A Data-Driven Predictive Control Driver for Racing Car Simulation

Ruslan Shaiakhmetov
Department of Computer Science and Engineering
Alma Mater Studiorum—Università di Bologna
Cesena, Italy
0009-0001-0869-6204

Danilo Pianini
Department of Computer Science and Engineering
Alma Mater Studiorum—Università di Bologna
Cesena, Italy
0000-0002-8392-5409

Valter Venusti
Digital Innovation and Vehicle Electronics Solutions
Dallara Automobili S.p.A.
Parma, Italy
0009-0001-0647-3015

Alessandro V. Papadopoulos
Division of Intelligent Future Technologies
Mälardalen University
Västerås, Sweden
0000-0002-1364-8127

***Abstract*—The capability to accurately simulate the behavior of a racing car is paramount in modern-day racing competitions to quickly find a good base setup to kick-start the work on track. Typically, a professional driver is employed to drive the simulated race car and provide feedback. However, this operation is expensive and time-consuming, as capable human drivers quickly become a bottleneck. In conjunction with highly accurate simulations of the physical car's behavior, a capable virtual driver could thus accelerate the car setup and development to a great extent. In this paper, we propose to apply a data-driven predictive control approach called Data-enabled Predictive Control to model a racing driver by tracking a pre-defined trajectory. We compare our proposed approach with an industrial first-choice Proportional-Integral-Derivative controller and state-of-the-art Model Predictive Control controller, finding that the approach is feasible, and it can provide significant improvements over the state-of-the-art, especially for trajectories whose feasibility is at the edge of the car's capabilities.**

***Index Terms*—Machine Learning, Model Predictive Control, Autonomous Racing, DeePC, Trajectory Tracking, Racecar Simulation**

I. Introduction

The relevance of simulation is motorsport racing has been growing steadily over the years, with teams investing millions of dollars to get ahead of their competitors[1], as better simulation infrastructures and a high correlation between the simulator and track can lead to significant gains: the base car setup can be found early, shrinking the search space and thus more effectively exploiting the free practice sessions to refine details. Simulation at a professional level can be distinguished into two main categories: 1) *Driver-in-Loop (DiL)* simulators, where a human driver sits in a cockpit and drives the car; and 2) *Virtual driver* simulators, where the car is driven by a computer program.

a) Problem statement: in this work, we focus on the virtual driver simulator case and, specifically, on the construction of a control system capable of tracking an ideal trajectory (e.g., coming from a real-world lap), for which we propose a data-driven predictive control approach. We do not consider the problem of generating the ideal trajectory provided the track limits and the car data. We compare our proposed approach with the current state of the practice (a Proportional-Integral-Derivative (PID) controller) and with Model Predictive Control (MPC).

b) Contribution: we 1) investigate the applicability of a data-driven, learning-based variant of MPC to the problem of trajectory tracking for race car simulation, showing that it is viable; 2) provide an implementation of such controller; and 3) compare it to a PID controller and a "classic" MPC controller on a real-world trajectory, showing competitive performance, especially when the trajectory is hard to follow.

II. Related Work

The problem of tracking a reference trajectory is a common control challenge, requiring the design of a suitable control algorithm to compute automatically the control inputs necessary to minimize the discrepancy between the optimal trajectory and the actual trajectory of the vehicle. Specifically, the control algorithm relies on feedback mechanisms to adjust the vehicle inputs in real-time. A standard approach to trajectory tracking in virtual driver simulations and autonomous driving leverages PID controllers [1]. However,they do not consider the impact

This work is supported by the project "Ottimizzazione computazione di un modello di simulazione dinamica del veicolo" funded by the European Union - NextGenerationEU and Dallara Automobili S.p.A. through the Italian "National Recovery and Resilience Plan" (PNRR) Mission 4, Component 2, Investment 3.3 (DM 352/2022) - CUP J33C22001400009 and the Swedish Research Council (VR) with the PSI project (No. #2020-05094)

[1]AMG Petronas Formula One Team—How Does F1 Simulation Work? https://archive.is/mrNjx, archived on 2024-04-02

of current decisions on the immediate future as a human operator would instead.

Instead, MPC approaches can anticipate future states based on a model of the system dynamics [2], and they are increasingly being used in autonomous driving [3], resulting in smoother, more efficient driving performance [4] at the cost of increased computational complexity, as MPC requires a detailed model of the system dynamics to make informed predictions, whose construction and calibration can be time-consuming and resource-intensive [5].

To address these limitations, some approaches try to *learn* the underlying system *implicitly* from data, using statistical models (e.g., NARX [6]) or deep neural networks [7]. Data-enabled Predictive Control (DeePC) [8] is part of this family. DeePC has shown promising results, particularly in handling the nonlinear dynamics (inherent in racing environments) [9] and, with sufficiently large datasets in adapting to complex and dynamic scenarios [10].

III. Proposed approach

In this work, we compare DeePC and "classic" MPC with the current state-of-the-practice PID controller to control a simulated racing vehicle. The main issue with PID controllers is that they are reactive and not predictive; thus, they cannot adjust in advance based on a future expectation of the car state. On the other hand, "classic" MPC approaches require a detailed model of the system dynamics, which, in many cases, can be hugely complex (due to their implementation using internally Finite element model (FEM), Multibody dynamics (MBD), etc). With DeePC, instead, the model is learned based on experience, relieving the designer from the daunting task of building it. Additionally, the approach can be valuable where an unknown reality gap exists, which is a common situation in high-performance racing cars[2].

A. DeePC

DeePC is a data-driven variation of MPC where the explicit model is substituted by a dataset. The intuition behind DeePC is that a linear combination of past realizations(trajectories) of the system could be used to match the current conditions and forecast future behavior. The algorithm can be configured with two key parameters. The first is the size of the dataset N. We expect larger datasets to provide faster convergence and better overall performance at the expense of increased computational complexity. A second important parameter is the prediction horizon ph, which determines how far into the future the controller should look, and it is equivalent to the length of future input and output traces. The selection of an appropriate prediction horizon is important for the performance of the controller, as we expect it to be unable to predict enough far in the future for very short horizons and to slowly degrade its performance for very long horizons, as the learned behavior will slowly drift away from reality.

B. DeePC for simulated racing car control

In this section, we introduce an instance of DeePC applied to the control of a simulated racing car.

a) Throttle and brake: Acceleration control ranges from -1 (maximum breaking) to 1 (full throttle). Combined brake/throttle actions and gearshift are not captured; although they can be included straightforwardly in an extended model, as the input and output spaces are not constrained by DeePC.

1) Steering: Steering is modeled as the angle between the centerline of the vehicle and centerline of the front wheels, ranging in $[-15°, +15°]$.

2) Outputs: The model outputs are: (i) the horizontal and (ii) vertical coordinates of vehicle's center of mass with respect to a global reference point (m) (iii) the speed ($^m/_s$) and (iv) the heading angle with respect to a global coordinate directed towards east (rad).

3) Dataset generation: DeePC requires a dataset of past realizations of the system to be able to make predictions. Thus, we run multiple random instances of the racecar model changing the seed, and record the car's responses to inputs.

IV. Experimental evaluation

In this section, we investigate how DeePC compares to "classic" MPC and traditional PID controllers in terms of tracking performance. Our experiment is meant to compare the performance on a real-world trajectory of a race car to gather evidence of whether the approach can be applied to realistic scenarios.

A. Experimental setup

To determine how accurately a vehicle follows a predetermined path using a specific control system, we measure the Residual Sum of Squares (RSS) of the distances between the reference trajectory and the actual path taken by the vehicle. To push the control algorithms to their limit, we make the lap increasingly more difficult to follow by varying the peak friction coefficient of the tires, emulating the effect of an increasingly slippery track[3]. We analyze conditions that range from a slippery track in which the target lap is plainly not achievable ($D_{RF} = 0.8$), to conditions of extremely high grip ($D_{RF} = 1.6$).

[2]It is often referred to as "correlation" between simulated and real-world data and it is paramount in modern high-level racing competitions. See, for instance: hhttps://archive.is/2E4QK, https://archive.is/zTLpj, and https://archive.is/hy34O(archived 2024-05-24).

[3]Track conditions impact significantly the grip available to the car in real-world racing. Even if the track remains dry, the rubber laid down by the cars can make it more or less slippery, and the track temperature can also affect the grip. Moreover, the tire compound can be chosen to be more or less grippy, typically presenting a trade-off between peak performance and durability.

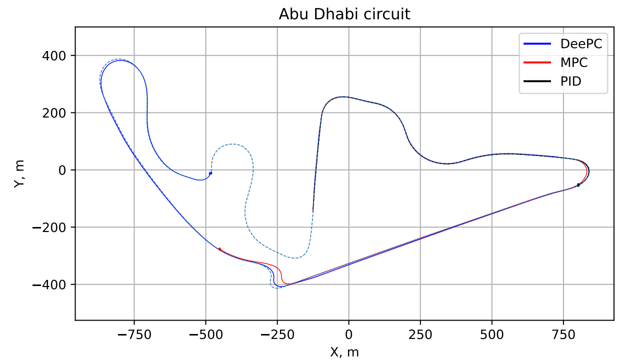

Fig. 1. Trajectory tracking in the Yas Marina circuit in Abu Dhabi. The dashed line defines the target trajectory, solid lines the trajectories as followed by the controllers.

TABLE I
PARAMETERS OF THE RACECAR MODEL

Parameter	Description	Value
D_{RF}	Peak value of the friction coefficient (Pacejika D)	1.0
C_{RF}	Shape factor of the friction coefficient (Pacejika C)	1.1
B_{RF}	Stiffness factor of the friction coefficient (Pacejika B)	25.0
m	Racecar mass	896 kg
I_z	Moment of inertia	1500 $kg \cdot m^2$
l_{RF}	Distance from Center of mass (CM) of Rear/Front wheel	1.125 m
ρ	Air density	1.225 kg/m^3
A	Cross sectional area	
C_D	Drag coefficient	$C_D A = 1.35m^2$
C_L	Downforce lift coefficient	$C_L A = 4.31m^2$
P	Motor power	620 hp

1) Reference trajectory: We test our algorithms on a real-world trajectory of an open-wheel racecar featuring high- and low-speed corners, straights, and chicanes, namely, the Yas Marina circuit in Abu Dhabi, a depiction of which can be seen in Figure 1, where several major racing championships take place, including Formula 1.

2) Racecar model: For this initial study, we use a simplified model of a racecar based on the well-known "single-track model", in which the car is approximated as a two-wheeled vehicle with a fixed wheelbase and the motion is described in terms of a few key parameters summarized in Table I. In the model, we consider nonlinear tire forces based on a simplified Pacejka model [11]. For the experiments, we set the parameters of the car model to mimic the race car the original trajectory was recorded from[4].

B. Baselines

We compare DeePC with two baselines: 1) a PID controller, and 2) a "classic" MPC controller.

[4]The parameters have been extrapolated with the help of domain experts from the data available at https://archive.fo/3erTM

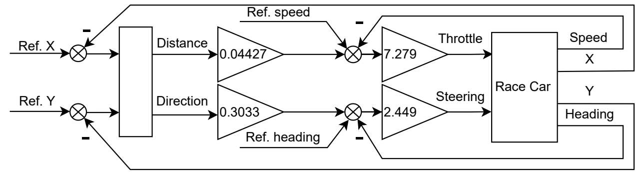

Fig. 2. PID controller scheme.

As PID controller, we employ the cascade scheme of four proportional controllers depicted in Figure 2. The inner loops are responsible for controlling the steering and throttle based on the reference speed and heading. The outer loops integrate these inner loops and are responsible for direction and distance control. The parameters for these controllers have been found using a Random search optimization [12] process, balancing stability and tracking performance.

The MPC controller, instead, is based on a simpler *kinematic* model of the car, which does not consider friction. This way, we replicate the imperfect correlation between the real world and its simulated model[5]. We set the prediction horizon to $60ms$ for both MPC and DeePC.

1) Reproducibility: The implementation of DeePC has been open sourced[6] for reuse and reproducibility. An archival copy is also available at Zenodo [13]. The trajectory could not be included in the repository as it is proprietary.

C. Results

The experiment results are depicted in Figure 4, and show that with complex trajectories featuring slow and fast turns, DeePC outperforms both baselines significantly. In fact, the simplified kinematic model of MPC is not capable to accurately predict the car's behavior in many conditions, while the data-based model internal to DeePC can produce more accurate predictions. Moreover, we found that the classic MPC approach is more sensitive to the precision of the numerical solver than DeePC: we encountered cases in which the behavior resulting from the MPC control is unstable, while DeePC, instead, is capable to follow the same trajectory. An example is shown in Figure 3, in which MPC accelerates too early and loses control of the car after a classic "pendulum" effect. Although the PID controller can achieve good performance for "very simple" trajectories (namely, with extremely grippy tires), it cannot compete in the most relevant conditions, those in which the car is close to its performance limits.

[5]Feeding MPC the same model of the car used for the evaluation would have produce perfect control, as if the controller were able to behave as an oracle. In reality (and in our experiments), the model is never perfect.

[6]https://github.com/Crylab/DeePC

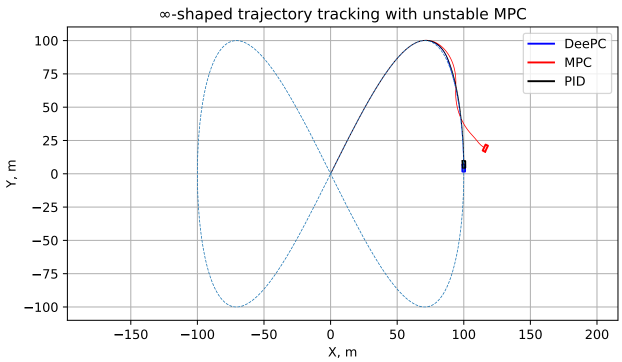

Fig. 3. ∞-shaped trajectory tracking with unstable MPC

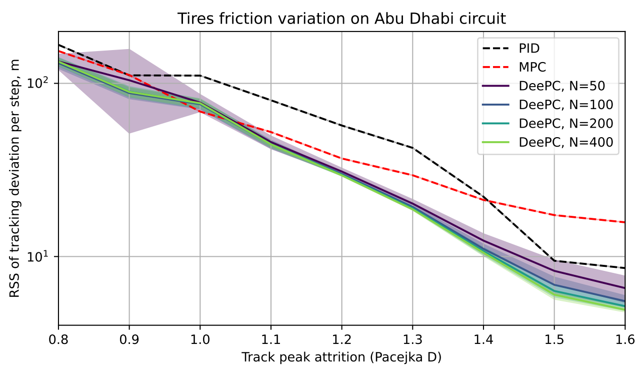

Fig. 4. Error with increasingly higher peak tyre friction in the Yas Marina circuit. DeePC outperforms both the baselines. Solid lines are averaged over 30 repetitions, colored shades areas depict $\pm$ one standard deviation.

V. Conclusion and future work

In this paper, we investigated the applicability of DeePC to the problem of trajectory tracking in the context of race car simulation, comparing it with a traditional PID controller and a "classic" MPC controller on a real-world trajectory tracking problem.

We have found that the approach is viable, and that it can outperform traditional controllers, especially in the most interesting conditions, namely, on complex tracks and with lap times that are on the verge of the car's capabilities. Additionally, its data-driven nature makes it suitable for problems where building an accurate model of the real world is difficult: no matter how complex the control object is, if the dataset is rich enough and the prediction horizon is balanced in such a way that predictions are sufficiently long without losing too much accuracy, then DeePC can be a viable solution.

In future work, we plan to study how the prediction horizon and the dataset size of DeePC influence its performance. Then, we will apply the approach to a state-of-the-art car model, including multibody dynamics, finite element models of tires, and precise aerodynamic effects. Moreover, we expect that a more sophisticated dataset generation process could generate better results with smaller datasets compared to the random control excitation from random initial conditions used in this work. From the point of view of the control algorithm, DeePC uses a linear combination of the trajectories in the dataset to produce predictions. In principle, other strategies, including non-linear combinations, could be investigated. Finally, tackling the whole problem of building a virtual driver requires the ability to generate trajectories based on the characteristics of the track and the car, an issue orthogonal to trajectory tracking, which we plan to address in future work.

References

[1] W. Farag, "Complex trajectory tracking using PID control for autonomous driving," *International Journal of Intelligent Transportation Systems Research*, vol. 18, no. 2, pp. 356–366, 2019.

[2] M. L. Darby and M. Nikolaou, "MPC: Current practice and challenges," *Control Engineering Practice*, vol. 20, no. 4, pp. 328–342, 2012.

[3] W. Y. Choi, S.-H. Lee, and C. C. Chung, "Horizonwise model-predictive control with application to autonomous driving vehicle," *IEEE Trans. Industrial Informatics*, vol. 18, no. 10, pp. 6940–6949, 2022.

[4] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1:43 scale RC cars," *Optimal Control Applications and Methods*, vol. 36, no. 5, pp. 628–647, 2014.

[5] S. Kozak, "From PID to MPC: Control engineering methods development and applications," in *K&I 2016*, 2016.

[6] D. Limon, J. Calliess, and J. Maciejowski, "Learning-based nonlinear model predictive control," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 7769–7776, 2017.

[7] M. Rokonuzzaman, N. Mohajer, S. Nahavandi, and S. Mohamed, "Model predictive control with learned vehicle dynamics for autonomous vehicle path tracking," *IEEE Access*, vol. 9, pp. 128 233–128 249, 2021.

[8] I. Markovsky, L. Huang, and F. Dörfler, "Data-driven control based on the behavioral approach: From theory to applications in power systems," *IEEE Control Systems*, vol. 43, no. 5, pp. 28–68, 2023.

[9] J. Coulson, J. Lygeros, and F. Dorfler, "Data-enabled predictive control: In the shallows of the deepc," in *18th ECC*, 2019, pp. 307–312.

[10] L. Huang, J. Lygeros, and F. Dörfler, "Robust and kernelized data-enabled predictive control for nonlinear systems," *IEEE Trans. Control Systems Technology*, vol. 32, no. 2, pp. 611–624, 2024.

[11] H. B. Pacejka and E. Bakker, "The magic formula tyre model," *Vehicle System Dynamics*, vol. 21, no. sup001, pp. 1–18, 1992.

[12] D. V. Quoc, "Tuning pid controller bases on random search algorithms," *International Journal of Engineering Research and*, vol. V10, no. 01, Jan. 2021. [Online]. Available: http://dx.doi.org/10.17577/IJERTV10IS010038

[13] R. Shaiakhmetov, "Crylab/deepc: Initial release," 2024. [Online]. Available: https://zenodo.org/doi/10.5281/zenodo.11515536