle at all times. The equations of motion for this model are derived from [2] and shown below by 3.

$$\begin{aligned}
\dot{X} &= v_x \cos(\varphi) - v_y \sin(\varphi) \\
\dot{Y} &= v_x \sin(\varphi) + v_y \cos(\varphi) \\
\dot{\varphi} &= \omega_b \\
\dot{v_x} &= \frac{1}{m}(F_{r,long} - F_{f,lat}\sin(\delta) + m v_y \omega_b) \\
\dot{v_y} &= \frac{1}{m}(F_{r,lat} + F_{f,lat}\cos(\delta) - m v_x \omega_b) \\
\dot{\omega}_b &= \frac{1}{I_z}(F_{f,lat} l_f \cos(\delta) - F_{r,lat} l_r)
\end{aligned} \tag{3}$$

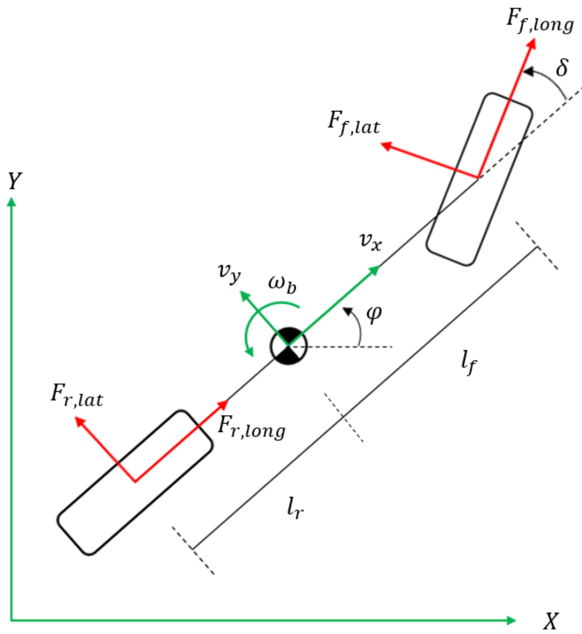

Figure 1: Kinetic One Track Model

2.3 Kinetic Two Track Model

The third model is a two track model. This model improves upon the one track model by accounting for load transfer between the 4 wheels due to lateral and longitudinal forces on the vehicle. We also choose to account for longitudinal tire slip in this model by including tire rotational speed as state variables. The equations of motion for this model are derived from [3] and [2].

$$\begin{aligned}
\dot{X} &= v_x \cos(\varphi) - v_y \sin(\varphi) \\
\dot{Y} &= v_x \sin(\varphi) + v_y \cos(\varphi) \\
\dot{\varphi} &= \omega_b \\
\dot{v_x} &= \frac{1}{m}(F_{rl,long} + F_{rr,long} - F_{fl,lat}\sin(\delta) - F_{fr,lat}\sin(\delta) \\
&\quad + F_{fl,long}\cos(\delta) + F_{fr,long}\cos(\delta) + m v_y \omega) \\
\dot{v_y} &= \frac{1}{m}(F_{rl,lat} + F_{rr,lat} + F_{fl,lat}\cos(\delta) + F_{fr,lat}\cos(\delta) \\
&\quad + F_{fl,long}\sin(\delta) + F_{fr,long}\sin(\delta) - m v_x \omega) \\
\dot{\omega}_b &= \frac{1}{I_z}(F_{fl,lat} l_f \cos(\delta) + F_{fr,y} l_f \cos(\delta) - F_{rl,lat} l_r \\
&\quad - F_{rr,lat} l_r - F_{rl,long} w_l + F_{rr,long} w_r) \\
\dot{\omega}_{w,fl} &= T_{fl} - r F_{fl,long} \\
\dot{\omega}_{w,fr} &= T_{fr} - r F_{fr,long} \\
\dot{\omega}_{w,rl} &= T_{rl} - r F_{rl,long} \\
\dot{\omega}_{w,rr} &= T_{rr} - r F_{rr,long}
\end{aligned} \tag{4}$$

The tire forces for both the one track and two track model are computed using Fiala tire model defined in the subsection below. The normal forces on each tire are subject to load transfer due to lateral and longitudinal forces on the vehicle as shown in Equation 5. Since the forces Fx and Fy are computed from the tire friction forces, for which Fz is needed, they are not yet available for the calculation. We instead use the tire forces from the previous time step to compute Fx and Fy.

$$\begin{aligned}F_{fl,z} &= \frac{1}{2}mgd_m - \frac{F_x h_{cg}}{l} + \frac{F_y h_{cg}}{\nu}\\ F_{fl,z} &= \frac{1}{2}mgd_m - \frac{F_x h_{cg}}{l} + \frac{F_y h_{cg}}{\nu}\\ F_{fl,z} &= \frac{1}{2}mg(1-d_m) - \frac{F_x h_{cg}}{l} + \frac{F_y h_{cg}}{\nu}\\ F_{fl,z} &= \frac{1}{2}mg(1-d_m) - \frac{F_x h_{cg}}{l} + \frac{F_y h_{cg}}{\nu}\end{aligned} \tag{5}$$

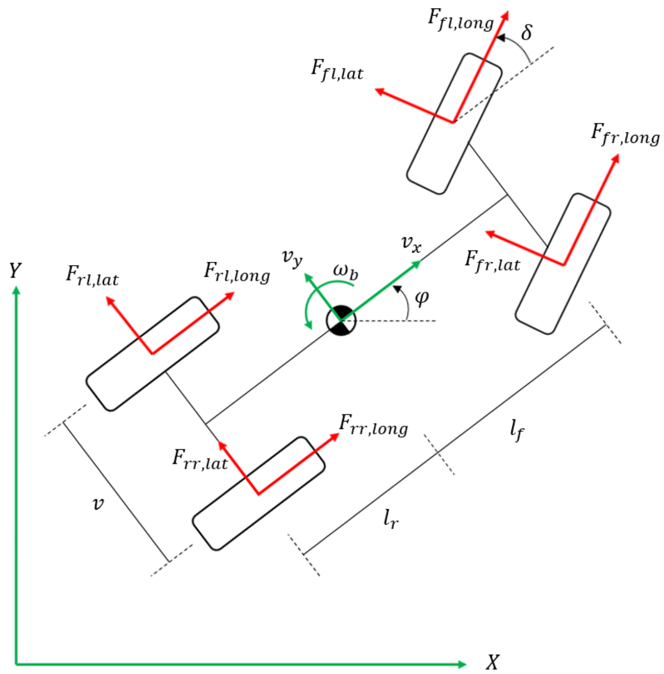

Figure 2: Kinetic Two Track Model

2.4 Fiala Tire Model

The Fiala tire model assumes lateral and longitudinal stiffness C_{lat} and C_{long} are equal. The total slip, σ is computed with equation 6.

$$\sigma = \sqrt{{\sigma_{lat}}^2 + {\sigma_{long}}^2} \tag{6}$$

Lateral and longitudinal slip are calculated with equations 7 and 8.

$$\begin{aligned}\sigma_{lat} &= \frac{r\omega_w - v_x}{r\omega_w}\text{during acceleration}\\ \sigma_{lat} &= \frac{r\omega_w - v_x}{v_x}\text{during braking}\end{aligned} \tag{7}$$

$$\sigma_{long} = \frac{v_x}{r\omega_w}\tan(\alpha) \tag{8}$$

The total force on each tire is calculated using 9

$$\begin{aligned} F_t &= \mu F_z\big(3\theta\sigma - 3(\theta\sigma)^2 + (\theta\sigma)^3\big) & if \sigma < \sigma_m \\ F_t &= F_z & if \sigma \geq \sigma_m \end{aligned} \tag{9}$$

Sliding is initiated when total force on the tire is greater than the coefficient of friction multiplied by the normal force on the tire, $F_t = \mu F_z$. At this point, the longitudinal and lateral forces on each tire can be calculated with equations 10 and 11.

$$F_{long} = \frac{\sigma_{long}}{\sigma} F_t \tag{10}$$

$$F_{lat} = \frac{\sigma_{lat}}{\sigma} F_t \tag{11}$$

where:

$$\theta = \frac{1}{\sigma_m} = \frac{C_r}{3\mu F_z} \tag{12}$$

Note: $C_r = C_f$

3 Model Analysis

In this section we describe in detail the model predictive control technique we plan to use. The basis of our controller is formulated in [1]. We discuss the changes made to improve upon this controller by modifying constraints and objective functions.

3.1 Progress Maximization Model Predictive Controller

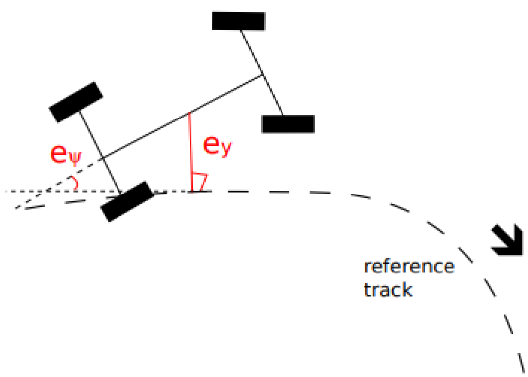

Figure 3: Lateral and heading errors, e_y and e_ψ. Diagram from [1]

The goal of the MPC controller is to output a set of inputs, u, that maximizes the predicted movements of the vehicle model along the track at each time step. It follows then, that the objective function must maximize the distance along the track. First, we need a measure of error relative to the track. In this case lateral and heading error are defined as the distance away the track and the heading error is the difference in angle of the body of the vehicle to the tangent of the track at the point nearest the vehicle. These terms are denoted as e_ψ and e_y and can be seen in Figure 3.

The distance traveled in one time step is Δs. Thus we are looking to minimize the negative of the sum of the steps. The amount the vehicle moves in one time step is calculated with equation 13.

$$\Delta s = T_s v \frac{\rho_s \cos(e_\psi)}{\rho_s - e_y} \tag{13}$$

Therefore, the objective function becomes:

$$J(x_t, u) = -\sum_{k=0}^{N-1} \Delta_{s,k} = -\sum_{k=0}^{N-1} T_s v_k \frac{\rho_{s,k} \cos(e_{\psi,k})}{\rho_{s,k} - e_{y,k}} \tag{14}$$

The prediction model is referred as f:

$$x_{k+1} = f(x_k, u_k) \tag{15}$$

The car's trajectory curvature κ is described with equation 16

$$\kappa = \frac{\tan(\delta)}{l} \tag{16}$$

where l is the total length of the car, $l = l_r + l_f$.
Our state's are $x = (e_\psi \;\; e_y \;\; v)^T$

$$x^{'} = f(x, u) = \begin{bmatrix} \frac{(\rho_s - e_y)\kappa}{\rho_s \cos(e_\psi)} - {\psi_s}' \\ \frac{(\rho_s - e_y)}{\rho_s} \tan(e_\psi) \\ 0 \end{bmatrix} \tag{17}$$

We can now formulate the optimization problem below in terms of its nonlinear objective function and constraints. This would require a sophisticated nonlinear optimization solver. The computation time for this would likely be impractical, so we opt to linearize the problem before solving.

$$\begin{aligned} \min_u \quad & -\sum_{k=0}^{N-1} v_k \frac{\rho_{s,k} \cos(e_{\psi,k})}{\rho_{s,k} - e_{y,k}} \\ \text{s.t.} \quad & x_o = x(t) \\ & x_{k+1} = f(x_k, u_k) \\ & x_k \in X \\ & u_k \in U \end{aligned} \tag{18}$$

Equation 19 shows the linearized objective function.

$$\begin{aligned} J(x_t, u) = -\sum_{k=0}^{N-1} \Bigg(& \bar{v}(k)\bar{\rho_s}(k) \frac{\cos(\bar{e_\psi}(k))}{\bar{\rho_s}(k) - \bar{e_y}(k)} + \frac{\cos(\bar{e_\psi}(k))\bar{\rho_s}(k)\bar{v}(k)}{(\bar{\rho_s}(k) - \bar{e_y}(k))^2}(e_y(k) - \bar{e_y}(k)) \\ & - \frac{\sin(\bar{e_\psi}(k))\bar{\rho_s}(k)\bar{v}(k)}{\bar{\rho_s}(k) - \bar{e_y}(k)}(e_\psi(k) - \bar{e_\psi}(k)) + \bar{\rho_s}(k)\frac{\cos(\bar{e_\psi}(k))}{\bar{\rho_s}(k) - \bar{e_y}(k)}(v(k) - \bar{v}(k)) \Bigg) \end{aligned} \tag{19}$$

The system is discretized using $\Delta s = vT_s$ as the elementary step and the model function is linearized by expanding out a Taylor series centered about our current trajectory estimate. This linearization is shown in equations 20 and 21.

$$x_{k+1} = x_k + \Delta s \left(\sum_{i=1}^{M_x} \frac{\partial f}{\partial x_i}(\bar{x_k}, \bar{u_k})(x_{i,k} - \bar{x_{i,k}}) + \sum_{i=1}^{M_u} \frac{\partial f}{\partial u_i}(\bar{x_k}, \bar{u_k})(u_{i,k} - \bar{u_{i,k}}) + f(\bar{x_k}, \bar{u_k}) \right) \quad (20)$$

$$x_{k+1} = \; A(x_k - \bar{x}_k) + B(u_k - \bar{u}_k) \quad (21)$$

M_u is the number of inputs, and M_x is the number of states.

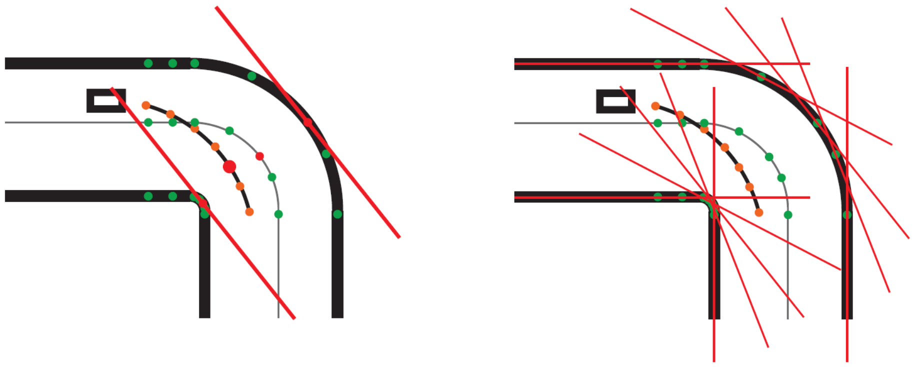

Figure 4: Corridor Planning from [2]

We now discuss in more detail the constraints in the optimization problem 22. Corridor planning is of particular interest as static and dynamic obstacles can be added. The cross sections of the track that the vehicle is able to pass through, unencumbered by obstacles or sides of the track define the corridors. The simplest case is when there is no obstacles in the track. The sides are defined by two opposing half-spaces such that $F_k x_k \geq f_{lb}$ and $F_k x_k \leq f_{ub}$. The complication happens when obstacles are added. Static obstacles that are close enough to the track sides such that the vehicle could not fit between the obstacle and the side, effectively move this side in. Therefore the corridor available for the car is either:

$$f_{lb} \leq F_k x_k \leq f_{ub} - \Delta f$$

or

$$f_{lb} + Deltaf \leq F_k x_k \leq f_{ub}$$

When the obstacle is centered such that the vehicle could go on either side of the obstacle then the corridors are:

$$f_{lb} \leq F_k x_k \leq f_{ob,l} \text{ or } f_{ob,u} \leq F_k x_k \leq f_{ub}$$

Dynamic obstacles will have a predicted path, which will define the obstacles upper and lower boundaries at each time step. The vehicle will not be able to cross these boundaries for all predicted time steps, just the same as static obstacles. More work will have to be done to formulate the dynamic obstacles predicted paths. The beginning trivial case we will investigate is a car following the track's centerline.

A few more constraints are needed to make the optimization problem effective and robust. We enforce that heading error be between -90 and 90 degree, so that the vehicle is always heading forward. We also set variable constraints on the vehicle steering input at each timestep. This is necessary when the kinematic prediction model is used so that the optimizer doesn't choose infeasible trajectories. The last constraint is on vehicle longitudinal acceleration. This can be in the form of direct acceleration or wheel torque commands. It is required that these terms be within realistic limits.

$$
\begin{aligned}
\min_{u} \quad & \quad J(u, x_t) \\
\text{s.t.} \quad & x_{k+1} = A(x_k - \bar{x}_k) + B(u_k - \bar{u}_k) \\
& f_{lb} \leq F_k x_k \leq f_{ub} \\
& -\frac{\pi}{2} \leq e_\psi(k) \leq \frac{\pi}{2} \\
& \delta_{k,min} \leq \delta_k \leq \delta_{k,max} \\
& a_{k,min} \leq a_k \leq a_{k,max}
\end{aligned}
\tag{22}
$$

References

[1] Florian Curinga. Autonomous racing using model predictive control, 2017.

[2] Alexander Liniger, Alexander Domahidi, and Manfred Morari. Optimization-based autonomous racing of 1: 43 scale rc cars. *Optimal Control Applications and Methods*, 36(5):628–647, 2015.

[3] S Van Koutrik. Optimal control for race car minimum time maneuvering. 2015.