Autonomous driving using Model Predictive Contouring Control

Abhijeet Kulkarni

May 2021

Abstract

This paper describes the control of autonomous vehicles based on the method of mathematical optimization. Using a dynamical model of the vehicle, control inputs are calculated by fixed horizon based predictive controller, where objective is to minimize the contouring and lag error subject to boundary constraints and avoiding obstacles. The control formulation used in this paper uses a linear time varying (LTV) approximation of the nonlinear dynamic model, to formulate a convex optimization program. The convex programs are solved sequentially to obtain an iteratively converged solution. The obstacle avoidance is incorporated by means of a high level corridor planner which results in a set of convex constraints. The Model Predictive contouring control is implemented in Python and tested for the track following and obstacle avoidance.

1 Introduction

High speed autonomous driving is a challenging task for automatic control systems due to the need for handling the vehicle close to its stability limits and in highly nonlinear operating regimes. Moreover, it becomes necessary for advanced path planning for previously unknown environment conditions. Highly nonlinear and fast dynamics of the vehicle puts an lower bound constraint on the control frequency, which usually lies within tens of milliseconds. It becomes even more challenging due to the limitation of computational resources on a mobile platform.

In this paper a convexified formulation of Nonlinear Model predictive contouring control , Liniger, Domahidi, and Morari 2014, is explored. The convex optimization problem can be efficiently solved in realtime on an embedded platform.

2 Models

2.1 Vehicle Model

For this implementation of MPCC, a two wheeled dynamic bicycle model is used, Velenis, Frazzoli, and Tsiotras 2009. The vehicle is modelled as rigid body with mass M and inertial I. Pitch and roll dynamics are neglected for simplicity. The brake force and engine thrust are modelled as longitudinal forces F_{x1} and F_{x2} on front and rear wheel respectively. The lateral forces F_{y1} and F_{y2}, which arise from tyre slip are modeled using Pacejka's magic formula, Pacejka 2012. Input steering angle is given by a_1 and a_2 are the distance between center of mass to front and rear wheel respectively.

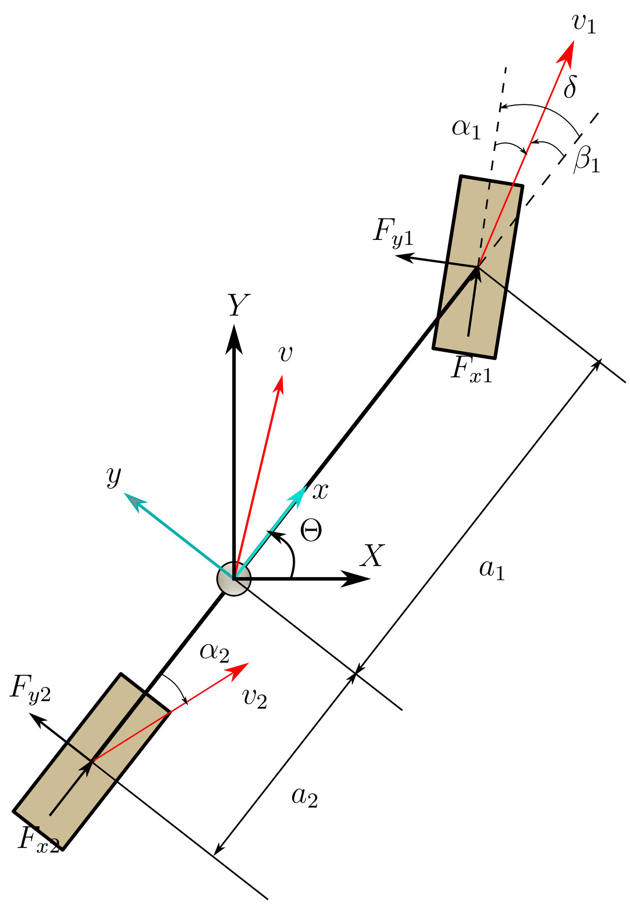


Figure 1: Dynamic Bicycle model

The state space of the Bicycle model is given by $q = \begin{bmatrix} X & Y & \theta & \dot{X} & \dot{Y} & \dot{\theta} \end{bmatrix}^T$. Inputs to the system is given by $u = \begin{bmatrix} \delta & F_{x1} & F_{x2} \end{bmatrix}^T$. The state-space equation of the system is given by,

$$\dot{q} = f(q,u) = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \\ \frac{F_{x1}\cos(\delta+\theta)+F_{x2}\cos(\theta)-F_{y1}\sin(\delta+\theta)-F_{y2}\sin(\theta)}{M} \\ \frac{F_{x1}\sin(\delta+\theta)+F_{x2}\sin(\theta)+F_{y1}\cos(\delta+\theta)+F_{y2}\cos(\theta)}{M} \\ \frac{-F_{y2}a_2+a_1(F_{x1}\sin(\delta)+F_{y1}\cos(\delta))}{I} \end{bmatrix} \tag{1}$$

Lateral forces are given by,

$$\begin{bmatrix} F_{y1} \\ F_{y2} \end{bmatrix} = \begin{bmatrix} -D_1 \sin\left(C_1 \operatorname{atan}\left(B_1\left(-\delta + \operatorname{atan}_2\left(-\dot{X}\sin(\theta)+\dot{Y}\cos(\theta)+\dot{\theta}a_1, \dot{X}\cos(\theta)+\dot{Y}\sin(\theta)\right)\right)\right)\right) \\ -D_2 \sin\left(C_2 \operatorname{atan}\left(B_2 \operatorname{atan}_2\left(-\dot{X}\sin(\theta)+\dot{Y}\cos(\theta)-\dot{\theta}a_2, \dot{X}\cos(\theta)+\dot{Y}\sin(\theta)\right)\right)\right) \end{bmatrix} \tag{2}$$

Where, D, C, B are the empirical constants obtained from, Liniger, Domahidi, and Morari 2014. For more information on the constants, reader is directed to Pacejka 2012.

2.2 Reference Path Model

For this paper, it is assumed that a track is provided. It's boundaries and center line are available as an vector of discrete planar points. In order to adapt MPCC for this paper's application, a local polynomial approximation is generated. Here, Bezier curves are used. A one dimensional Bézier curve of degree M by-

$$b(t) = \sum_{j=0}^{M} \alpha_j \frac{M!}{j!(M-j)!} t^j (1-t)^{M-j} \tag{3}$$

, where $0 \leq t \leq 1$ is the parameter and α's are the coefficients. Therefore, the reference path is given by, $\left(X^{\text{ref}}(t), Y^{\text{ref}}(t)\right) = (b_x(t), b_y(t))$. The coefficients are of the Bézier curve are found by least square fitting on the local points to the current vehicles position. Only a small part of the track is considered as reference path for the MPCC. The length of the reference path is analogous to the visibility. The algorithms for operations on the bezier curve are refered from the website Pomax 2011.

2.3 Error Model

The error is modelled as in Lam, Manzie, and Good 2010. In the figure 2[1], parameter t_r

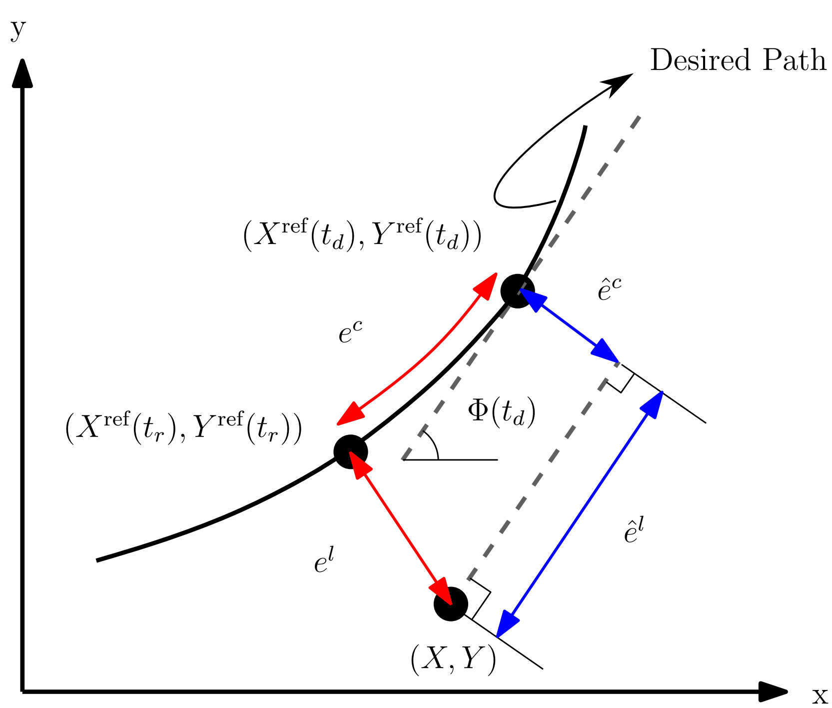


Figure 2: Contouring error, lag error and their approximations.

corresponds to closest point on reference trajectory to the current position of vehicle (X, Y). Parameter t_d corresponds to the desired progress on the reference trajectory. The lag and contouring error are e^l and e^c respectively. Finding t_r is an optimization program in itself. Therefore, to reduce the complexity, $t_r \approx t_d$ is assumed. Then the approximate errors become,

$$\hat{e}(q, t_a) = \begin{bmatrix} \hat{e}^c \\ \hat{e}^l \end{bmatrix} = \begin{bmatrix} \left(X - X^{ref}(t_a)\right)\sin\left(\Phi(t_a)\right) - \left(Y - Y^{ref}(t_a)\right)\cos\left(\Phi(t_a)\right) \\ -\left(X - X^{ref}(t_a)\right)\cos\left(\Phi(t_a)\right) - \left(Y - Y^{ref}(t_a)\right)\sin\left(\Phi(t_a)\right) \end{bmatrix} \tag{4}$$

where t_a is the approximately close parameter, i.e. t_d.

2.4 Boundary Constraints

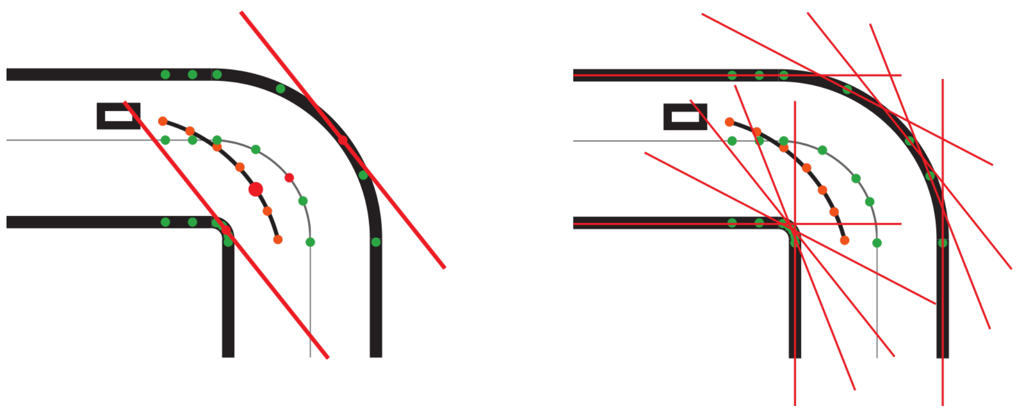


Figure 3: Left Figure: Each location (big red point) of center of mass is bounded by Halfspace (red lines). Right Figure: Resulting halfspaces (red lines) of the whole horizon.

One of the main advantage of using a Model predictive control based controller is the inclusion of constraints on the system. The boundary constraints that are available are non-convex in nature. To convexify these, a linear approximation of the boundaries are obtained by projecting the nominal position of the center of mass on the boundaries. The Figure 3[2]. shows the resulting convex bounds on the location of the COM at each time step.

3 MPCC

Now we have all the models necessary for the MPCC. But before implementation, the models need to linearized around the reference trajectory, i.e. the Bezier curve. The Errors and Dynamics is linearized at parameter values $t_{a,k+1} = t_{a,k} + v_k \mathrm{dt}$ for $k = 0, .., N-1$, where v_k can be viewed as a virtual input to the progress rate and N number of time steps in finite

[1]Figure 2 taken from Lam, Manzie, and Good 2010

[2]Figure 3 taken form Liniger, Domahidi, and Morari 2014

horizon of MPC. The convex optimization program is given by,

$$
\begin{aligned}
\underset{q_k,u_k,v_k,t_{a,k}}{\text{minimize}} \quad & \sum_{k=1}^{k=N} \left\| \begin{bmatrix} q_k \\ t_{a,k} \end{bmatrix} \right\|_{Qe_k}^2 + G_{e_k} \begin{bmatrix} q_k \\ t_{a,k} \end{bmatrix} - q_v v_k + \|u_k - u_{k-1}\|_{R_{\Delta u}}^2 \\
\text{subject to} \quad & q_0 = \bar{q}_0, t_{a,0} = \bar{t}_{a,0} \\
& q_{k+1} = A_k q_k + B_k u_k + G_k, k = 0, .., N-1 \\
& lb \leq q_{k+1} \leq ub \in \text{track}, k = 0, .., N-1 \\
& t_{a,k+1} = t_{a,k} + v_k \text{dt}, k = 0, .., N-1 \\
& |\bar{q}_k - q_k| \leq q_\Delta, k = 1, .., N \\
& |\bar{u}_k - u_k| \leq u_\Delta, k = 0, .., N-1 \\
& |\bar{v}_k - v_k| \leq v_\Delta, k = 0, .., N-2 \\
& 0 \leq t_{a,k} \leq 1, k = 1, ..N \\
& 0 \leq v_k \leq v_{\max}, k = 0, ..N-1
\end{aligned} \tag{5}
$$

Here, $(\bar{q}_k, \bar{t}_{a,k})$ are the nominal values of variables around which system is linearized. Q_{e_k} and G_{e_k} are obtained by linearizing the errors around the reference trajectory and weighted squaring it as follows,

$$Q_{e_k} = J_{e_k}^T W_e J_{e_k} \tag{6}$$

$$G_{e_k} = 2(e(\bar{q}_k, \bar{t}_{a,k}) - J_{e_k} \begin{bmatrix} \bar{q}_k \\ \bar{t}_{a,k} \end{bmatrix}) W_e J_{e_k} \begin{bmatrix} q_k \\ t_{a,k} \end{bmatrix} \tag{7}$$

where, J_{e_k} is the error gradient and is given by,

$$J_{e_k} = \frac{\partial \hat{e}(q_k, t_{a,k})}{\partial \begin{bmatrix} q_k \\ t_{a,k} \end{bmatrix}} \tag{8}$$

Constraint 2 is obtained as an Linear time varying approximation of the discretized nonlinear dynamics of the vehicle in 1. The matrices are given by,

$$A_k = (I + \left.\frac{\partial f}{\partial q}\right|_{\bar{q}_k, \bar{u}_k} dt) \tag{9}$$

$$B_k = \left.\frac{\partial f}{\partial u}\right|_{\bar{q}_k, \bar{u}_k} dt \tag{10}$$

$$G_k = f(\bar{q}_k, \bar{u}_k) dt - (A_k - I)\bar{q}_k - B_k \bar{u}_k \tag{11}$$

I is the identity matrix. Constraint 3 ensures the satisfaction of boundary constraint. Constraints 4-6 ensures the closeness of the values to the linear approximations. The above problem is solved sequentially and solution is obtained upon convergence of higher level optimization.

Algorithm 1: MPCC

```
Initialize q̄, v̄, ū;
while True do
    i ← 0 ;
    while (i < MAXIter) do
        Project Current COM position on centerline(blue dashed line);
        Perform CurveFitting to get bezier coefficients(2.2);
        Update constraints in eq (5);
        (q, v, u) ←Solve QP in eq (5);
        if ||q̄ − q||_2 < thresh then
            break;
        end
        (q̄, v̄, ū) ← (q, v, u);
        i ← i + 1 ;
    end
    Send first control input u_0 to vehicle;
    Pop first elements of (q̄, v̄, ū) and duplicate last element.(warm start);
end
```

The algorithm 1, gives the basic structure of the algorithm used to solve the MPCC problem for this project.

4 Simulation Results

The vehicle model used for the simulation are adapted from the ones used in Liniger, Domahidi, and Morari 2014.

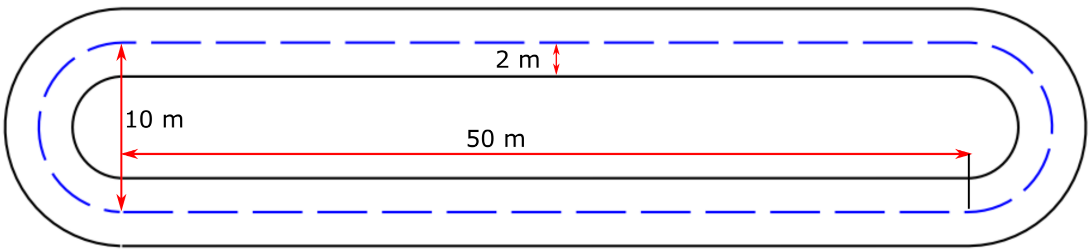
Parameter	Symbol	Unit	Value
TimeStep	dt	sec	0.005
Prediction Horizon	N	-	30
Gravity	g	m/s^2	9.81
Mass	M	kg	0.041
Inertia @COM	I	kgm^2	$27.8e^{-6}$
COM to front Length	a_1	m	0.029
COM to rear Length	a_2	m	0.033
Static Friction Coefficient	μ	-	0.9

Table 1: Simulated Vehicle's Inertia, Dimensions, and simulation parameters.

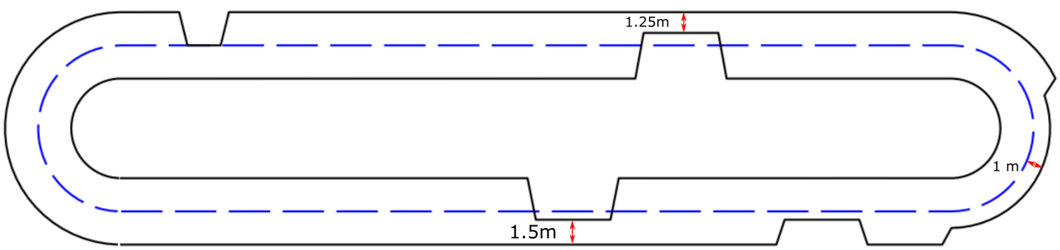
The controller will be tested on the vehicle parameters given in table 1 and 2. The track used are given in the figure (4a) and (4b).

Parameter	Value
D_1	$\mu M g \frac{a_2}{a_1+a_2}$
D_2	$\mu M g \frac{a_1}{a_1+a_2}$
C_1	1.2
C_2	1.2691
B_1	2.579
B_2	3.3852

Table 2: Simulated Vehicle's tyre model parameters.



(a) No Obstacle



(b) With Obstacles

Figure 4: Dashed blue line is the center line and is also the initial guess. Black line are the boundaries of the track.

4.1 Effect of Visibility

In this section the effect of visibility of the track is studied. The length of reference path, as defined in 2.2, is varied to see the effect on the time taken by the system to traverse the track. See figure

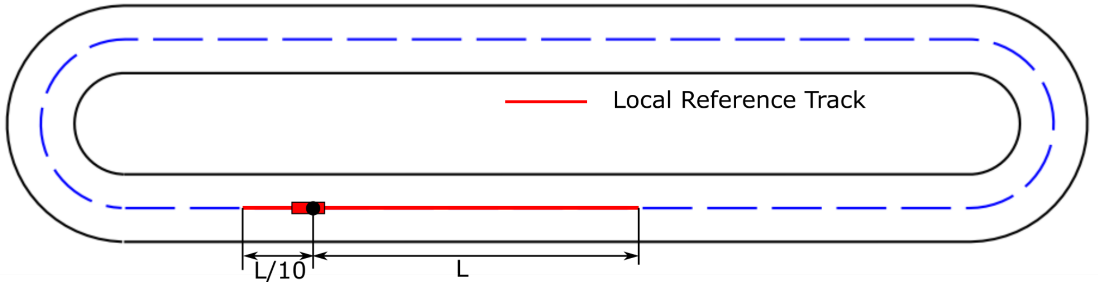


Figure 5: Red Line on the track is the track that is known by MPCC.

Reference Path Length (L) (m)	Time for 1 Lap (sec)
15	14
17.5	12.5
20	11.4

Table 3: Effect of visibility on lap time.

From the table 3, it is evident that as more information is available of the path ahead, the maximum speed, that can be achieved safely, increases. This effect is similar to human drivers on foggy weather condition.

4.2 Effect of Weights on contouring and lag errors

The quality of tracking the center line can be modified by changing the weight on the cost of the contouring error as defined in first row of equation (4).

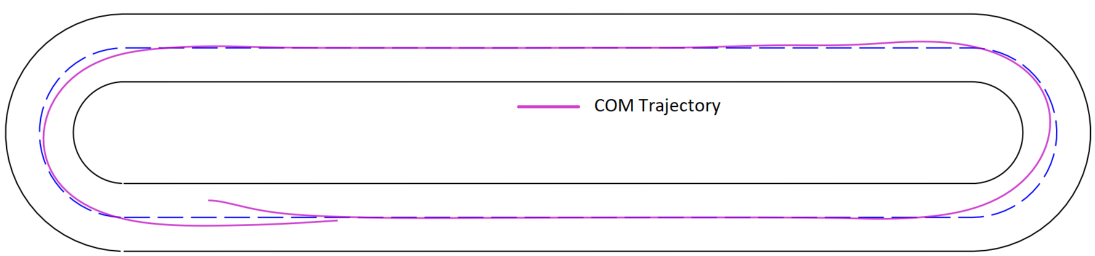


Figure 6: The weights on errors are set as $W_e = \text{diag}(20, 100)$.

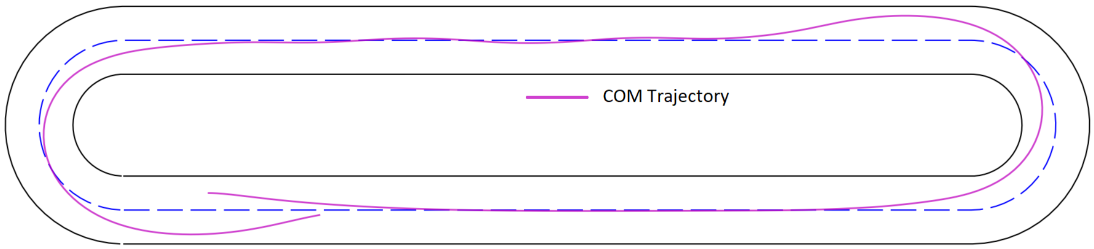


Figure 7: The weights on errors are set as $W_e = \text{diag}(1, 100)$.

The cost on virtual progress velocity is $q_v = 50$ and on the change in input is $R_{\Delta u} = \text{diag}(70, 20, 20)$. When cost on contouring error is 20, figure 6, the error between Centerline (reference trajectory) and the COM trajectory is low compared to when the weight on contouring error is 1, figure 7. By varying this weight it makes it easy to define the objective on tracking quality.

4.3 Obstacle Avoidance

In this section, the obstacle avoidance capability of MPCC is presented. The weights on the errors are $W_e = \text{diag}(5, 100)$. Cost on virtual progress $q_v = 50$. Cost on change in inputs $R_{\Delta u} = \text{diag}(70, 20, 20)$.

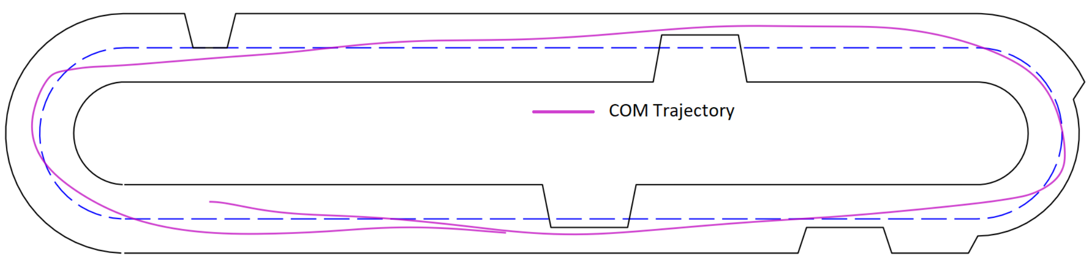


Figure 8: MPCC control of dynamic vehicle on an obstacle course.

5 Conclusion

From section 4, it can be seen that MPCC is an useful approach while considering designing algorithms for autonomous driving. The vehicle is able to complete a lap of the given track without and with obstacles. As the core of the controller depends on an optimization problem, the achieved performance of the controller greatly depends on the weights and prediction horizon chosen. This is evident from sec 4.2 and 4.1, variation in weight cause an significant impact on the performance of the controller. The decision of choosing weights has a learning curve, with trial and error parameters with acceptable performance can be chosen. Improper choice of parameter cause the optimization problem to be stuck in an local optimum which corresponds to undesired motion of the vehicle. Nevertheless, this type of controller reduces the required expertise for designing controller, for dynamic vehicles, by providing a intuitive mapping of tunable parameters to behaviour changes.

References

[LDM14] Alexander Liniger, Alexander Domahidi, and Manfred Morari. "Optimization-based autonomous racing of 1:43 scale RC cars". In: *Optimal Control Applications and Methods* 36.5 (July 2014), pp. 628–647. ISSN: 0143-2087. DOI: 10.1002/oca.2123. URL: http://dx.doi.org/10.1002/oca.2123.

[LMG10] Denise Lam, Chris Manzie, and Malcolm Good. "Model predictive contouring control". In: *49th IEEE Conference on Decision and Control (CDC)*. 2010, pp. 6137–6142. DOI: 10.1109/CDC.2010.5717042.

[Pac12] "Copyright". In: *Tire and Vehicle Dynamics (Third Edition)*. Ed. by Hans B. Pacejka. Third Edition. Oxford: Butterworth-Heinemann, 2012, p. iv. ISBN: 978-0-08-097016-5. DOI: https://doi.org/10.1016/B978-0-08-097016-5.02001-5. URL: https://www.sciencedirect.com/science/article/pii/B9780080970165020015.

[Pom11] Pomax. *A Primer on Bézier Curves*. 2011. URL: https://pomax.github.io/bezierinfo/.

[VFT09] Efstathios Velenis, Emilio Frazzoli, and Panagiotis Tsiotras. "On steady-state cornering equilibria for wheeled vehicles with drift". In: *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*. 2009, pp. 3545–3550. DOI: 10.1109/CDC.2009.5399782.