

Model- and Acceleration-based Pursuit Controller for High-Performance Autonomous Racing

Jonathan Becker, Nadine Imholz, Luca Schwarzenbach
Edoardo Ghignone, Nicolas Baumann and Michele Magno

Abstract—Autonomous racing is a research field gaining large popularity, as it pushes autonomous driving algorithms to their limits and serves as a catalyst for general autonomous driving. For scaled autonomous racing platforms, the computational constraint and complexity often limit the use of Model Predictive Control (MPC). As a consequence, geometric controllers are the most frequently deployed controllers. They prove to be performant while yielding implementation and operational simplicity. Yet, they inherently lack the incorporation of model dynamics, thus limiting the race car to a velocity domain where tire slip can be neglected. This paper presents Model- and Acceleration-based Pursuit (MAP) a high-performance model-based trajectory tracking controller that preserves the simplicity of geometric approaches while leveraging tire dynamics. The proposed algorithm allows accurate tracking of a trajectory at unprecedented velocities compared to State-of-the-Art (SotA) geometric controllers. The MAP controller is experimentally validated and outperforms the reference geometric controller four-fold in terms of lateral tracking error, yielding a tracking error of 0.055 m at tested speeds up to 11 m/s on a scaled racecar. Code: <https://github.com/ETH-PBL/MAP-Controller>.

I. INTRODUCTION

Motorsport racing has proven to enable knowledge transfer of cutting-edge research to the automotive industry [1, 2, 3]. In particular, autonomous racing presents a new frontier that promises to revolutionize autonomous driving by enabling and stress-testing new technologies and algorithms in the field of Self Driving Cars (SDC) [4, 5, 6, 7]. For this reason, many autonomous racing competitions have recently emerged, featuring different platforms and form factors, from full-scaled Indy Autonomous [6] and Formula Student Driverless [8] to scaled F1TENTH [5, 9]. A major challenge in these competitions is the completion of a racetrack in a high-speed setting [3, 5, 6, 8], which presents a demanding task for autonomous control systems, as the car's behavior is highly non-linear [10, 11, 12, 13]. This is particularly challenging when the controller needs to operate at the edge of traction, beyond the linear characteristics of the complex tire dynamics [10, 11, 12, 13]. Therefore, tracking trajectories accurately at cornering speeds that saturate the tire friction capacity, presents a challenging and central controls task for autonomous racing [3, 6, 8].

Classically, the control strategies in the autonomous racing setting could be loosely categorized into the following three

methods, listed in the order of increasing complexity and performance:

- I **Reactive Methods**: Compute control actions directly based on the sensor input they receive [14, 15]. This allows for simple and reactive control, without requiring proper state estimation. However, disregarding information about the track and the car, limits the performance of the controllers, especially in high-speed settings.
- II **Geometric Methods**: Utilize the geometric properties of the vehicle, typically derived from the kinematic single-track model [16]. Methods such as *Pure Pursuit* [17] or *Stanley Control* [18] separate longitudinal and lateral control. While these methods yield a great racing performance advantage over their reactive counterparts, they do not take into account the dynamic model of the vehicle, thus leading to sub-optimality.
- III **Model-Based Controllers**: A typical example of a model-based controller is the family of MPC methods. They consist of applying optimal control over a receding horizon and can take into account the dynamic model of the vehicle as well as state and input constraints thus yielding highly performant autonomous racing results [8, 10, 11]. However, precise state-estimation and knowledge of the car model and delays present in the system are essential to achieve good MPC performance, making this method the most computationally demanding and difficult to implement control strategy of the aforementioned methods [19].

The development of autonomous race cars brings about many often unforeseen challenges. Especially at small scale, energy and size constraints necessitate the deployment on embedded processors [20, 21] with limited computational resources. Constraints on sensors make state-estimation more difficult compared to full-scale cars and often introduce delays. Yet, for high-speed applications low latency and real-time throughput need to be guaranteed [11]. Many refrain from deploying MPC due to the added implementational and computational complexity given these constraints and prefer simpler alternatives [3, 17]. In order to maintain the performance advantage of model-based control, we propose a way of incorporating model knowledge into the simple control architecture of geometric controllers.

This paper introduces MAP, a high accuracy and low complexity model-based lateral controller to track an arbitrary raceline trajectory in a high-speed autonomous racing scenario.

Manuscript received 2 September 2022; revised 6 February 2023; accepted 17 January 2023. All authors are associated with Center for Project Based Learning, D-ITET, ETH Zurich. (Jonathan Becker, Nadine Imholz, Luca Schwarzenbach, Edoardo Ghignone and Nicolas Baumann contributed equally to this work.) (Corresponding author: Nicolas Baumann.)

The controller has been evaluated on the 1:10 scaled F1TENTH autonomous racing resource-constrained platform [5, 9]. Yet, it is to be mentioned, that the control strategy can be applied to full-scaled vehicles as well.

The proposed MAP controller incorporates the highly non-linear *Pacejka* tire model [13] into lateral geometric control, inspired by aerospace fixed-wing L_1 guidance [22]. This results in a more accurate control strategy that would need to be newly classified between geometric and model-based methods, as it leverages the accuracy of model-based lateral controls with the simplicity of a geometric controller. Furthermore, the proposed controller is computationally efficient enabling real-time operation on the computationally constrained On Board Computer (OBC) platform. The controller has been experimentally evaluated on the F1TENTH testing platform while racing at a tested top speed of 11 m/s. The main contribution of the paper is the design and evaluation of the proposed MAP controller which offers the following benefits:

- **High-Performance:** The model-based lateral controller results in a four-fold higher tracking accuracy at high velocity compared to geometric controllers such as *Pure Pursuit*. Namely, 0.055 m lateral error at speeds up to 11 m/s. By incorporating the tire dynamics, the controller is able to compute the steering angle necessary to track the desired trajectory with great accuracy.
- **Low Complexity:** As the controller still separates longitudinal and lateral control, it is simple to implement, comparable to a geometric controller, yet has the benefit of high accuracy by including model-based properties. Thus, it is significantly less complex to implement and optimize than a full-fledged MPC.
- **Computational Efficiency:** The computation of a single control cycle amounts to 6 ms¹. Therefore the low latency of the controller allows for real-time operation in a high-speed setting, on computationally constrained hardware.

II. RELATED WORK

The current SotA high-performance racing controllers, on full-scaled vehicles, are based on optimal control methods such as MPC [8, 23, 24]. MPC can offer the ability to optimally plan and track a racing trajectory, even incorporating system and safety constraints, yet at the expense of complexity and computation, [25, 26] of formulating and solving the Quadratic Programming (QP) problem.

However, for the reason of complexity and computational burden, MPC has rarely been used in scaled and computationally constrained autonomous racing competitions [3] such as F1TENTH or drone racing [27], where the robotic vehicles need to work with limited processing power. High-performance racing requires upper latency bounds of a few tens of milliseconds [11], therefore onboard processing on low-power embedded computing platforms calls for a computationally lightweight controller [25, 26].

To overcome the computational limits, the most utilized high-performance controller in scaled autonomous racing is of

the geometric type, most commonly a *Pure Pursuit* controller [17]. The *Pure Pursuit* controller is a lateral controller, that pursues a point on the raceline that lies in front of the vehicle, the distance towards said point is defined as the lookahead distance L_d , from which one can compute the necessary steering angle based on the *Ackermann* steering model [28]. As L_d is its only tune-able parameter, setting the L_d correctly is crucial for high-performance racing. In the traditional *Pure Pursuit* method [17], the L_d is a chosen constant. This inherently presents a trade-off; a too-short L_d produces oscillations at high velocities; too high L_d results in inaccurate tracking and the cutting of corners.

While the *Pure Pursuit* controller plays an important role in the domain of SDC and autonomous racing, the downsides of this control strategy become apparent once the controller is pushed to the physical edge of traction [10, 11]. Under high lateral acceleration the car's driving behavior becomes highly non-linear, which pure geometric controllers fail to capture, resulting in tracking errors [10, 11]. Previous works proposed methods of dynamically changing the lookahead distance L_d of *Pure Pursuit*, to mitigate this effect [29, 30, 31]. Others introduced fuzzy logic based on the lateral deviation and track curvature [32, 33, 34], they however did not address the underlying issue of disregarding tire slippage.

Similar to *Pure Pursuit*, a lateral guidance law can be found in the aerospace domain for fixed-wing Unmanned Aerial Vehicles (UAV)s, namely the L_1 guidance [22]. This nonlinear guidance logic is designed for curved trajectory tracking of a fixed-wing UAV, by choosing a reference point on the trajectory in front of the aircraft, namely the L_1 distance. Therefore, the similarities to *Pure Pursuit* with its lookahead distance L_d , can be seen. However, *Pure Pursuit* commands a steering angle, whereas as the L_1 guidance outputs a lateral acceleration a_{lat} governed by Eq. (1), where V is the velocity and η is the angle between the velocity vector and the reference point.

$$a_{lat} = 2 \frac{V^2}{L_1} \sin(\eta) \quad (1)$$

This paper proposes and demonstrates how to leverage the acceleration-based guidance signal a_{lat} to incorporate model knowledge in a geometric control setting while applying well-known adaptive L_d techniques. The *Pacejka Magic Tire Formula* [13] allows the computation of the required a_{lat} to map to the steering angle. The incorporation of the dynamic model allows the controller to consider tire slip while inherently scaling with the velocity, allowing for accurate lateral control in the high-speed and non-linear domain. This yields a high-performance racing controller that outperforms SotA geometric racing controllers in terms of lap time and four times in tracking error. The controller has been fully designed, implemented, and evaluated in a scaled F1TENTH car [35].

III. METHODOLOGY

In this section, we show through abstraction that *Pure Pursuit* uses the same L_1 guidance law that was proposed for UAVs [22]. With this, we show that *Pure Pursuit* disregards

¹On a single embedded ARMv8 core of the NVIDIA Jetson NX CPU

wheel slip, an assumption that fails at high speeds. We propose a method able to better capture and control the real-world dynamics of the car. This is done by identifying the car dynamics in Section III-B and using them to convert guidance to control commands via a Lookup Table (LUT) described in Section III-C. Finally, in Section III-D a mathematical reason for changing the lookahead distance is given.

A. Proposed Control System Overview

The L_1 guidance of [22] results in a desired centripetal acceleration a_c of the vehicle, equal to the tangential speed v_t multiplied with the yaw rate $\dot{\Psi}$:

$$a_c = v_t \cdot \dot{\Psi} = 2 \frac{v_t^2}{L_d} \sin(\eta) \quad (2)$$

Where L_d is the lookahead distance and η is the angle between the velocity vector and lookahead point.

Pure Pursuit can be derived from this equation, by mapping the lateral acceleration to a steering angle under the assumption of *Ackermann* steering. This assumes that no side slip occurs and the longitudinal velocity in the car's frame $v_x = v_t$:

$$\delta = \tan^{-1} \left(\frac{a_c \cdot l_{wb}}{v_x^2} \right) \quad (3)$$

where l_{wb} is the length of the wheelbase. Inserting the desired acceleration from Eq. (2) yields the formula used in *Pure Pursuit*:

$$\delta = \tan^{-1} \left(\frac{2 \sin(\eta) l_{wb}}{L_d} \right) \quad (4)$$

When driving corners at higher speeds, tire slip starts to occur and the underlying assumption of *Ackermann* steering from *Pure Pursuit* no longer holds. The desired acceleration a_c is no longer correctly commanded and the car starts to drift away from the trajectory. Therefore, we propose a method of incorporating the tire slip into the conversion from centripetal acceleration to the steering angle. This is done, by formulating the single-track model [16] in terms of the lateral forces acting on the tire, which yields the following equations of motion for the lateral velocity v_y and the yaw-rate $\dot{\Psi}$, with forces and angles defined in Fig. 1.

$$\dot{v}_y = \frac{1}{m} [F_{y,r} + F_{y,f}] - v_x \cdot \dot{\Psi} \quad (5)$$

$$\dot{\Psi} = \frac{1}{I_z} [-l_r F_{y,r} + l_f F_{y,f}] \quad (6)$$

Where m is the mass of the vehicle and I_z is the moment of inertia around the yaw axis. The lateral forces produced by the front (index f) and rear axle (index r) can be related to the tire slip angles α with the general formula ($i \in (f, r)$):

$$F_{y,i} = \mu F_{z,i} \cdot f_i(\alpha_i) \quad (7)$$

where μ is a friction coefficient and $F_{z,i}$ the vertical load on axle i . The respective tire slip angles α are found through the relation:

$$\alpha_f = \arctan \frac{v_y + \dot{\Psi} l_f}{v_x} - \delta \quad (8)$$

$$\alpha_r = \arctan \frac{v_y - \dot{\Psi} l_r}{v_x} \quad (9)$$

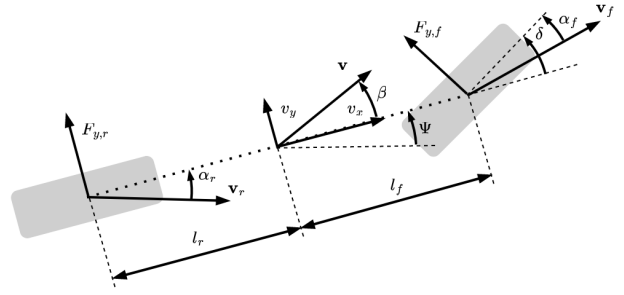


Figure 1: Overview of the single-track model [28] with the relevant forces and angles defined in their positive directions.

And the load on the front and rear axle are found depending on the longitudinal acceleration a_{long} and the mass distribution of the car, depending on the height of the Center of Gravity (CG) above the ground h_{cg} :

$$F_{z,f} = \frac{mgl_r - ma_{long}h_{cg}}{l_r + l_f} \quad (10)$$

$$F_{z,r} = \frac{mgl_f + ma_{long}h_{cg}}{l_r + l_f} \quad (11)$$

The steering angle δ enters these equations only by affecting the front tire slip angle α_f . Hence, it is necessary to find a model for the lateral force before being able to map the desired acceleration to a steering angle.

B. Tire Model

The *Pacejka Magic Formula* model [13] is able to accurately capture tire slip and model the transition of static to dynamic friction. In the original formulation, the authors included shift factors that are omitted in our version for simplicity. The model relates the tire slip angle at axle i to the respective lateral force $F_{y,i}$ via the non-linear relationship:

$$F_{y,i} = \mu F_{z,i} \cdot D_i \sin[C_i \cdot \arctan(B_i \alpha_i - E_i (B_i \alpha_i - \arctan(B_i \alpha_i)))] \quad (12)$$

The authors described the parameters as follows:

- B : Stiffness Factor
- C : Shape Factor
- \tilde{D} : Peak Value
- E : Curvature Factor

Where in Eq. (12) the peak value \tilde{D} is in fact $\mu F_{z,i} \cdot D_i$. A data-driven method was employed to identify these parameters. The data was gathered in steady-state cornering experiments as proposed by Voser et. al. [36], in which the car was driven at a constant speed and the steering angle was increased slowly (0.02 rad/s). This made it possible to relate the lateral forces at the front and rear tires to the measured steady-state acceleration by the Inertial Measurement Unit (IMU) $a_{y,imu}$ via the relationship

$$F_{y,f} = \frac{m \cdot l_r}{(l_r + l_f) \cdot \cos \delta} \cdot a_{y,imu} \quad (13)$$

$$F_{y,r} = \frac{m \cdot l_f}{l_r + l_f} \cdot a_{y,imu} \quad (14)$$

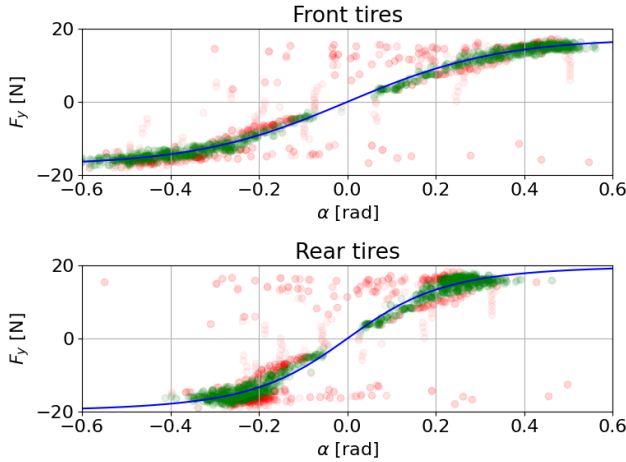


Figure 2: Data points obtained from the steady-state cornering experiment and the resulting model fit of the Pacejka model. Outliers are marked red, inliers green, and the model prediction is shown in blue for a fixed load of 16.4 N on the front axle and 18.6 N on the rear axle.

The parameters were then found through a least squares regression, in which the shape factor C was limited to 1.5 and the curvature factor E was limited to 1.1 to prevent the function to curve back towards zero for high tire slip angles. With this, the data fit in Fig. 2 was obtained. Outliers were rejected by performing three steps k of Expectation Maximization (EM) in which only points with an error smaller than $\frac{10}{2^k}$ were kept. This resulted in 25.5% of points being rejected as outliers and a resulting fit with an average residual of 0.87 N for the front and 1.35 N for the rear tires.

C. Lookup Table Generation

For a set of fixed steering angles at a certain speed v_x , the dynamics of v_y and $\dot{\Psi}$ in Eq. (6) and Eq. (5) converge to a steady-state, where the car turns at a constant radius and with constant centripetal acceleration. Since the guidance law results in smooth changes of commanded centripetal acceleration, the car deviates slightly from this steady-state condition. Therefore, the proposed method is able to neglect the transient phase, commanding steering angles resulting in the desired steady-state centripetal acceleration. The feedback loop and intrinsic stability of the guidance law [22] mitigate the inaccuracies during the transient phase. Since Eq. (5) cannot be solved analytically for the steady-state centripetal acceleration, a LUT was generated to obtain a mapping from steering angle to acceleration. For this, the system was simulated using the single-track dynamic model. The state was propagated with a range of constant longitudinal velocities and steering angles for which the resulting steady-state centripetal acceleration was recorded. With this, the controller is able to retrieve the required steering angle for a certain velocity and desired acceleration by interpolating between the closest elements in the table.

As can be seen in the visualization in Fig. 3, there is a lack of solutions for high δ and v_x . In this region, the simulated

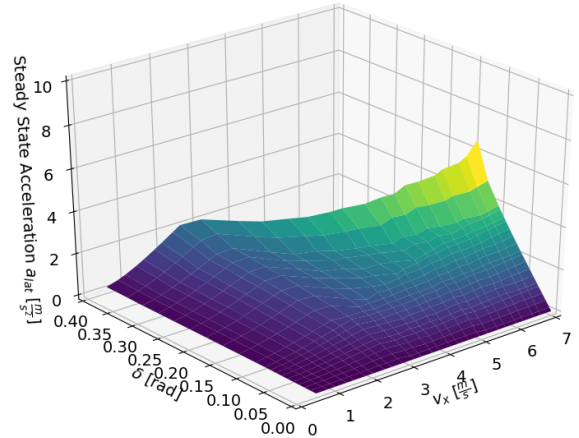


Figure 3: Visualization of the lookup table generated from the identified car model. The steady-state centripetal acceleration is shown as a function of the steering angle and velocity.

model did not converge to a steady state acceleration but rather an unstable drift, resulting in an upper bound for the achievable lateral acceleration at a given speed v_x .

From an implementation point of view, first, the lateral acceleration a_{lat} for the desired L_d is computed; secondly, the LUT returns the steering angle based on the model dynamics; from here on the MAP can be treated in the same way as a regular *Pure Pursuit* controller.

D. Adaptive Lookahead Point

Park et al. showed that the evolution of the lateral distance to the trajectory as a result of the guidance law in Eq. (2) could be approximated by a second order system with a time constant $\tau = L_d/v_x$ and natural frequency $\omega_n = \sqrt{2}v_x/L_d$ [22]. Therefore, to converge to the trajectory as quickly as possible, L_d should be chosen as small as possible. On the other hand, the natural frequency increases with the velocity. For stability, the natural frequency of the controller must be smaller than half the frequency of the entire vehicle dynamics including delays. Tests showed that for higher speeds this criterion was no longer met and the system became unstable. To address this, L_d was scaled with the velocity with the affine mapping $L_d = m + qv_{ctrl}$, where v_{ctrl} is the speed commanded to the car, and m, q are tunable coefficients. Relating the lookahead distance with speed made the natural frequency of the guidance logic independent of the velocity and ensured stability at higher speeds.

IV. EXPERIMENTAL RESULTS

This section presents the lap time and tracking benefits of incorporating the tire dynamics in a high-speed racing setting, as we compare the proposed MAP controller with a purely geometric *Pure Pursuit* controller. Both *Pure Pursuit* and the MAP controller were set up and tuned at equal conditions such that a fair comparison based on lap time and lap-wise Root Mean Square (RMS) lateral error to the global trajectory can be conducted using an FITENTH compliant car.

A. Experimental Setup

The experiments were conducted on an F1TENTH racecar based on [35], using the Robot Operating System (ROS). The scaled RC car chassis Traxxas Slash 1:10 4WD RTR was equipped with a Hokuyo UST-10LX LiDAR. *Pure Pursuit* and the MAP controller were tuned to minimize the lap-wise RMS lateral error on a reference track while tracking the same global trajectory. The global trajectory was planned with the global trajectory optimizer from [37], which generates a minimum curvature path and calculates a velocity profile with a forward-backward solver. As the velocity profile is computed for perfect tracking, the velocity was scaled down linearly to 60% for tuning, where 100% represents 8.5 m/s which is a high-speed for scaled tracks and is further highly track-dependent. The tuning regarded the lookahead distance L_d by varying the aforementioned coefficients m , q in Section III-D such that the respective controllers minimized the RMS lateral error to the reference trajectory at a 60% speed scaling. Allowing both controllers to be tuned individually to their optimal working point.

B. Velocity Scaling of Performance

To test the performance of both MAP and *Pure Pursuit* at speeds higher than that of tuning, the controllers were tested at different speeds, starting with a velocity scale of 60% and increasing it in steps of 2.5% until the controller was no longer capable of maintaining sufficiently accurate tracking, resulting in a crash. For each velocity and controller, 10 rounds were driven with the car. The track used was the same as the one used for tuning. As can be seen in Fig. 4, lateral error for the MAP controller is significantly lower than that of *Pure Pursuit* at all speeds. Specifically, at 82.5% speed, the tracking performance is nearly twice as good for the MAP with respect to the *Pure Pursuit* controller. In practice, this resulted in *Pure Pursuit* being at the edge of collision with the track boundaries, while MAP was still tracking the trajectory reasonably well. Regarding lap times, it can be seen that having more accurate trajectory tracking allows for planning more aggressive trajectories ultimately yielding better performance. The MAP controller achieves better lap times at every tested speed, achieving an average improvement of 0.227 s. The biggest improvement is seen at the highest speed, where MAP finishes the lap 0.350 s faster than the *Pure Pursuit* controller, corresponding to a 5% time improvement. However, the lower tracking error of the MAP permits tracking even faster trajectories, enabling even higher lap time improvements.

C. Effect of Tire Dynamics

To assess the effect of using different tire models, nominal MAP and *Pure Pursuit* are deployed alongside MAP using a linear tire model, whose coefficients, that linearly relate force to tire slip angle, are found by using regression on the same data as in Fig. 2. Tests are conducted at two different reference speed scales, 70% and 80% over five consecutive laps. The resulting lap times are slightly different from the previous paragraph as the track needed to be slightly changed for logistical reasons. In Fig. 6 one can see the different

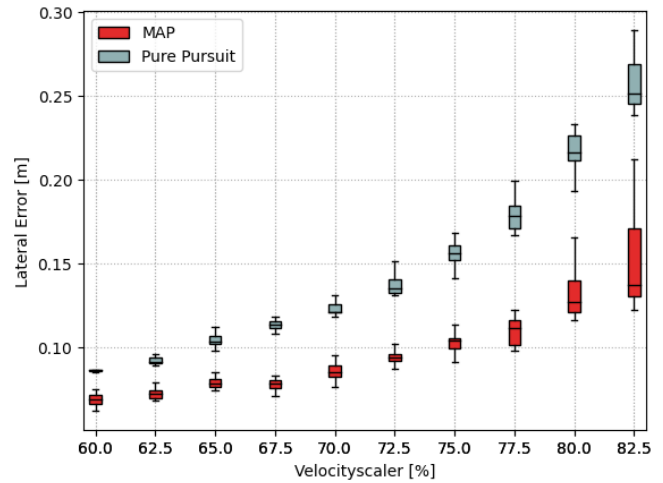


Figure 4: Lap-wise RMS lateral errors of *Pure Pursuit* and MAP with increasing velocity on the reference track. The box represents the lower and upper quartile; the whiskers represent the minimum and maximum measured values; the horizontal line represents the median value. It can be seen that *Pure Pursuit* displays consistently a higher (worse) lateral error with increasing velocity.

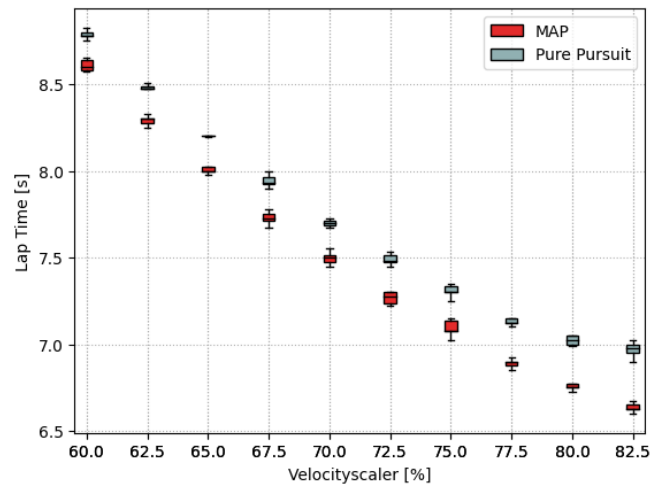


Figure 5: Lap-wise lap times of *Pure Pursuit* and MAP with increasing velocity on the reference track. The box represents the lower and upper quartile; the whiskers represent the minimum and maximum measured values; the horizontal line represents the median value. It can be seen that *Pure Pursuit* consistently displays higher (worse) lap times than the MAP.

trajectories driven by the two algorithms. The left plots show the results for the slower velocity: at this speed, the controller with the linear tire model is seen cutting the corners. The data in Table I shows that this results in a 2.3 % faster lap time at the cost of a 54.2 % larger lateral tracking error compared to nominal MAP with the *Pacejka* tire model. *Pure Pursuit* drives wide in all corners. Hence, the nominal MAP outperforms *Pure Pursuit* with 58.2 % lower average deviation, 45.5 % lower maximum deviation, and 5.1 % faster lap time at this

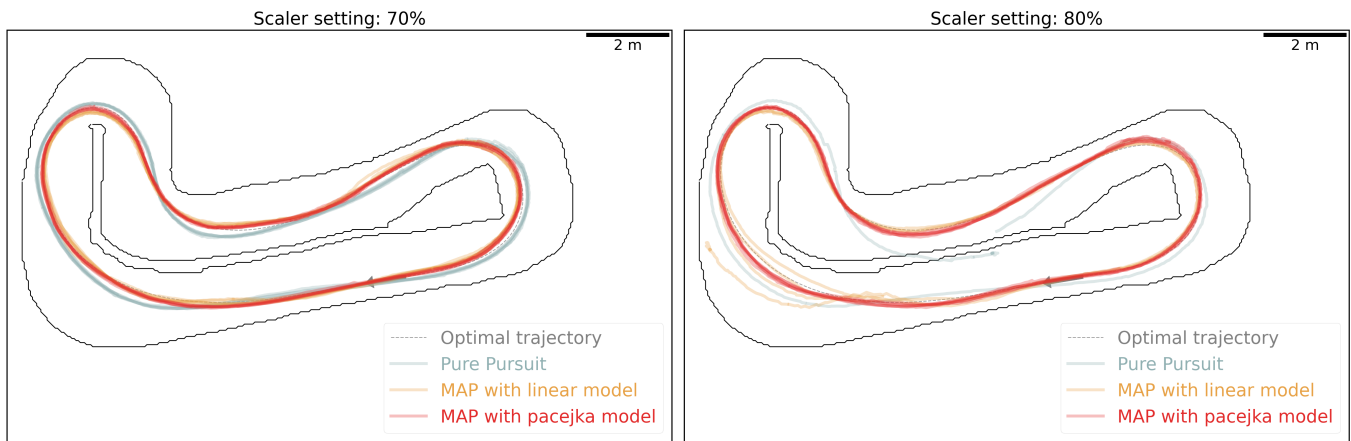


Figure 6: Trajectories of the *Pure Pursuit* controller and the MAP controller with both linear and *Pacejka* tire model. The reference line is shown dashed in gray and an arrow indicates the direction of racing. On the left, the reference speed is scaled to 70% of the nominal one, and to the right to 80%. The tracking benefit of accurate tire dynamics can be seen by noticing that in the right plot, at higher speeds, both *Pure Pursuit* and the MAP with linear tire model crash into the track boundaries (driven paths end abruptly at the edge). The MAP controller with the *Pacejka* tire model, instead, manages to successfully track the given trajectory over five laps.

speed.

At higher speeds, the benefit of using a more accurate tire model in the controller becomes even more apparent, as the *Pacejka* version of MAP is the only one able to complete five laps, achieving robust tracking of the trajectory. The linear version fails to complete a lap at higher speeds, as oscillations cause it to collide with the track boundaries, and *Pure Pursuit* crashes due to under-steering in the third corner. The MAP controller with *Pacejka* tire model again achieves the lowest tracking error and deviation. Using the linear tire model resulted in a ten-fold increase in average lateral error during the two laps driven before the crash. It is therefore clear that having a precise tire model capable of capturing the non-linear tire dynamics is key to achieving high performance in autonomous racing.

C_{glob}	Controller	t_{lap}^{μ} [s]	$ d^{\mu} $ [m]	$ d^{max} $ [m]
0.7	<i>Pure Pursuit</i>	8.74	0.115	0.33
	MAP with linear	8.10	0.074	0.20
	MAP with Pacejka	8.29	0.048	0.18
0.8	<i>Pure Pursuit</i>	N.C. (7.93*)	N.C. (0.220*)	N.C. (0.53*)
	MAP with linear	N.C. (7.35**)	N.C.(0.55**)	N.C.(0.32**)
	MAP with Pacejka	7.39	0.055	0.23

Table I: Lap times, average and maximum deviation from the trajectory for *Pure Pursuit*, MAP with linear tire model, and nominal MAP with *Pacejka* tire model.

N.C.: Not Completed

*: Crashed after one lap

**Unstable and crashed after two laps

D. Race Results

The MAP controller with *Pacejka* tire parameters was deployed on the real racecar also in the 2022 *FITENTH Autonomous Grand Prix Germany*, obtaining the best performance in the field, winning the competition and achieving the fastest lap time, reaching speeds over 8 m/s and lateral

accelerations of 1.25 g. By measuring the maximal lateral force that could be applied to the racecar with a spring balance, resulting in 45 N and the car’s weight of 3.5 kg, affirming the claim that the racecar can operate at the edge of friction.

V. CONCLUSIONS

We presented MAP, a highly performant racing controller suitable for computationally constrained hardware, that incorporates model dynamics while remaining simple to implement and operate. The proposed controller significantly outperforms SotA geometric controllers while conserving their benefits of design simplicity and real-time computational feasibility. By leveraging model knowledge and incorporating tire models capturing lateral tire slip into the control law, tracking performance is increased four-fold during our tests at higher velocities of up to 11 m/s.

Future work will focus on adaptive tire parameter estimation for the MAP controller such that system identification would not be needed to be recomputed offline in different racing environments and an in-depth comparison with MPC under realistic conditions. Furthermore, testing the behavior on a full-scaled racing car would be of great interest. The full MAP code is available on Github: <https://github.com/ETH-PBL/MAP-Controller>.

ACKNOWLEDGMENT

The authors would like to thank all team members of the *ForzaETH* racing team, as well as Dr. Christian Vogt and Dr. Andrea Carron of ETH Zürich, for their constructive and fruitful algorithmic discussions.

REFERENCES

- [1] S. Jarvenpaa and W. Standaert, “Emergent ecosystem for radical innovation: Entrepreneurial probing at formula e,” in *Proceedings of the 50th HICSS*, 2017.

- [2] M. Finn, "From accelerated advertising to fanboost: mediatized motorsport," *Sport in Society*, vol. 24, no. 6, pp. 937–953, 2021. [Online]. Available: <https://doi.org/10.1080/17430437.2019.1710131>
- [3] J. Betz, H. Zheng, A. Liniger, U. Rosolia, P. Karle, M. Behl, V. Krovi, and R. Mangharam, "Autonomous vehicles on the edge: A survey on autonomous vehicle racing." [Online]. Available: <https://arxiv.org/abs/2202.07008>
- [4] S. Jung, S. Cho, D. Lee, H. Lee, and D. H. Shim, "A direct visual servoing-based framework for the 2016 IROS autonomous drone racing challenge," *J. field robot.*, vol. 35, no. 1, pp. 146–166, Jan. 2018.
- [5] M. O’Kelly, H. Zheng, A. Jain, J. Auckley, K. Luong, and R. Mangharam, "TunerCar: A superoptimization toolchain for autonomous racing," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 5356–5362.
- [6] A. Wischniewski, M. Geisslinger, J. Betz, T. Betz, F. Fent, A. Heilmeyer, L. Hermansdorfer, T. Herrmann, S. Huch, P. Karle, F. Nobis, L. Ögretmen, M. Rowold, F. Sauerbeck, T. Stahl, R. Trauth, M. Lienkamp, and B. Lohmann, "Indy autonomous challenge – autonomous race cars at the handling limits," 2022. [Online]. Available: <https://arxiv.org/abs/2202.03807>
- [7] J. Betz, A. Wischniewski, A. Heilmeyer, F. Nobis, T. Stahl, L. Hermansdorfer, and M. Lienkamp, "A software architecture for an autonomous racecar," in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, 2019, pp. 1–6.
- [8] J. Kabzan, M. I. Valls, V. J. F. Reijgwart, H. F. C. Hendriks, C. Ehmke, M. Prajapat, A. Bühler, N. Gosala, M. Gupta, R. Sivanesan, A. Dhall, E. Chisari, N. Karnchanachari, S. Brits, M. Dangel, I. Sa, R. Dubé, A. Gawel, M. Pfeiffer, A. Liniger, J. Lygeros, and R. Siegwart, "AMZ driverless: The full autonomous racing system," *J. field robot.*, no. rob.21977, Aug. 2020.
- [9] M. O’Kelly, H. Zheng, D. Karthik, and R. Mangharam, "F1tenth: An open-source evaluation environment for continuous control and reinforcement learning," in *NeurIPS 2019 Competition and Demonstration Track*. PMLR, 2020, pp. 77–89.
- [10] J. L. Vázquez, M. Brühlmeier, A. Liniger, A. Rupenyan, and J. Lygeros, "Optimization-based hierarchical motion planning for autonomous racing," 2020. [Online]. Available: <https://arxiv.org/abs/2003.04882>
- [11] 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, p. 628–647, Jul 2014. [Online]. Available: <http://dx.doi.org/10.1002/oca.2123>
- [12] A. Liniger, "Pushing the limits of friction: A story of model mismatch," 2021, iCRA21 Autonomous Racing. [Online]. Available: https://www.youtube.com/watch?v=_rTawyZghEg&t=136s
- [13] H. B. Pacejka and E. Bakker, "The magic formula tyre model," *Vehicle system dynamics*, vol. 21, no. S1, pp. 1–18, 1992.
- [14] V. Sezer and M. Gokasan, "A novel obstacle avoidance algorithm: "follow the gap method"," *Robotics and Autonomous Systems*, vol. 60, no. 9, pp. 1123–1134, 2012. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0921889012000838>
- [15] N. Otterness, "The "disparity extender" algorithm, and f1/tenth," April 2019.
- [16] P. Polack, F. Altché, B. d’Andréa Novel, and A. de La Fortelle, "The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?" in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 812–818.
- [17] J. M. Snider, "Automatic Steering Methods for Autonomous Automobile Path Tracking," February 2009. [Online]. Available: https://www.ri.cmu.edu/pub_files/2009/2/Automatic_Steering_Methods_for_Autonomous_Automobile_Path_Tracking.pdf
- [18] C. Lyu, D. Lu, C. Xiong, R. Hu, Y. Jin, J. Wang, Z. Zeng, and L. Lian, "Toward a gliding hybrid aerial underwater vehicle: Design, fabrication, and experiments," *J. field robot.*, vol. 39, no. 5, pp. 543–556, Aug. 2022.
- [19] A. Gonzalez, A. Castillo, P. Garcia, and P. Albertos, "Robust stabilization of time-varying delay systems with predictor-observer based controller," *IFAC-PapersOnLine*, vol. 52, no. 1, pp. 213–218, 2019, 12th IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems DYCOPS 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405896319301491>
- [20] B. P. Duisterhof, S. Krishnan, J. J. Cruz, C. R. Banbury, W. Fu, A. Faust, G. C. de Croon, and V. J. Reddi, "Tiny robot learning (tinyrl) for source seeking on a nano quadcopter," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 7242–7248.
- [21] S. An, F. Zhou, M. Yang, H. Zhu, C. Fu, and K. A. Tsintotas, "Real-time monocular human depth estimation and segmentation on embedded systems," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 55–62.
- [22] S. Park, J. Deyst, and J. How, "A new nonlinear guidance logic for trajectory tracking," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*. Reston, Virginia: American Institute of Aeronautics and Astronautics, Aug. 2004.
- [23] A. Jain, M. O’Kelly, P. Chaudhari, and M. Morari, "BayesRace: Learning to race autonomously using prior experience," *arXiv:2005.04755 [cs, eess]*, Nov. 2020, arXiv: 2005.04755. [Online]. Available: <http://arxiv.org/abs/2005.04755>
- [24] L. P. Fröhlich, C. Küttel, E. Arcari, L. Hewing, M. N. Zeilinger, and A. Carron, "Model learning and contextual controller tuning for autonomous racing," 2021.
- [25] F. Fuchs, Y. Song, E. Kaufmann, D. Scaramuzza, and P. Durr, "Super-human performance in gran turismo sport using deep reinforcement learning,"

- IEEE Robotics and Automation Letters*, vol. 6, no. 3, p. 4257–4264, Jul 2021. [Online]. Available: <http://dx.doi.org/10.1109/LRA.2021.3064284>
- [26] Y. Song, H. Lin, E. Kaufmann, P. Dürr, and D. Scaramuzza, “Autonomous overtaking in gran turismo sport using curriculum reinforcement learning,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 9403–9409.
- [27] Y. Song, M. Steinweg, E. Kaufmann, and D. Scaramuzza, “Autonomous Drone Racing with Deep Reinforcement Learning,” *arXiv:2103.08624 [cs]*, Aug. 2021, arXiv: 2103.08624. [Online]. Available: <http://arxiv.org/abs/2103.08624>
- [28] M. Althoff, M. Koschi, and S. Manzingler, “CommonRoad: Composable benchmarks for motion planning on roads,” in *2017 IEEE Intelligent Vehicles Symposium (IV)*. Los Angeles, CA, USA: IEEE, Jun. 2017, pp. 719–726. [Online]. Available: <http://ieeexplore.ieee.org/document/7995802/>
- [29] V. Sukhil and M. Behl, “Adaptive lookahead pure-pursuit for autonomous racing,” 2021. [Online]. Available: <https://arxiv.org/abs/2111.08873>
- [30] W.-J. Wang, T.-M. Hsu, and T.-S. Wu, “The improved pure pursuit algorithm for autonomous driving advanced system,” in *2017 IEEE 10th International Workshop on Computational Intelligence and Applications (IWCIA)*, 2017, pp. 33–38.
- [31] R. Wang, Y. Li, J. Fan, T. Wang, and X. Chen, “A novel pure pursuit algorithm for autonomous vehicles based on salp swarm algorithm and velocity controller,” *IEEE Access*, vol. 8, pp. 166 525–166 540, 2020.
- [32] A. Ollero, A. García-Cerezo, and J. Martínez, “Fuzzy supervisory path tracking of mobile robots1,” *IFAC Proceedings Volumes*, vol. 26, no. 1, pp. 277–282, 1993, 1st IFAC International Workshop on Intelligent Autonomous Vehicles, Hampshire, UK, 18-21 April. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1474667017493128>
- [33] L. Chen, N. Liu, Y. Shan, and L. Chen, “A robust lookahead distance tuning strategy for the geometric path tracking controllers,” in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 262–267.
- [34] Y. Shan, W. Yang, C. Chen, J. Zhou, L. Zheng, and B. Li, “Cf-pursuit: A pursuit method with a clothoid fitting and a fuzzy controller for autonomous vehicles,” *International Journal of Advanced Robotic Systems*, vol. 12, no. 9, p. 134, 2015. [Online]. Available: <https://doi.org/10.5772/61391>
- [35] F1TENTH Foundation, “Building the F1TENTH Car.” [Online]. Available: <https://f1tenth.org/build.html>
- [36] C. Voser, R. Y. Hindiyeh, and J. C. Gerdes, “Analysis and control of high sideslip manoeuvres,” *Vehicle System Dynamics*, vol. 48, no. sup1, pp. 317–336, 2010. [Online]. Available: <https://doi.org/10.1080/00423111003746140>
- [37] A. Heilmeier, A. Wischnewski, L. Hermansdorfer, J. Betz, M. Lienkamp, and B. Lohmann, “Minimum curvature trajectory planning and control for an autonomous race car,” *Vehicle System Dynamics*, vol. 58, no. 10, pp. 1497–1527, 10 2020.