MEEN 689 Final Project Report from Shivam Vashi (334003552)

Reimplementing the research by Vittorio Cataffo [1], Giuseppe Silano [2], Luigi Iannelli [1], Vicenç Puig [3], and Luigi Glielmo [1] titled-

A Nonlinear Model Predictive Control Strategy for Autonomous Racing of Scale Vehicles

1. Introduction

The research paper discusses a novel approach to enhancing the control of autonomous racing cars, particularly in the context of the F1/10 Autonomous Racing competition. The primary focus is on utilizing Nonlinear Model Predictive Control (NMPC) to optimize racing car motion while addressing various constraints and challenges. The authors propose a unified NMPC framework that combines trajectory generation, obstacle avoidance, and trajectory tracking to achieve effective and safe control for 1:10 scale racing cars.

The paper highlights the proven effectiveness of the Model Predictive Control (MPC) approach, specifically NMPC, in controlling the motion of autonomous racing cars. NMPC is identified as a suitable solution, especially when the agility of the racing cars plays a crucial role. However, NMPC's capability to incorporate physical constraints comes at the cost of high computational complexity.

Advances in computational capabilities and algorithm efficiency have made it possible to manage this complexity in real-time. The paper points out that various software frameworks have been developed to facilitate modeling, control design, and simulation for NMPC applications over the years. Researchers have explored NMPC strategies, both as trajectory generators and tracking controllers. A common challenge addressed in the paper is the tracking of the lane center line while avoiding collisions with obstacles along the track.

Most commonly, NMPC is used in the outer loop of a cascaded control architecture to provide a reference trajectory for an inner loop tracking controller. This approach is effective but can lead to issues because the NMPC generator does not always consider the limitations of the low-level controller. Consequently, generated trajectories may violate the vehicle's actuator limits, resulting in infeasible solutions.

To overcome these limitations, the paper introduces an innovative NMPC architecture for lane center line tracking in the context of 1:10 scale racing cars. The proposed framework unifies trajectory generation, obstacle avoidance constraints, and trajectory tracking within a single optimization problem. This approach enables tracking of the lane center line while preventing the vehicle from moving outside the track boundaries and simultaneously considering actuator limitations.

In this report, the NMPC Problem is solved using a modified simulation approach and the results are compared with the ones given in the paper. Furthermore, the claim that the optimization problem minimizes lap time is scrutinized.

2. Problem Formulation

A small-scale racing car is required to race along a track minimizing the lap time and remaining within the track boundaries. Meanwhile, the car is required to safely compete avoiding static obstacles populating the track. In addition, the control system is demanded to comply with the vehicle's actuation limits while fulfilling the mission objectives.

The Dynamic Bicycle Model of the F1/10 car is-

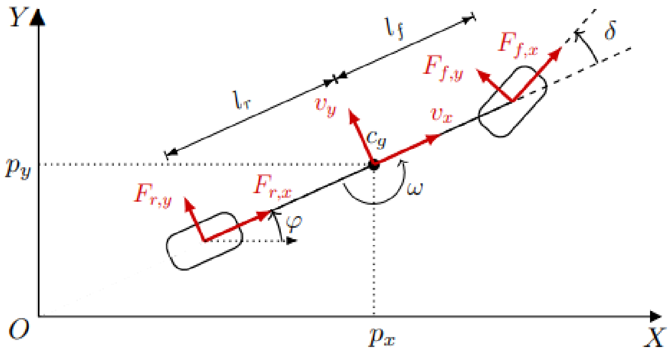

Figure 1. A Representation of the Dynamic Bicycle Model

$$
\begin{aligned}
&\dot{p}_x = v_x \cos\varphi - v_y \sin\varphi \\
&\dot{p}_y = v_x \sin\varphi + v_y \cos\varphi \\
&\dot{\varphi} = \omega \\
&m\dot{v}_x = F_{r,x} - F_{f,y}\sin\delta + F_{f,x}\cos\delta + m v_y \omega' \\
&m\dot{v}_y = F_{r,y} + F_{f,y}\cos\delta + F_{f,x}\sin\delta - m v_x \omega \\
&J_z\dot{\omega} = l_f F_{f,y}\cos\delta + l_f F_{f,x}\sin\delta - l_r F_{r,y}
\end{aligned}
\qquad (1)
$$

The model (1) describes a nonlinear dynamic system $\dot{\mathrm{x}} = f_c(\mathbf{x}, \mathbf{u})$, with state $\mathbf{x} = [p_x, p_y, \varphi, v_x, v_y, \omega]^{\mathrm{T}} \in \mathbb{R}^6$ and control input $\mathbf{u} = [\tilde{d}, \delta]^{\mathrm{T}} \in \mathbb{R}^2$. Where $\tilde{d} \in [0,1]$ is the Pulse Width Modulation (PWM) signal applied to motors ($\tilde{d} = 1$ corresponds to full throttle, while $\tilde{d} = 0$ to full braking).

The lateral forces $F_{f,y}$ and $F_{r,y}$ acting on the vehicle are described using the simplified Pacejka's magic formula-

$$F_{f,y} = D_f \sin\left(C_f\, arctan\left(B_f \alpha_f\right)\right) \qquad (1a)$$

$$F_{r,y} = D_r \sin\left(C_r \arctan\left(B_r \alpha_r\right)\right) \qquad (1b)$$

where B_f and B_r are the stiffness factors, C_f and C_r are the shape factors, and D_f and D_r are the peak factors. The slip angles α_f and α_r are described as-

$$\alpha_f = -arctan\left(\frac{\omega l_f + v_y}{v_x}\right) + \delta \qquad (1c)$$

$$\alpha_r = arctan\left(\frac{\omega l_r - v_y}{v_x}\right) \quad (1d)$$

For ease of modeling, the longitudinal forces $F_{f,x}$ and $F_{r,x}$ are assumed to be equal, i.e., $F_{f,x} = F_{r,x} = F_x$, and described using the following drivetrain model-

$$F_x = (C_{m1} - C_{m2}v_x)\tilde{d} - C_{m3} - C_{m4}v_x^2 \quad (1e)$$

Where C_{m1}, C_{m2}, C_{m3} *and* C_{m4} are empirical parameters used to shape the model's response curve to fit the drivetrain characteristics.

All the constants present in equations (1a) through (1e) were identified by the authors using system identification and the rest of the vehicle parameters were assumed to be known. Table 1 provides all the constants.

Table 1. Model's Parameter Values

Sym.	Value	Sym.	Value	Sym.	Value
l_f	0.178 m	l_r	0.147 m	m	5.692 kg
J_z	0.204 kg m^2	B_f	9.242	B_r	17.716
C_f	0.085	C_r	0.133	D_f	134.585 N
D_r	159.919 N	C_{m1}	20 N	C_{m2}	6.92×10^{-7} kg s^{-1}
C_{m3}	3.99 N	C_{m4}	0.67 kg m^{-1}	M	1674

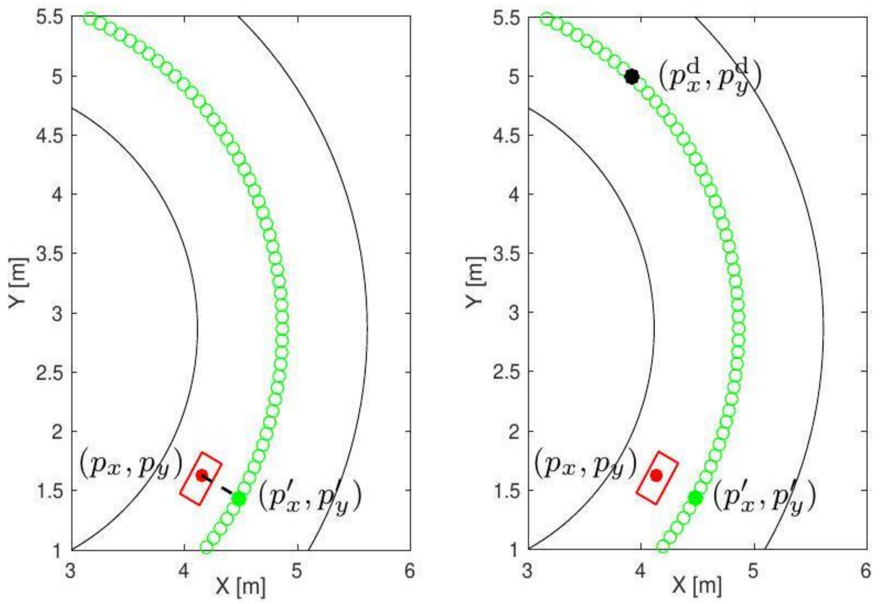

Figure 2. Car's position (p_x, p_y), the car's projection on the lane center line (p'_x, p'_y) and the reference coordinate (p^d_x, p^d_y), P samples looking-ahead w.r.t. the car's center lane projection.

Collision Avoidance Constraint-

The collision avoidance constraint is formulated through the positions of the vehicle $\mathbf{p}$ and the j-th obstacle $\mathbf{p}_j^{\mathrm{o}}$:

$$\|\mathbf{p}-\mathbf{p}_j^{\mathrm{o}}\|^2 \geq \Gamma_j^2 \quad (2)$$

where $\Gamma_j \in \mathbb{R}_{>0}$ is a threshold distance value that the vehicle has to maintain to avoid collisions with obstacles. This is defined accounting for the obstacle sizes and the vehicle's body dimensions to ensure a sufficient room margin for maneuvers while remaining within the track boundaries.

Track Boundary Constraint-

By considering the vehicle's position $\mathbf{p}=\left[p_x, p_y\right]^{\top}$ and its projection on the lane center line $\mathbf{p}'=\left[p_x', p_y'\right]^{\top}$, the track boundary constraint can be formulated as follows-

$$\|\mathbf{p}-\mathbf{p}'\|^2 \leq \left(R_g-R_c\right)^2 \quad (3)$$

where R_g and R_c are defined considering the track width and the car dimensions.

Final Optimization Problem-

Thus, the optimal control problem, with a prediction horizon of $N \in \mathbb{N}_{>0}$ steps, is described as the minimization of the distance between the last predicted vehicle's position $\mathbf{p}_N$ and the reference point coordinates $\mathbf{p}^{\mathrm{d}}$. Therefore, at each time step $t_i=iT_s$, with $i \in \mathbb{N}_{>0}$ and T_s being the sampling time, it can be formulated an optimization problem as follows-

$$\underset{\mathbf{u}}{\operatorname{minimize}} \|\mathbf{p}_N-\mathbf{p}^{\mathrm{d}}\|^2 \mathbf{Q}_1+\sum_{k=0}^{N-1} \|\mathbf{u}_k-\mathbf{u}_{k-1}\|^2 \mathbf{Q}_2 \quad 4(a)$$

such that-

$$\mathbf{u}_{-1}=\mathbf{u}(t_{i-1}) \quad 4(b)$$

$$\mathbf{x}_0=\mathbf{x}(t_i) \quad 4(c)$$

$$\mathbf{x}_{k+1}=f(\mathbf{x}_k, \mathbf{u}_k), k=0, \ldots, N-1 \quad 4(d)$$

$$\|\mathbf{p}_k-\mathbf{p}_{j_k}^{\mathrm{o}}\|^2 \geq \Gamma_j^2, k=0, \ldots, N, j \in\{1, \ldots, O\} \quad 4(e)$$

$$\|\mathbf{p}_k-\mathbf{p}_k'\|^2 \leq \left(R_g-R_c\right)^2, k=0, \ldots, N-1 \quad 4(f)$$

$$\underline{\boldsymbol{\mu}} \leq \mathbf{u}_k \leq \overline{\boldsymbol{\mu}}, k=0, \ldots, N-1 \quad 4(g)$$

$$\underline{\boldsymbol{\gamma}} \leq \mathbf{x}_k \leq \bar{\gamma}, k=0, \ldots, N \quad 4(h)$$

Where (4a) is the objective function with $\mathbf{Q}_1, \mathbf{Q}_2 \in \mathbb{R}^{2\times2}$ being diagonal weighting matrices, $\mathbf{x}_k$ and $\mathbf{u}_k$ are the sampled predicted state and control input, respectively, at the k-th sample of the current MPC interval, (4b) and (4c) initialize the control and the state, (4d) describes the discretized dynamic model for the vehicle (1). (4e) is the collision avoidance constraint taken from (2), (4f) is the track boundary constraint taken from (3), and (4g) and (4h) are the control input and state limits, respectively. The values of all the constants used in the optimization problem are provided in Table 2.

Table 2. Control Parameter Values Used in the Constrained Optimization Problem

Sym.	Value	Sym.	Value	Sym.	Value
P	90	T_s	0.033 s	N	50
R_c	0.24 m	R_g	2 m	Γ	1.5 m
$\mathbf{Q}_1$	diag(10, 10)	$\mathbf{Q}_2$	diag(10, 10)	R_j	1 m
$\underline{\gamma}_{v_x}$	0	$\bar{\gamma}_{v_x}$	5	d_s	0.1 m
$\underline{\mu}$	$(0, -\pi/6)^\top$	$\bar{\mu}$	$(1, \pi/6)^\top$	-	-

3. Literature Review and Solution Approach

3.1 Literature Review

The authors have studied the NMPC strategies undertaken by other researchers to solve the same problem (trajectory generation, obstacle avoidance, and trajectory tracking for minimizing lap time). Various works have investigated NMPC strategies both as a trajectory generator [3], [4] and as a tracking controller [5], [6]. A common problem is the tracking of the lane center line while avoiding collisions with obstacles placed along the track. In most cases, NMPC is used in the outer loop of a cascaded architecture to provide a reference trajectory to an inner loop tracking controller [5], [4]. This approach allows to attain the tracking-lane objectives, but it can cause problems since the NMPC generator does not consider the limitations posed by the low-level controller [7]. As a consequence, the generated trajectory could violate the vehicle's actuator limits resulting in an unfeasible solution.

To overcome this limitation, the authors ideated that NMPC can be used to combine trajectory generation, subject to obstacle avoidance constraints, and trajectory tracking, subject to actuation limits and track boundaries, in a single optimization problem taking inspiration from [8], [9] and [10]. The so-formulated problem (4) allows to keep tracking of the lane center line, preventing critical configurations that could move the vehicle out of the limited-width track, while taking the actuator limitations into account.

3.2 Authors' Solution Approach

The authors have encoded the problem (4) using a single shooting implementation with the track boundaries (3) and obstacle avoidance (2) constraints treated using the augmented Lagrangian and the penalty method approaches, respectively, following the constraints formulation of the OpEn framework (https://alphaville.github.io/optimization-engine) [1], [2]. The vehicle's dynamics were integrated using a Forward Euler integration method with a sampling time of Ts = 33 ms. The prediction horizon considered N = 50 steps, which gave an ahead prediction of 1.65 s.

<u>Authors' Methodology</u>- The optimization solution, at every time step, provides control input values for N=50 steps, from which the first input is applied to the vehicle in the F1-tenth simulator. The vehicle's odometry topic provides the new state of the vehicle and the optimization problem is solved again, the first input is applied to the vehicle just like the previous step. This is repeated atleast until finishing two laps of the race track. Relevant vehicle data is stored at every time step for plotting. This methodology adopted by the authors provided the results shown in

Figure 3 and Figure 4.

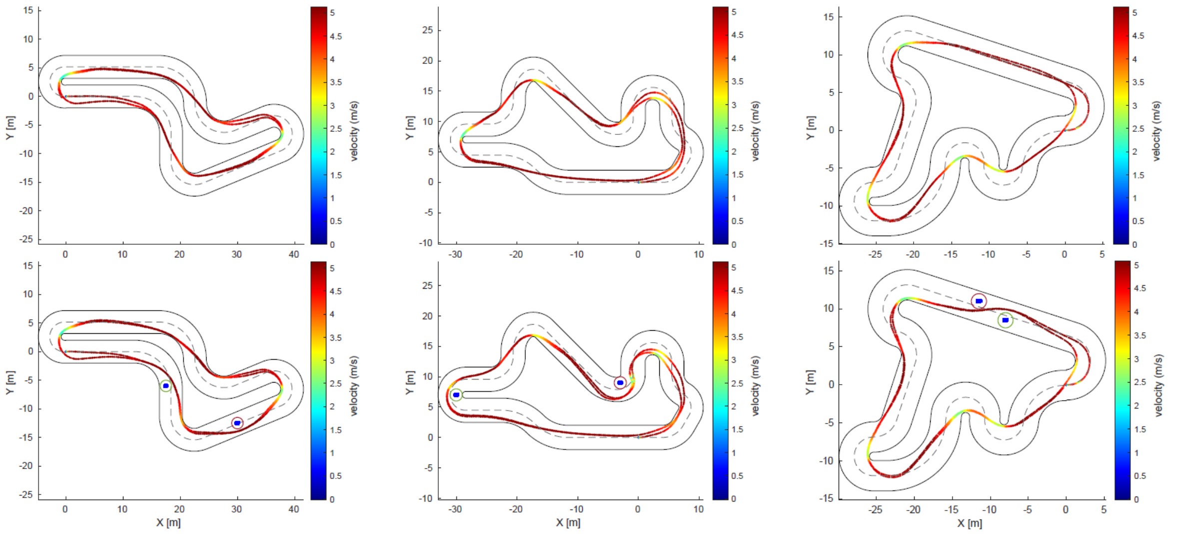

Figure 3. Authors' Simulation Results. The color gradient indicates the car's velocity. Track scenarios with obstacles are at the bottom

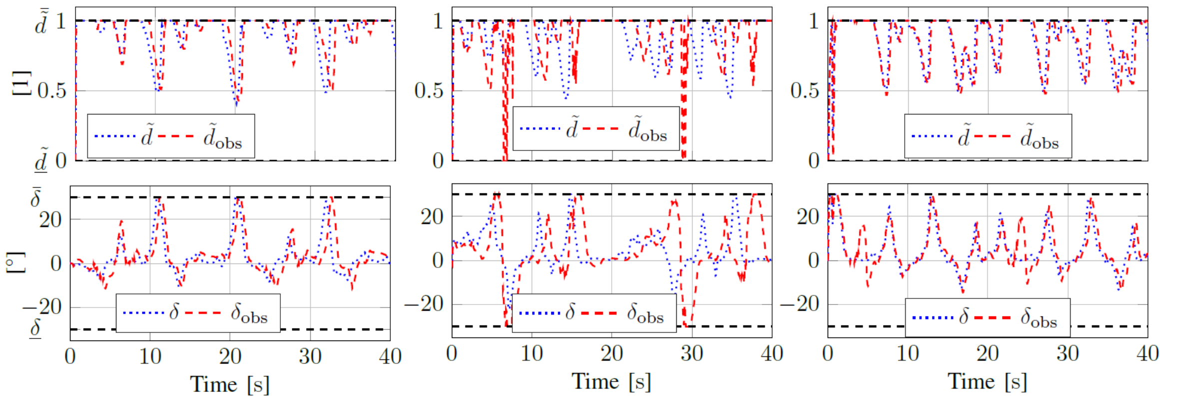

Figure 4. Authors' Control inputs u in cases with $(\tilde{d}_{obs}, \delta_{obs})$ and without $(\tilde{d}, \delta)$ obstacles. The scenarios are in the same order as

Figure 3

3.3 Modified Solution Approach

In order to focus on the NMPC optimization problem and its features, along with the time constraints of project submission, it was decided to avoid using the F1-tenth simulator. The following methodology was implemented using the Forward Euler Method instead of the F1-tenth Simulator used by the authors (authors' methodology provided in 3.2 above).

<u>Modified Methodology</u>- The optimization solution, at every time step, provides control input values for N steps, from which the first input is applied to the vehicle. The vehicle's new state is calculated using the state dynamics equations (1 and 1a through 1e) and the Forward Euler Method $(x_{next} = x_{now} + \Delta T(\dot{x})_{now})$. The optimization problem is solved again with this new state; the first

input is then applied to the vehicle just like the previous step. This is repeated atleast until finishing two laps of the race track. Relevant vehicle data is stored at every time step for plotting.

4. Results of the Modified Solution Approach

In order to recreate the results obtained by the authors, the same value of N = 50 along with identical Model and Control Parameter Values (Table 1 and Table 2) were used that provided the following results (Figure 5 and Figure 6) for Track 1.

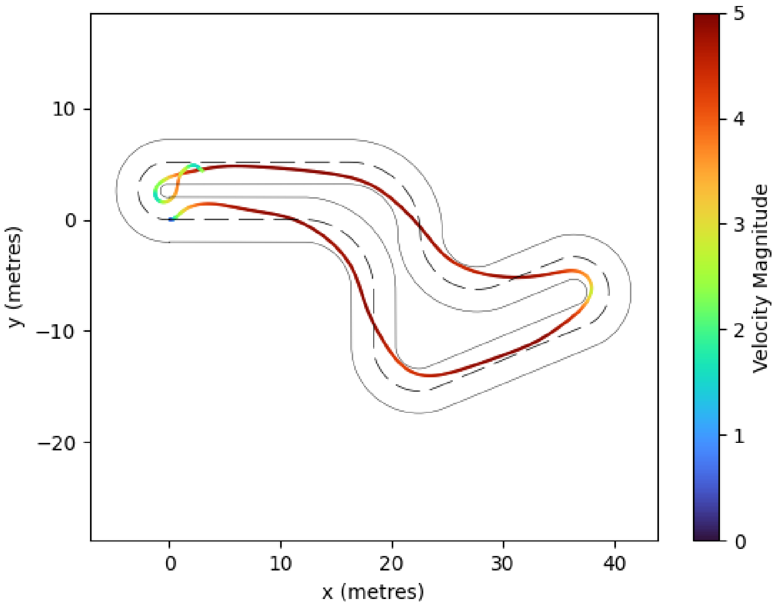

Figure 5. Track 1 Results for N = 50 without obstacles

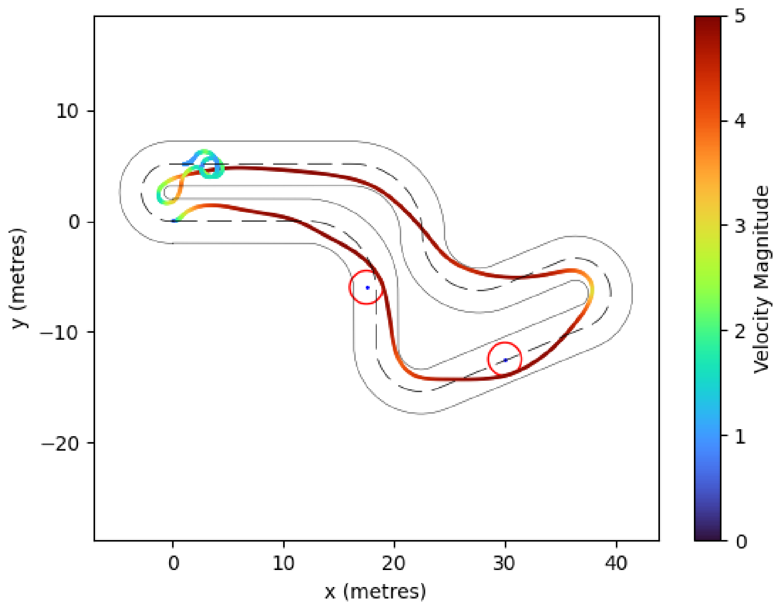

Figure 6. Track 1 Results for N = 50 with obstacles

From the results for N = 50 shown in Figure 5 and Figure 6, it can be observed that when the vehicle starts from (0,0), instead of following a straight line, it takes a longer path unlike the authors' results (

Figure 3). The deviation of the results is due to the difference in the simulation approach. The authors used the F1-tenth Simulator, unlike a simple Forward Euler Method that mathematically evaluates the vehicle states at every time step using the State Dynamics Equations. The F1-tenth Simulator has friction physics built into it which produces complex behaviour of the vehicle for the same input when compared to the mathematical Forward Euler Method. In conclusion, the simulation approach presented in this report is less realistic than the one used by the authors because the Forward Euler Method is a simplified approach for simulating the vehicle. However, since the focus of this project is solving the NMPC Problem and not on simulation, it was decided to carry on with the Forward Euler Method.

The observation that the vehicle is unnecessarily taking a longer curve led to the idea of changing the parameter- N i.e., the planning horizon at each time step. It was hypothesized that decreasing the value of N would improve vehicle's tracking. To test this hypothesis, the value of N was reduced and the plots were compared visually with the authors' results and it was observed that similar results were obtained with the Forward Euler Method for N = 40 as shown in Figure 7 and Figure 8.

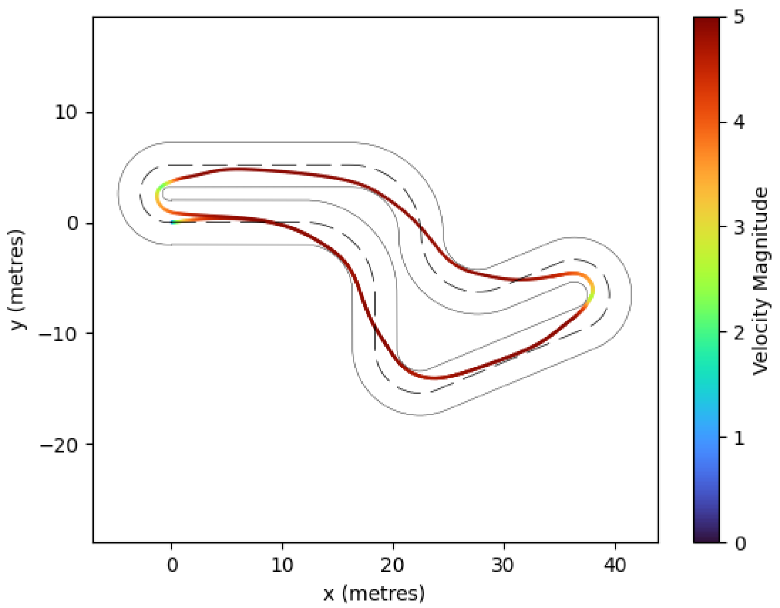

Figure 7. Track 1 Results for N = 40 without obstacles

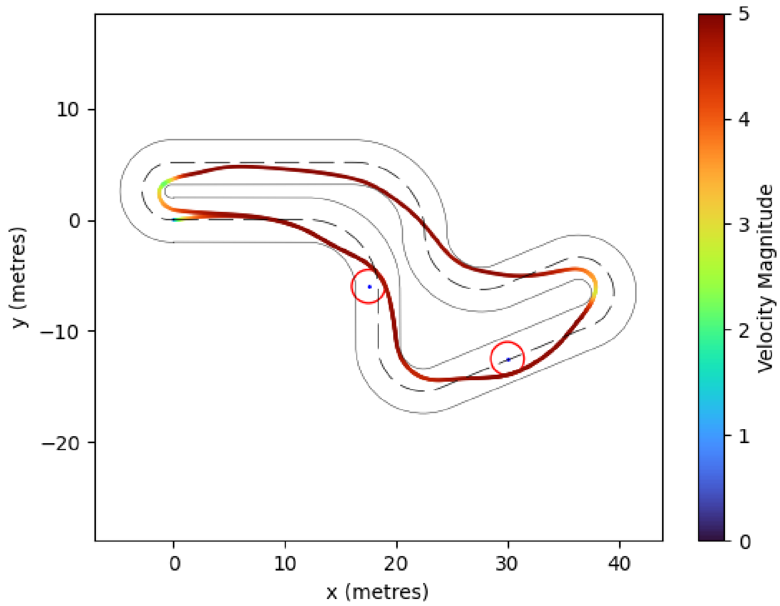

Figure 8. Track 1 Results for N = 40 with obstacles

One of the objectives of the project was to examine the authors' claim that the solution to the optimization problem minimizes the lap time. To do this, the lap time was quantified as the time it takes for the vehicle to complete two laps; after the first two laps, the vehicle will just repeat the second lap plan, so it is sufficient to time the first two laps. While changing the values of N, lap time was also noted. The data as shown in Table 3 was obtained.

Table 3. Lap Time Data

N	Lap Time					
	Track 1		Track 2		Track 3	
	w/o obstacles	w obstacles	w/o obstacles	w obstacles	w/o obstacles	w obstacles
44	No Solution	No Solution	40.425	43.494	38.346	38.676
43	No Solution	No Solution	40.326	43.428	38.214	38.544
42	41.349	42.108	40.26	43.362	No Solution	No Solution
41	41.382	42.141	40.194	43.263	37.95	38.247
40	41.415	42.174	40.161	43.23	37.917	38.214
39	41.448	42.207	41.448	43.164	37.884	38.181
38	41.514	42.273	40.095	43.164	37.851	38.148
37	41.547	42.306	40.095	No Solution	37.884	38.214
36	41.613	42.339	40.095	49.104	37.884	38.214
35	41.646	42.372	40.095	No Solution	37.818	38.148
34	41.712	42.405	40.095	No Solution	37.785	38.181
33	41.745	42.438	40.095	No Solution	37.818	38.181
32	41.778	42.504	40.128	No Solution	37.785	38.181
31	41.844	42.537	40.161	No Solution	37.818	38.247

Here, "No Solution" represents scenarios where the vehicle failed to obey the racetrack and/or obstacle constraints or returned an absurdly curvy path at some part of the tracks similar to what was obtained for N = 50 (Figure 5 and Figure 6).

Note that for N < 31 and N > 44, the solutions for many scenarios (different tracks and presence or absence of obstacles) were absurd and are thus excluded from the analysis. The tabular data was visualized for better understanding as shown hereafter.

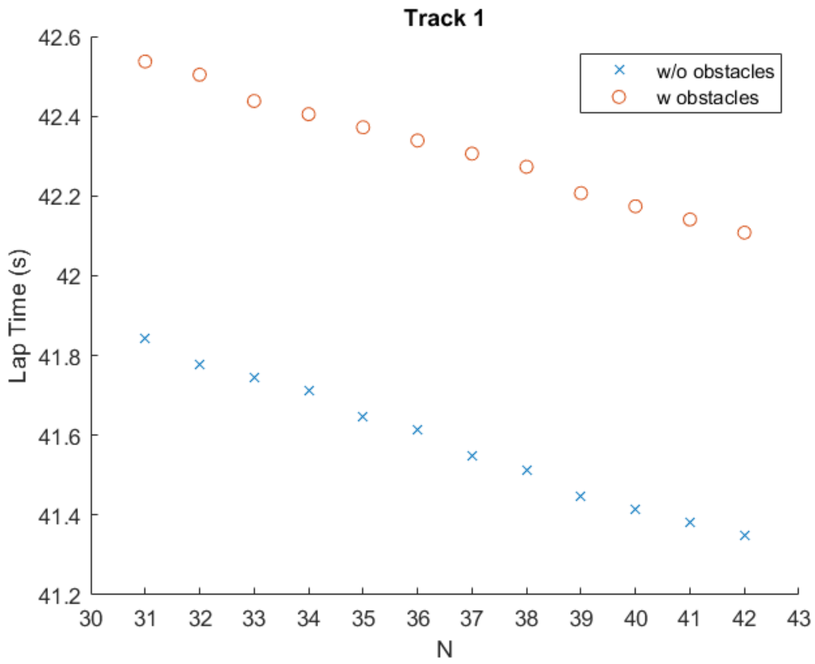

Figure 9. Lap Time Data for Track 1

For Track 1 data as shown in Figure 9, The lap time decreases as N increases but No Solution for N>42.

Minimum lap time for Track 1 is obtained for N = 42

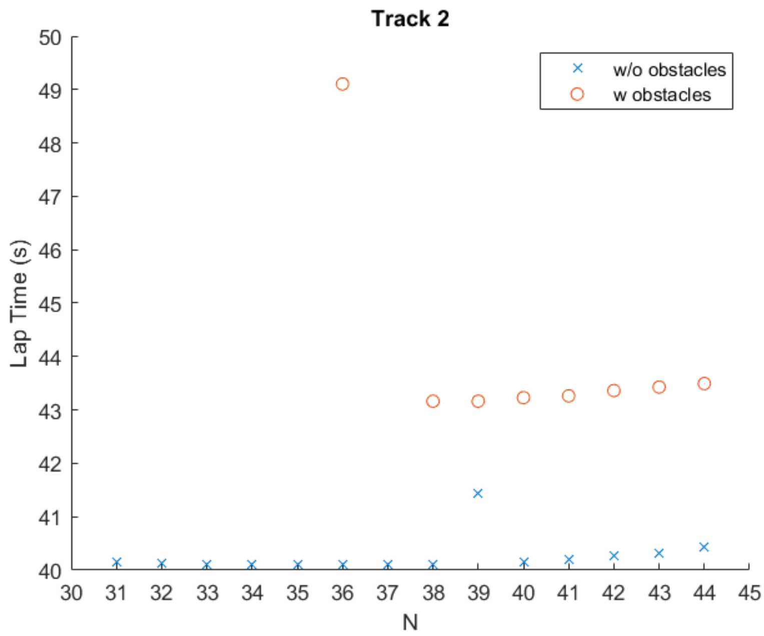

Figure 10. Lap Time Data for Track 2

For Track 2 data as shown in Figure 10, the following observations are noted-

In the presence of obstacles:

The Solver returned no solution for N = 35 and N = 37. The lap time for N = 36 was 49.104s which is inconsistently higher compared to values of N >= 38. Minimum lap time of 43.164s is obtained for N=38 and 39.

In the absence of obstacles:

Minimum lap time of 40.095s is obtained for N = 35, 36, 37, and 38

For Track 2, the value of N that minimizes lap time simultaneously in both the presence and absence of obstacles is N = 38

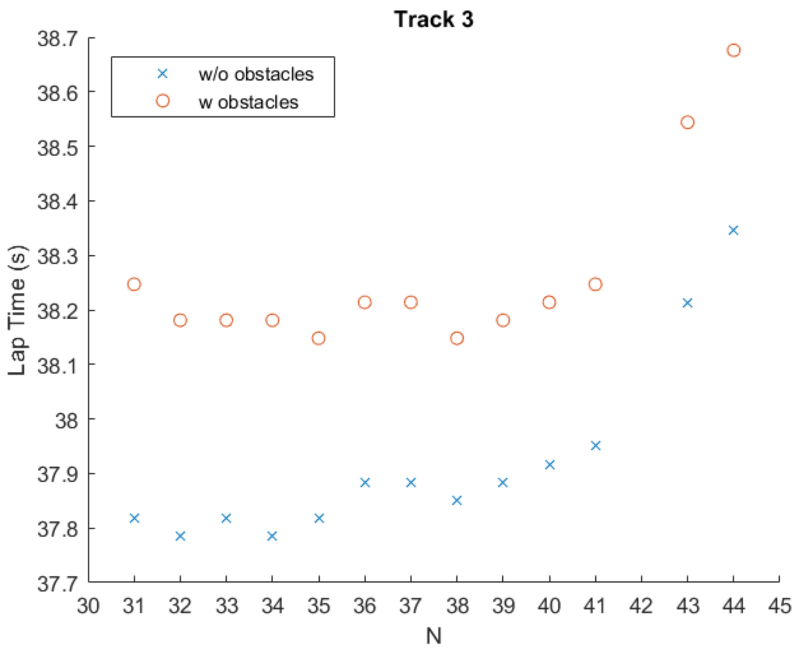

Figure 11. Lap Time Data for Track 3

For Track 3 data as shown in Figure 11, the following observations are noted-

In the presence of obstacles:

The Solver returned No Solution for N = 42. A minimum lap time of 38.148s is obtained for N = 35 and 38.

In the absence of obstacles:

The Solver returned no solution for N = 42. Minimum lap time of 37.785s is obtained for N = 32 and N = 34

There is not a unique value of N that minimizes lap time for both the obstacle-filled track and the empty track simultaneously. But the best choices would be N = 32, 34 or 35

Conclusion for Lap Time Data Analysis-

Based on all the lap time data, it can be concluded that there is not a unique best value of N for which lap time minimization can be guaranteed across all tracks. The range that could provide a relatively smaller lap time for the given racetracks and the given vehicle model is observed to be 32<= N <=42. For further analysis, N = 38 is selected as it provides small lap times without returning absurd solutions for all tracks agnostic to the presence of obstacles.

The analysis has provided key insights into lap time minimization that the authors failed to elicit in the paper. Lap time minimization can be achieved for a given race rack and a given vehicle model by varying the value of N, but it is not a default outcome of the optimization problem designed by the authors.

The following are the results obtained for N = 38 using the Modified Solution Approach (explained in section 3.3)

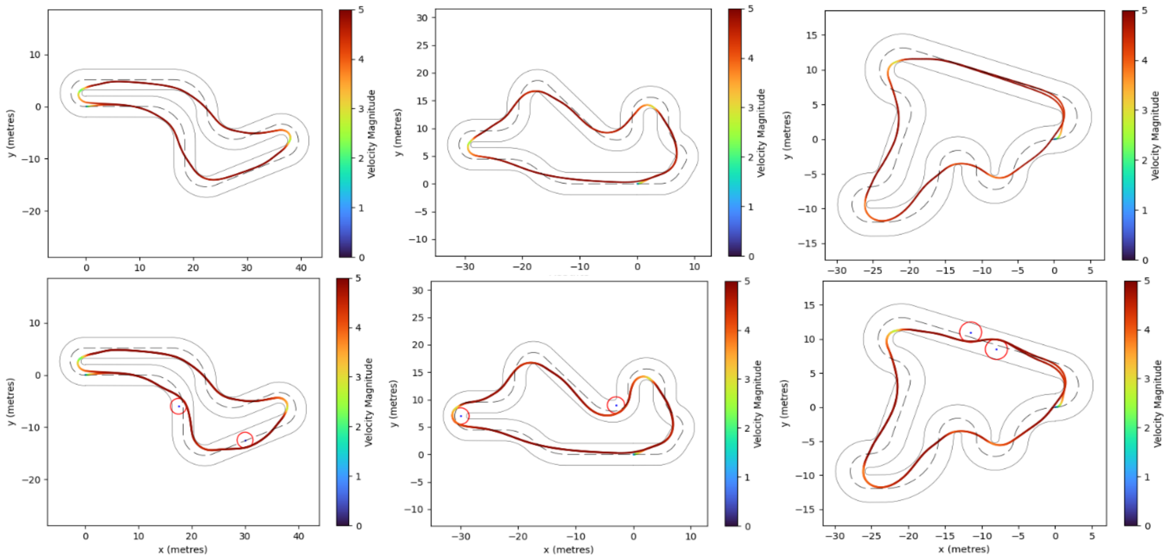

Figure 12. Simulation Results for N = 38. The color gradient indicates the car's velocity. Track scenarios with obstacles are at the bottom

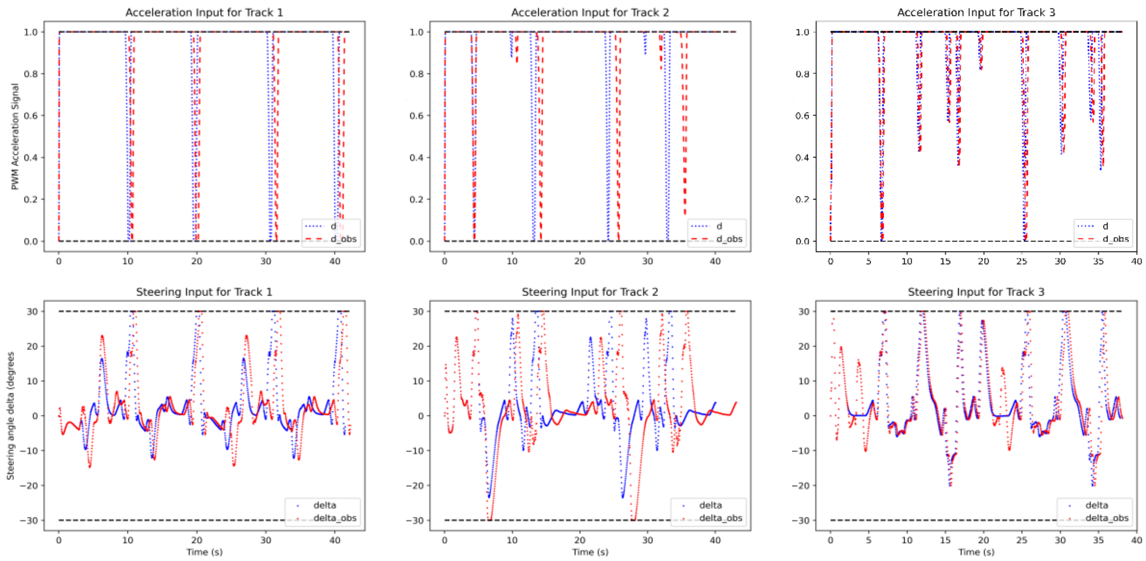

Figure 13. Control inputs u in cases with (d_obs, delta_obs) and without (d, delta) obstacles. The scenarios are in the same order as Figure 12

The results shown in Figure 12 and Figure 13 are similar to the ones obtained by the authors shown in

Figure 3 and Figure 4. The race track and obstacle avoidance constraints are obeyed and the control inputs are bounded as desired (plotted as black dotted lines in Figure 13). The minimum velocity of just under 3 m/s as shown in Table 4 is concordant with the authors' results.

Table 4. Minimum Velocity within each Scenario

Minimum Velocity (m/s)					
Track 1		Track 2		Track 3	
w obstacles	w/o obstacles	w obstacles	w/o obstacles	w obstacles	w/o obstacles
2.124	2.123	2.929	2.507	2.716	2.924

5. Conclusions

- The results from the paper were successfully replicated for a different value of horizon length N = 38 as opposed to the authors' value of N = 50. The difference stemmed from the use of a Forward Euler Method to simulate the vehicle unlike the F1-tenth Simulator used by the authors.

- Authors' claim regarding lap time minimization was challenged. It was realized that the lap time can be minimized for a given race rack and a given vehicle model by varying the value of N, but it is not a default outcome of the optimization problem designed by the authors. Further modification to the optimization problem is needed to guarantee lap time optimality.

- Solving the lap time minimization problem would be useful in multiple applications requiring the fastest policy. Autonomous fail-safe algorithms to avoid accidents, autonomous emergency services (ambulances, fire brigades, etc.) and traffic management for autonomous vehicle fleet are among many such applications.

- Thus, the authors' NMPC Problem was successfully reimplemented with an improved understanding of lap time minimization.

References

[1] A. Sathya et al., “Embedded nonlinear model predictive control for obstacle avoidance using PANOC,” in European Control Conference, 2018, pp. 1523–1528.

[2] S. Pantelis et al., “OpEn: Code Generation for Embedded Nonconvex Optimization,” in IFAC-PapersOnLine, vol. 53, no. 2, 2020, pp. 6548 – 6554.

[3] E. Alcal´a et al., “LPV-MP planning for autonomous racing vehicles considering obstacles,” Robotics and Autonomous Systems, vol. 124, p. 103392, 2020

[4] J. V. Frasch et al., “An auto-generated nonlinear MPC algorithm for real-time obstacle avoidance of ground vehicles,” in European Control Conference, 2013, pp. 4136–4141.

[5] ——, “Autonomous racing using Linear Parameter Varying-Model Predictive Control (LPV-MPC),” Control Engineering Practice, vol. 95, p. 104270, 2020.

[6] R. Verschueren et al., “Time-optimal race car driving using an online exact hessian based nonlinear MPC algorithm,” in European Control Conference, 2016, pp. 141–147.

[7] ——, “Learning How to Autonomously Race a Car: A Predictive Control Approach,” IEEE Transactions on Control Systems Technology, vol. 28, no. 6, pp. 2713–2719, 2020.

[8] U. Rosolia et al., “Autonomous Vehicle Control: A Nonconvex Approach for Obstacle Avoidance,” IEEE Transactions on Control Systems Technology, vol. 25, no. 2, pp. 469–484, 2017.

[9] H. Guo et al., “Simultaneous Trajectory Planning and Tracking Using an MPC Method for Cyber-Physical Systems: A Case Study of Obstacle Avoidance for an Intelligent Vehicle,” IEEE Transactions on Industrial Informatics, vol. 14, no. 9, pp. 4273–4283, 2018.

[10] C. Guoying et al., “Design and experimental evaluation of an efficient MPC-based lateral motion controller considering path preview for autonomous vehicles,” Control Engineering Practice, p. 105164, 2022.