MPC for Trajectory Planning of Race Cars with Obstacle Avoidance

Henrik Bey, Lukas Brunke, Grischa Gottschalg, Lukas Ortmann

December 18, 2015

Abstract - *A Nonlinear Model Predictive Control technique is used to determine the optimal trajectory on a race track simulation. The optimization problem is constrained by the structure of the race track and by a simple vehicle model, which incorporates limited lateral acceleration. Various cost functions are compared including soft constraints in order to perform obstacle avoidance.*

1 Overview

Planning the optimal trajectory[1] of a race car on a track is a widely discussed topic at control engineering conferences [1, 2, 3]. A common approach to identify the fastest path is Model Predictive Control (MPC).

To properly define this optimization problem, constraints (section 2) and a well defined cost function (section 3) are required. We compare different cost functions with respect to their resulting lap time on a predefined race track. In addition, obstacle avoidance with static and dynamic obstacles on the race track is implemented.

The constraints for the problem are nonlinear. The same applies for most of the cost functions. The used optimizer, `fmincon` in Matlab®, is working with both, nonlinear and linear, constraints.

To visualize the optimization results a simulation of the car on the race track is implemented in Matlab® which shows the closed-loop trajectory as well as the open-loop trajectories of the MPC.

2 Constraints

The first part of the constraints arises from the modeled car dynamics which are represented by a bicycle model. The second part of the constraints guaranties, that the car stays on the track.

[1] The optimal solution is the trajectory that minimizes the lap time.

2.1 Model Constraints

The model dynamic equations are obtained from [3]. Therefore, two consecutively states have to satisfy equation (1). Additionally, there are state constraints, e.g. the maximum speed. Moreover, the model inputs are constrained, by the maximum and minimum acceleration as well as the relative heading angle (steering angle).

$$z_{k+1} = z_k + m(z_k, u_k)\Delta t \qquad \forall k = 0, ..., N-1 \tag{1}$$
$$v_k \leq v_{max} \qquad \forall k = 0, ..., N \tag{2}$$
$$a_{min} \leq a_k \leq a_{max} \qquad \forall k = 0, ..., N \tag{3}$$
$$|\beta_k| \leq \beta_{max} \qquad \forall k = 0, ..., N \tag{4}$$
$$z_0 = \bar{z}_0 \tag{5}$$

In equation (1) $m(z_k, u_k)$ represents the modeled vehicle dynamics with state $z = (x, y, v, \psi)^T$ and input $u = (a, \beta)^T$.

$$\dot{x} = v\cos(\psi + \beta) \tag{6}$$
$$\dot{y} = v\sin(\psi + \beta) \tag{7}$$
$$\dot{v} = a \tag{8}$$
$$\dot{\psi} = \frac{v}{l}\sin\beta \tag{9}$$

This model is sufficient for low speed trajectory planning. However, the independence of the relative heading angle and the car's velocity results in an unrealistic behavior regarding the turning radius in higher speeds. Therefore, the constraints for β are modified in order to limit the lateral acceleration to a fixed value $a_{L,max} = 1.5g$.

$$|\beta_k| \leq \arctan\left(\frac{1}{2}\frac{l/2 \cdot a_{L,max}}{v^2}\right) \quad \forall k = 0, ..., N \tag{10}$$

2.2 Car on Track

The track is defined by a series of points which mark the center line. As long as the distance between the center of the car and the center line is smaller than half of the track width, the car is considered being on the track (figure 1). First, the closest track point to the car center is determined. With the previous and the consecutive points a normal vector to the track tangent is formed. The distance vectors to six check points on the car edges are projected onto the normal vector. All of the projections have to be shorter than half of the track width.

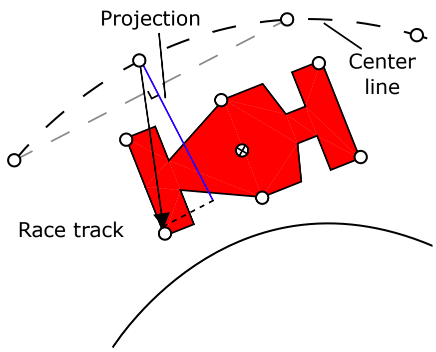

Figure 1: Computing distance to the track

3 Cost Function

Though the goal is to minimize the lap time, finding a cost function is difficult. This is due to the fact, that taking the actual lap time as the cost is not possible, because the time horizon has to be significantly shorter than one complete lap to maintain a reasonable simulation time. Furthermore, the cost function must be efficient in calculation time but still result in a fast lap time. In the following, three different approaches are presented and their performances are analyzed and evaluated. The previously introduced constraints apply for all the different cost functions.

3.1 Speed Maximization

The first approach is to take the negative speed as the cost [4]. In that way the optimizer tries to maximize the speed of the car. There are two possibilities: looking at every point of the horizon or only at single points, e.g. the last one. Summing over the speeds in every step, with a constant time step, leads to the maximization of the covered distance.

The results with this cost function are reasonable and yield a lap time of 25.80 seconds. However, with certain horizons the problem becomes infeasible, because the car accelerates too much before the horizon reaches a turn. Consequently, the car cannot slow down in time to take the turn. Another problem is, that the car tends to take a longer way at a high speed although a shorter path at a lower speed might result in a lower lap time. E.g. right after the start the car starts to meander in order to cover a greater distance.

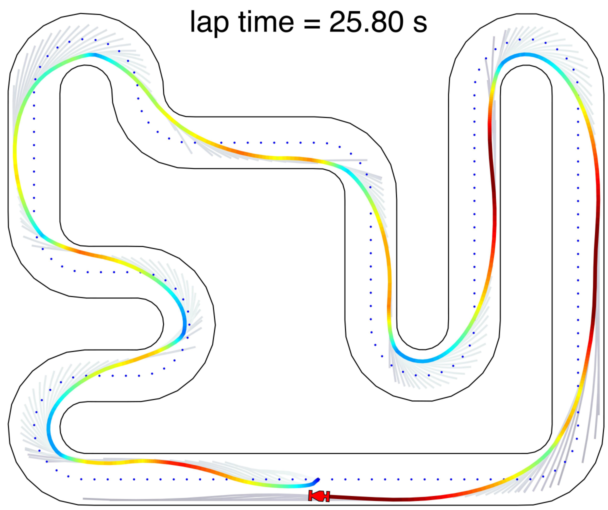

Figure 2: Trajectory of speed maximization[2]

3.2 Cheese Cost

The second approach is to place a goal point at a certain distance in front of the car and to minimize the distance between the last point of the horizon and that goal point. The car tries to reach the "cheese", which is moving ahead from it (see figure 3).

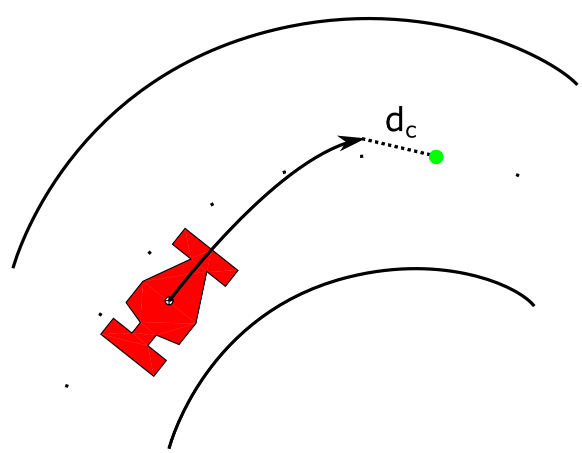

Figure 3: Computing Cheese Cost

The distance between car and cheese d_c is a crucial parameter for this cost function. If it is too short the car will not take the optimal path and only follows

[2]Gray lines are representing the open-loop trajectories of the MPC while the colored line is showing the actual trajectory of the race car (the more red, the higher the velocity of the car).

on the center line. Also if the goal point is within the horizon, the car will not accelerate further, even if it could. If the distance is too long, the car will get stuck in turns when the goal point is already on the other side and the car tries to go directly and does not see the way around the curve.

The lap time with a working parameter is 26.90 seconds (see figure 4).

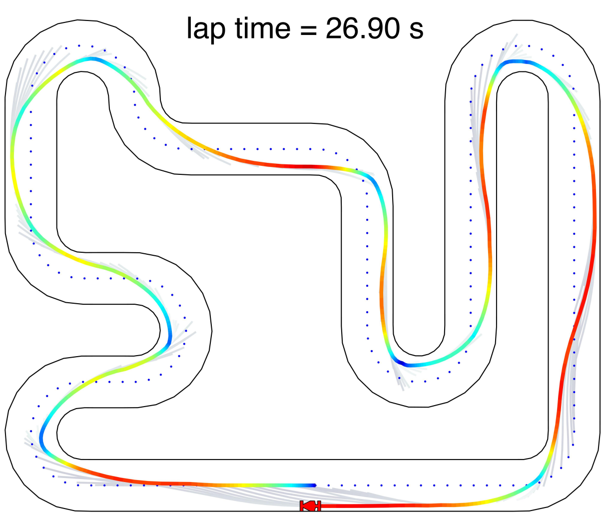

Figure 4: Trajectory of cheese cost

3.3 Pacman Cost

The last approach is intuitive: The car tries to cover as much of the racetrack as possible. This is implemented by maximizing the passed points on the track.[3] The cost function is defined by the number of points between the closest racetrack point to the current position and the closest racetrack point to the last position of the horizon. This is multiplied with the uniform distance between two racetrack points. Since this function is not continuous, which makes it hard to form a gradient, the projection of the vector between the last position and the racetrack point onto the racetrack tangent is added, see figure 5. The result is a good estimate of the actual covered track distance.

Even though the pacman-cost-function resulted in the fastest lap time of 24.60 seconds, the calculated trajectory is not optimal as can be seen in the approaches to some turns in figure 6. This can be explained by the rather short horizon of approximately two seconds which was chosen in order to keep the simulation time

[3]The name is based on the scoring in the computer game *pacman*.

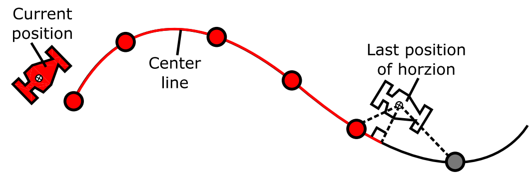

Figure 5: Computing Pacman Cost

acceptable. The pacman cost is also the computationally most intensive cost function.

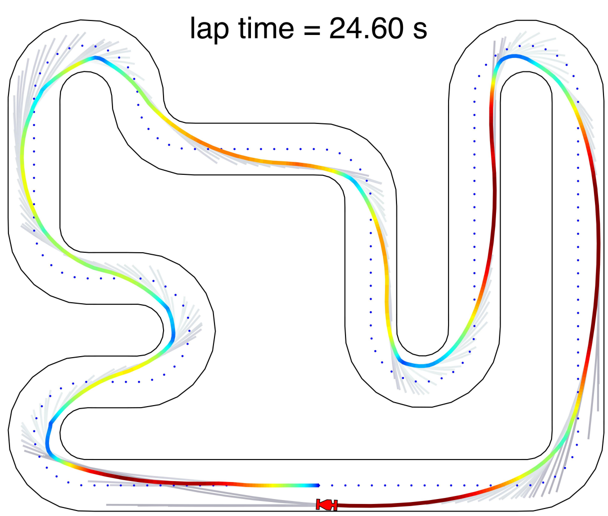

Figure 6: Trajectory of pacman cost

4 Obstacle Avoidance

In order to implement obstacle avoidance in the trajectory planning additional constraints have to be added in the optimization problem. One option is to represent the obstacles as hard constraints, e.g. to include these in the nonlinear constraints of `fmincon` (together with the constraints in section 2). In this paper we decided to rather treat the obstacles as soft constraints, e.g. include them in the cost function of the MPC. The only cost function we used for obstacle avoidance was pacman cost, because with the other approaches `fmincon` is not able to find persistent feasible solutions.

4.1 Fixed Obstacles

For penalizing obstacles on the trajectory as soft constraints a barrier function in the cost function p is necessary. Equation (12) shows the exponential function we used for this purpose.

$$p = p_{pacman} + p_{obs} \tag{11}$$

$$p_{obs} = \sum_{i=1}^{N_o} 100 \cdot \exp\left(-\frac{d - r_a}{0.1}\right) \tag{12}$$

r_a represents the avoided radius around the obstacles and N_o the number of obstacles on the track. The distance d_i between the car and the ith obstacle is determined by an Euclidean norm (Equation (13)).

$$d_i = \|x_{car} - x_{obstacles,i}\|_2 \tag{13}$$

Placing ($N_o = 5$) obstacles on the race track, the MPC calculates the trajectory illustrated in figure 7. The obstacles force the car to change its trajectory and therefore increase the lap time by approximately four seconds.

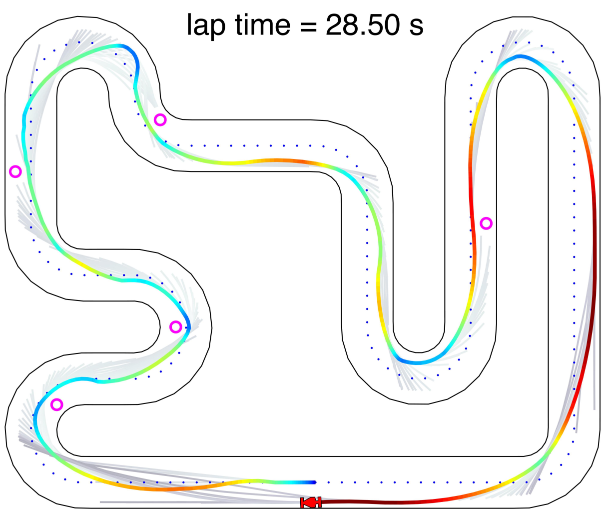

Figure 7: Trajectory of pacman cost with obstacles[4]

4.2 Dynamic Obstacles

The dynamic obstacle is another (green) car, which is driven manually around the course. The trajectory of the green car is far from optimal, so the MPC controlled red car tries to pass, but must not touch the green car. The soft constraints for this case are more complicated than the one for the fixed obstacles. Instead of the Euclidean norm, a weighted sum (over the horizon N, linear decreasing weights g_i) of the penalty function is used with the square of the lateral distance d_l and the square of the tangent distance d_t as the input (equation (14) - (18))[5].

For the calculation of the distances d_l and d_t, the position vector of the cars is denoted by $\boldsymbol{x}$ and the normal vectors in tagent and normal direction by $\boldsymbol{n}_{normal}$ and $\boldsymbol{n}_{tangent}$ respectively (see figure 8).

The car is approximated as an ellipsoid (with the car width w_{car} and the car length l_{car}) which is obtained in equation (16). In that way the rather oblong shape of the moving obstacle (green car) is taken into account. $d_e < 1$ means that the red car and the green car are colliding and is therefore penalized by the exponential function.

$$d_l = (\boldsymbol{x}_{greenCar} - \boldsymbol{x}_{redCar}) \cdot \boldsymbol{n}_{normal} \tag{14}$$

$$d_t = (\boldsymbol{x}_{greenCar} - \boldsymbol{x}_{redCar}) \cdot \boldsymbol{n}_{tangent} \tag{15}$$

$$d_e = \left(\frac{d_l}{w_{car}}\right)^2 + \left(\frac{d_t}{l_{car}}\right)^2 \tag{16}$$

$$p = p_{pacman} + p_{obs} \tag{17}$$

$$p_{obs} = \sum_{i=0}^{N} g_i \cdot \exp\left(-\frac{d_e - 1}{0.01}\right) \tag{18}$$

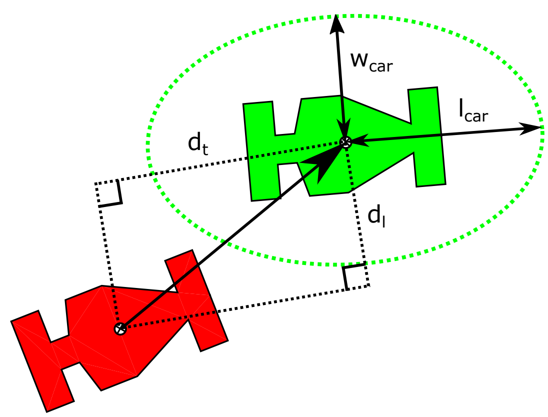

Figure 8: Ellipsoid distance around dynamic obstacle

In addition, the MPC tries to predict where the green car moves, based on its current velocity and heading angle (see figure 9).

The predicted positions are also penalized in the horizon, but with a decreasing weight for future positions, see figure 10.

[4]The magenta rings representing the obstacles in the race track.

[5]In this paper, vectors are denoted by bold letters.

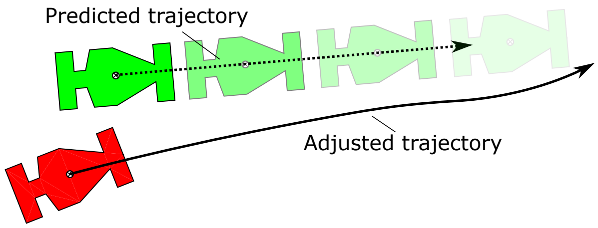

Figure 9: Dynamic obstacle trajectory prediction

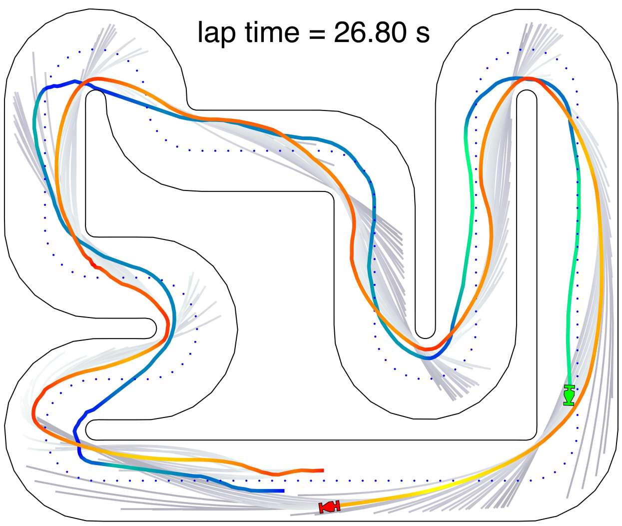

Figure 10: Trajectory of pacman cost vs green car

5 Conclusion

A nonlinear MPC is used to determine the fastest trajectory on a race track. The adjusted bicycle model (with bounded lateral acceleration) forms the constraints together with the bounds of the race track. This results in reasonable car dynamics.

The comparison of the three presented cost functions illustrates the complexity of formulating a proper cost for the optimization problem which is leading to the aimed goal and stays feasible for the used solver `fmincon`. Taking everything into consideration the pacman cost (rewarding the covered distance) is achieving the shortest lap times combined with a persistent feasibility.

Furthermore, the mentioned cost function also yields the best results for obstacle avoidance (soft constraints) with respect to feasibility.

Since we did not give any information about the race track to the MPC, the developed approach will also work on other settings. An evaluation of the performance on different tracks is subject to future work.

As a further outlook we propose to compare the implemented exponential function in the soft constraints of the obstacle avoidance with a logarithmic barrier function. Moreover, integrating more sophisticated car dynamics is of higher interest.

References

[1] Mir Aamir Abbas, Ruth Milman, and J Mikael Eklund. Obstacle avoidance in real time with nonlinear model predictive control of autonomous vehicles. In *Electrical and Computer Engineering (CCECE), 2014 IEEE 27th Canadian Conference on*, pages 1–6. IEEE, 2014.

[2] A. Liniger, A. Domahidi, and M. Morari. Optimization-Based Autonomous Racing of 1:43 Scale RC Cars. *Optimal Control Applications and Methods*, 36(5):628 647, July 2014.

[3] Razvan Solea and Urbano Nunes. Trajectory planning with velocity planner for fully-automated passenger vehicles. In *Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE*, pages 474–480. IEEE, 2006.

[4] E Velenis and P Tsiotras. Minimum time vs maximum exit velocity path optimization during cornering. In *2005 IEEE international symposium on industrial electronics*, pages 355–360, 2005.