IEEE Open Journal of
Intelligent Transportation
Systems

Received 11 February 2022; revised 16 April 2022; accepted 24 May 2022. Date of publication 8 June 2022; date of current version 24 June 2022.

Digital Object Identifier 10.1109/OJITS.2022.3181510

Autonomous Vehicles on the Edge: A Survey on Autonomous Vehicle Racing

JOHANNES BETZ[1] (Member, IEEE), HONGRUI ZHENG[1] (Student Member, IEEE), ALEXANDER LINIGER[2], UGO ROSOLIA[3], PHILLIP KARLE[4], MADHUR BEHL[5], VENKAT KROVI[6] (Senior Member, IEEE), AND RAHUL MANGHARAM[1] (Member, IEEE)

[1]Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104, USA

[2]Department of Information Technology and Electrical Engineering, ETH Zurich, 8092 Zurich, Switzerland

[3]AMBER Lab, California Institute of Technology, Pasadena, CA 91125, USA

[4]Institute of Automotive Technology, Technical University of Munich, 80333 Munchen, Germany

[5]Department of Computer Science, University of Virginia, Charlottesville, VA 22904, USA

[6]College of Engineering, Computing and Applied Sciences, Clemson University, Clemson, SC 29634, USA

CORRESPONDING AUTHOR: J. BETZ (e-mail: joebetz@seas.upenn.edu)

This work was supported in part by the Department of Transportation and in part by NSF.

ABSTRACT The rising popularity of self-driving cars has led to the emergence of a new research field in recent years: Autonomous racing. Researchers are developing software and hardware for high-performance race vehicles which aim to operate autonomously on the edge of the vehicle's limits: High speeds, high accelerations, low reaction times, highly uncertain, dynamic, and adversarial environments. This paper represents the first holistic survey that covers the research in the field of autonomous racing. We focus on the field of autonomous racecars only and display the algorithms, methods, and approaches used in the areas of perception, planning, control, and end-to-end learning. Further, with an increasing number of autonomous racing competitions, researchers now have access to high-performance platforms to test and evaluate their autonomy algorithms. This survey presents a comprehensive overview of the current autonomous racing platforms, emphasizing the software-hardware co-evolution to the current stage. Finally, based on additional discussion with leading researchers in the field, we conclude with a summary of open research challenges that will guide future researchers in this field.

INDEX TERMS Autonomous systems, autonomous vehicles, intelligent vehicles, advanced driver assistance, simultaneous localization and mapping, path planning, control.

I. INTRODUCTION

WHAT aerospace engineering is to aviation, motorsport is to automotive technology. For over a century now, racing series such as Formula 1, Indy Car, or the World Rally Championship have inspired research and product innovation to improve performance and safety in commercial road vehicles. These developments include well-known elements such as the disc brake, the turbocharger, or production measures for fiber composites (e.g., carbon). In more recent years, developments in the hybrid powertrain and the connectivity (real-time measurement and transfer of vehicle data) of the vehicles have emerged. With millions of dollars of investment and prestige at stake, these developments target a singular goal: The racecar must achieve the fastest lap time and thus win the race at the end. With a vehicle that moves at the dynamic limits of handling, that reaches high velocities, that is designed for aerodynamic efficiency, and that consists of pure lightweight construction, traditionally the target of the minimum lap time can only be achieved through extremely novel, sophisticated, and radical developments. But the technical development of the racecar is only half of the effort.

The review of this article was arranged by Associate Editor Jiaqi Ma.

In motorsport racing, it all boils down to the ability of the driver to operate the racecar at its limits [1], [2]. Expert race drivers are extremely proficient in pushing the racecar to its dynamical limits of handling, while accounting for continuously changing parameters such as tire wear, changing brake bias, and engine maps from turn to turn, communicating strategy and status with the team in the pits, and trying to maintain track position or overtake fierce competitors all while driving at speeds exceeding 300 km/h.

All of these facets manifest themselves in the innumerable research and development challenges sought to be addressed in the burgeoning field of *autonomous vehicle racing*. The current state of the art of autonomous driving software – either from commercial companies or researchers – can operate autonomously but only to a limited velocity. Everything that we find in classic motorsport can also be found in autonomous motorsport - with one difference: The racecar-driver is based on software only. This means that a highly sophisticated autonomous driving software needs to replace the human pilot. It should be capable of detecting other vehicles, localizing the vehicle position relative to the opponents and the track while driving at high speeds, planning dynamic trajectories to allow overtaking in adversarial environments, and correcting at high frequency to the steering angle to stay on the racetrack. Furthermore, the vehicle needs to execute a performance assessment on its own by adjusting the aerodynamics, energy distribution, differential settings, or brake balance settings based on tire wear, temperature, and weather. On this premise, an autonomous racecar exceeds the requirements for the software to a vast extent in comparison to an average passenger car - which provides many learning outcomes, research questions, and new algorithmic developments [3], [4]. It is also worthy of note that the requirements for autonomous racecars versus passenger vehicles can be quite disparate - precluding a direct transplant of a complete autonomous driving stack. Therefore, autonomous racing has emerged as a field where advanced algorithmic approaches are tested and then individually transferred to the development of autonomous passenger vehicles - similar to classic motorsport.

This paper provides the first survey of state-of-the-art research in the field of autonomous vehicle racing. By summarizing, classifying, and evaluating the different software and hardware developments, we provide a holistic overview of the research in this field. Finally, we discuss future research directions by highlighting open questions and challenges in autonomous racing.

A. CONTRIBUTIONS

In this survey, we present the efforts and research conducted in recent years in the field of autonomous racing. This work has four main contributions:

1) We provide the first survey to comprehensively cover the topic of autonomous vehicle racing for both software and hardware developments.
2) We extensively review all research papers that developed new autonomy software for autonomous racecars. By splitting this software review into subsections of perception, planning, and control, we detail which methods and approaches were used. We compare the different approaches and explain their algorithmic setup. Furthermore, we discuss recent efforts made with techniques from the field of deep neural networks (DNN) and reinforcement learning (RL) to achieve a partial or full end-to-end pipeline for autonomous racing.
3) We review the current autonomous racing competitions, which provide hardware, a racing environment, and an organization (e.g., sports and technical regulations). We compare the different racing series and hardware against each other and give a holistic overview for potentially interested researchers.
4) Finally, we list open research questions and challenges in the field of autonomous racing. We discuss that these open challenges can be applied to the area of passenger cars, too, and provide opportunities for future researchers to work on relevant research topics.

B. PRELIMINARY REMARKS

1) DEFINITION: AUTONOMOUS RACING

Although the term autonomous racing can be referred to different applications (e.g., drone racing), we focus on research in the field of autonomous racing *cars*. These racecars need to have four wheels, can either have a combustion engine or electrical engine as the main power unit, can be real racecars (e.g., Formula 1 car) or small-scale vehicles (e.g., 1:10 scale). In addition, the software and hardware surveyed here must have a clear connection to the field of automobile racing. This means that the authors of these papers either used specific hardware that is acting in a racing environment (e.g., racetrack, adversarial setup) they used a particular simulation that displays a racecar within a racing environment (e.g., a racing game for PC) or their research demonstrates specific solutions for a racing problem (e.g., driving fast autonomously around the racetrack). Although some authors present results and algorithms for high-speed autonomous driving on the freeway, this work is not covered in this survey because neither the aspect of handling the vehicles at the limits nor the adversarial context is given here sufficiently.

2) RESEARCH CATEGORIZATION

The field of autonomous racing provides plenty of development and research categories. For this survey, we want to display both software and hardware efforts in the area of autonomous racing and therefore, we use the following *perception - planning - control* pipeline [239] depicted in Figure 1 to categorize the research papers.

Research and developments in the field of autonomous racing hardware (sensors and vehicle hardware) are discussed in Section III. Unfortunately, we have not seen any specific

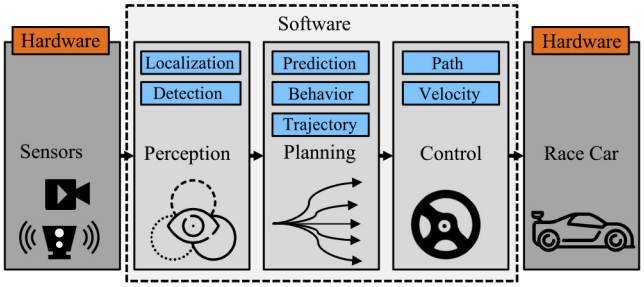

FIGURE 1. Autonomous driving pipeline including both hardware and software that provides the categorization for the survey in the field of autonomous racing.

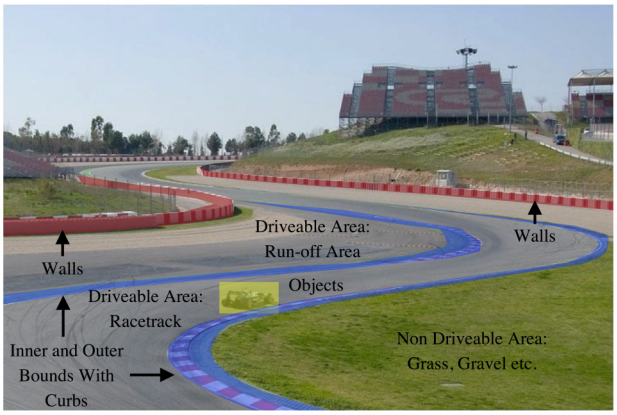

FIGURE 2. Racetrack with environmental specifics: Inner-and outer bounds, racecar objects, walls and run-off area.

developments of sensors for the purpose of autonomous racing and therefore, we heavily focus on the used racing vehicle platforms. By presenting vehicle setups (sensors, computation hardware, racing environment), we provide a clear overview of which hardware is available for researchers. The biggest part of this survey paper presents research and software in the field of perception, planning, and control in Section II. In Section II-A *perception*, we cover all algorithms that provide either a solution for mapping, localization or object detection. In Section II-B *planning* we display global and local trajectory and behavior planners. The final Section II-C is used to present algorithms in the field of *control* and displays the solutions for path and velocity tracking at the handling limits. Unfortunately, many papers out there act on the intersection of planning and control. Those papers have no clear distinction in which field they belong and therefore, we decided to categorize them in either field based on their focus. The method of DNNs has become popular in recent years and different authors proposed so-called *end-to-end* approaches that solve the autonomous driving task. These techniques are described in Section II-D. In addition, some authors proposed evaluations with racecars, complete software pipelines, modeling efforts, and simulation environments for autonomous racecars that do not fit in the proposed categories. Those papers are listed in Section II-E. In summary, this survey covers all research papers in the field published until the beginning of 2022.

II. AUTONOMOUS RACING SOFTWARE

A. PERCEPTION

Perception is the general term for all algorithms that perceive the environment and derive knowledge about it. In particular, perception includes detecting objects, detecting the free space, mapping the environment as well as localizing the autonomous vehicle. In an autonomous racing environment we deal with high speeds and therefore the question arises: How fast is too fast? Falanga *et al.* [56] tried to answer this question for autonomous robots with an additional case study on autonomous quadrocopters. The authors conclude that the maximum latency an autonomous system can tolerate to guarantee safety (not crashing in an object) is related to the desired speed, the agility of the system (e.g., the maximum acceleration it can produce) and the perception parameter of the sensors (e.g., the sensing range). For autonomous racecars the same parameters can be taken into account but no particular evaluation regarding high speed perception for autonomous racecars has been done yet. The current state of the art in the field of autonomous racing perception is summarized, categorized and displayed in Table 1.

A racetrack normally provides some specifics that can not be found on normal streets. As depicted in Figure 2, a racetrack consists of a single lane that is the main driveable space. This lane is defined by an inner and outer bound that can consists of additional curbs in the turns. On both left and right side of the track there are zones consisting of grass, gravel or tarmac (run-off area) where racecars can drive in an evasive maneuver or if they miss the race line. The racetrack is finally surrounded by walls. Depending on the racing series these types of the racetrack features can vary (e.g., Formula E: no gravel or grass). We define the fundamental problems for autonomous racing perception as the following:

- High speed object detection.
- High speed localization and state estimation.
- Localization on wide areas without specific landmarks.
- Precise localization information necessary to achieve high dynamic trajectory planning and control.

Although the racetrack provides a very simple structure with a single driveable lane, the long distance to the walls and non-existing landmarks make this environment quite difficult to perceive. None of the cited papers use predefined High-Definition Maps (HD-Maps) that are known from passenger autonomous driving development. An open-source library [79] of racetracks provides a simple 2D-birds-eye-view with inner and outer bounds (x-and y-Position) of about 30 racetracks around the world that can be used for planning but not for localization. In Nobis *et al.* [136] an adaption and enhancement of well-known simultaneous localization and mapping (SLAM) algorithms (*Google Cartographer* [240], *GMapping*) is described to create maps of large-scale outdoor environments. Palafox *et al.* [144] use a vision-based method

TABLE 1. Overview of research in the field of autonomous racing perception.

Name and Reference	Year	Perception Category	Method	Sensor Type	Tested on Hardware	Racing Series	Max. Speed (km/h)
Nobis et al. [136]	2019	Mapping	SLAM	LiDAR	Yes	Roborace	30
Palafox et al. [144]	2019	Free Space Detection	Semantic Segmentation	Camera	No	Roborace	-
Valls et al. [194]	2018	Localization	SLAM, EKF	LiDAR; Odometry	Yes	FSD	80
Brunnbauer et al. [28]	2019	Localization	Cone Detection	Camera	Yes	F1TENTH	-
Gotlieb et al. [70]	2019	Localization	AMCL	LiDAR; Odometry	Yes	F1TENTH	-
Stahl et al. [177]	2019	Localization	AMCL	LiDAR	Yes	Roborace	150
Wischnewski et al. [213]	2019	Localization	Kalman Filter	GPS; LiDAR; Odometry	Yes	Roborace	150
Massa et al. [129]	2020	Localization	EKF, AMCL	LiDAR; Odometry	Yes	Roborace	200
Renzler et al. [158]	2020	Localization	Distortion correction	LiDAR	Yes	Roborace	90
Zubaca et al. [224]	2020	Localization	H ∞ Filter	LiDAR; Odometry	Yes	Roborace	60
Schratter et al. [229]	2021	Localization	NDT	LiDAR	Yes	Roborace	122
Gosala et al. [69]	2019	Localization	SLAM	LiDAR; Odometry	Yes	FSD	90
Andresen et al. [13]	2020	Localization	GraphSLAM; Cone Detection	Camera; LiDAR; Odometry	Yes	FSD	-
Srinivasan et al. [175]	2020	Localization	RNN	Odometry; Vehicle Data	Yes	FSD	40
Le Large et al. [118]	2021	Localization	Graph SLAM; EKF SLAM	LiDAR	Yes	FSD	-
Peng et al. [149]	2021	Localization	GraphSLAM; Cone Detection	Camera; LiDAR; Odometry	Yes	FSD	-
Sauerbeck et al. [168]	2022	Localization	OpenVSLAM	Camera;LiDAR	Yes	IAC	300
Strobel et al. [182]	2020	Object Detection & Localization	YOLO v3	Stereo camera and Monocular camera	Yes	FSD	-
Dhall et al. [48]	2019	Object Detection	YOLO v2	Camera	Yes	FSD	-
De Rita et al. [47]	2019	Object Detection	Tiny-YOLO; Proteins	Camera	Yes	FSD	-
Puchtler et al. [153]	2020	Object Detection	SSD MobileNet v2	Camera	Yes	FSD	-
Vödisch et al. [49]	2022	Object Detection	Dataset	Camera	Yes	FSD	-

to detect the free space when no lane lines are present by only using camera images and depth information as input for a DNN.

Most perception research for autonomous racing is listed in the field of localization techniques. Although many research vehicles are equipped with differential GPS (dGPS) that delivers a high localization accuracy, the goal of many autonomous racing researchers is to deliver software based localization solutions only. Both [28] and [70] use a 1:10 scale vehicle for their localization techniques. While Brunnbauer and Bader [28] use the camera to detect cones that create features to enhance the odometry localization, Gotlib *et al.* [70] map the track with an onboard 2D-LiDAR and run a Robot Operating System (ROS) based Adaptive Monte Carlo Localization (AMCL). The same AMCL approach is also used and adapted by Stahl *et al.* [177] to run on the Roborace research vehicle. By using pre-generated maps based on LiDAR data, the car achieves a mean absolute lateral error of 0.086m at a velocity of 150 km/h. A comparison to this work is done in [213] were odometry, GPS and LiDAR data is fused in a Kalman Filter (KF) based on a purely kinematic vehicle dynamics model to achieve localization at high speeds. The Roborace vehicle is also used for the localization research of Renzler *et al.* [158], Zubaca *et al.* [224] and Schratter *et al.* [229].

To increase the localization performance at high speeds the distortion of the LiDAR measurement is analyzed and a compensation is proposed in [158]. It is shown that this correction can be straightforward and has a high performance with objects moving at faster speeds. The work from Zubaca *et al.* [224] presents an extended H∞ Filter (EHF) based on a kinematic motion model assuming constant turn-rate and acceleration to fuse LiDAR, IMU (inertial measurement unit), and vehicle dynamic sensors' measurements. The proposed EHF shows slightly better estimation performance in high dynamic driving scenarios in comparison to an extended Kalman Filter (EKF). In [229] a complete process for both mapping an localization on racetracks with the Normal Distributions Transform (NDT) method is displayed. Based on this approach the Roborace vehicle reaches up to 122 km/h and an average localization error of 0.2 m. Massa *et al.* [129] are using two multi-rate EKFs and an extend AMCL that exploits some a priori knowledge of the environment on the Roborace vehicle. The authors showed that the pose error heavily depends on the car's velocity, and varies in average from 0.1 m (at 60 km/h) to 1.48 m (at 200 km/h) laterally and from 1.9 m (at 100 km/h) to 4.92 m (at 200 km/h) longitudinally.

For Formula Student Driverless (FSD) vehicles Le Large *et al.* [118] show a comparison between GraphSLAM and an EKF-SLAM. Based on their experimental analysis with the FSD vehicle and the maps generated by the algorithms they showed that GraphSLAM outperforms EKF-SLAM in terms of accuracy. In [13], [69], [194] the localization and mapping approaches for the FSD vehicle of the AMZ team is presented. While in [69] only a LiDAR based SLAM was used, the student team extended the work with an additional LiDAR and

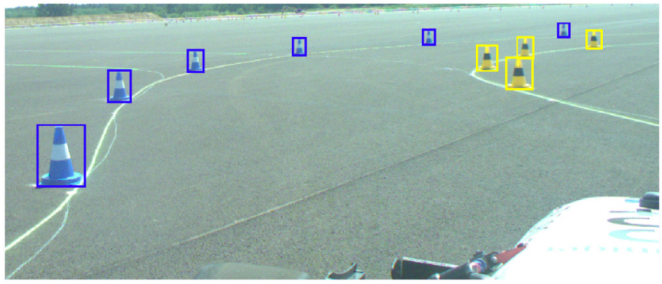

FIGURE 3. Exemplary cone detection in the Formula Student Driverless competition. The racetrack is defined by a left boundary (blue cones) and a right boundary (yellow cones) which need to be detected by the teams. This image is part of the FSOCO dataset [49] the supports FSD teams with the development of their perception pipelines.

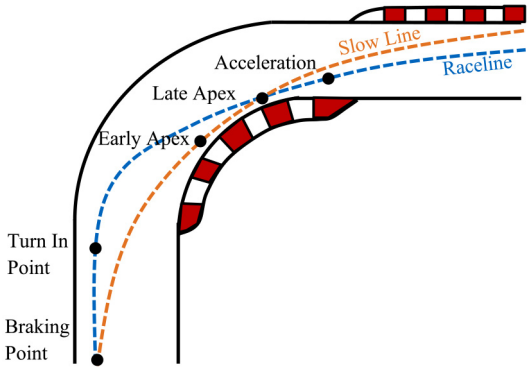

FIGURE 4. Theory of raceline (blue) and slow line (orange) on the racetrack.The raceline provides the global fastest path around the complete racetrack.

camera based object detection for cones on the track [13]. With this setup the team achieved a velocity of 10 m/s while doing mapping, localization and planning at the same time with an RSME error of 0.29 m. On the same vehicle a recurrent neural network (RNN) was applied [175] to derive an accurate velocity estimation. By taking different vehicle sensors (e.g., IMU, wheel encoder) into account this learning based approach reached 15x better performance than an EKF approach with an RSME of $v_x = 0.141$ m/s and $v_y = 0.059$ m/s. Finally, for the IAC vehicle Sauerbeck *et al.* [168] provide a multi-sensor localization approach by focusing on decoupling the lateral and longitudinal position of the vehicle. With this approach the vehicle reached 300 km/h with an average deviation of 1.471 m in simulation. The current state of the art in autonomous racing is heavily based on single vehicle races. Therefore the subcategory of object detection algorithms for high speed applications was not given much attention. Nevertheless, in the FSD competition teams need to detect both color and form of cones to let the vehicle drive autonomously as depicted in Figure 3. In [47] a case study with different convolutional neural network (CNN) methods (Tiny-YOLO, Proteins) are done in comparison to a YOLO v2 setup [48] to display the best approach for cone detection in the FSD scenario. Strobel *et al.* [182] present a combination of a YOLO v3 based object detection, pose estimation, and time synchronization that uses data from both stereo and monocular cameras. Furthermore, besides these software focused developments the authors of [153] evaluated the performance and energy consumption of popular, off-the-shelf commercial devices for DNN inference in the formula student context. Finally, to help and support other FSD teams in their development, Vödisch *et al.* [49] presented a collaborative dataset for vision-based cone detection systems that is open-source available.

B. PLANNING

In the following subsection we cover algorithms that plan trajectories for the autonomous race vehicle to drive around the racetrack. Strategies that drive the vehicle end-to-end directly from perception to actuation is excluded from this part and is discussed with further details in Section II-D. We split the discussion into the three following parts.

(i) Global planning provides an optimal path, better known as *raceline* (depicted in Figure 4), around the racetrack. In the context of racing, global planning often optimize for the lowest lap time. Therefore, when following this raceline, the car drives an optimal path around the racetrack – under the constraints of the raceline generation – as fast as possible.

(ii) Local planning (or *motion planning*) plans on a finer granularity compared to global planning, usually under the assumption that an optimal global trajectory is provided. Local planners operate in a certain time horizon, and aim to avoid obstacles while still provide a fast and reliable path that does not deviate too much from the optimal global raceline.

(iii) Behavioral planning provides information about the high-level mission planning of the racecar. This can include the decision making about overtaking maneuvers (overtaking left/overtaking right/stay behind), the energy management strategy, interaction with other vehicles and the reaction to inputs from race control (e.g., flags, speed limits). As a summary, Table 2 provides an overview of research efforts in the field of planning for autonomous racing. We define the fundamental problems for autonomous racing planning as the following:

- Minimum-time optimization for a global optimal raceline.
- Long local planning horizon for recursive feasibility.
- Obstacle avoidance and vehicle reaction at high speeds.
- High re-planning frequency for real-time capability.
- Decision making under high uncertainty.
- Interaction planning with non-cooperative agents.

(i) Global Planning: Research from the field of global planning can be roughly divided into different strategies using the objectives of the overall optimization: lap time, geometric properties of the race lines, or energy spent. Racing, as a context for optimization, provides a clear measure of quality in lap time t_{lap} on participating agents. So naturally lowering t_{lap} is a popular choice when it comes to global planning (Figure 5).

TABLE 2. Overview of research in the field of autonomous racing planning.

Name and Reference	Year	Planning Category	Method	Tested on Hardware	Racing Series
Metz et al. [130]	1989	Global Planning	Near time optimal control	No	-
Braghin et al. [26]	2008	Global Planning	Optimization (Min. Curvature)	No	-
Kelly et al. [106]	2010	Global Planning	Time optimal control	No	-
Cardamone et al. [38]	2010	Global Planning	Optimization (Min. Curvature)	No	-
Quadflieg et al. [154]	2011	Global Planning	CMA-ES (Min. time)	No	-
Theodosis et al. [188]	2011	Global Planning	Multi-phase Geometric based	Yes	-
Theodosis et al. [189]	2012	Global Planning	Non-linear Optimization (Min. time)	Yes	-
Rucco et al. [163]	2015	Global Planning	Optimization (Min. time)	No	-
Bevilacqua et al. [25]	2017	Global Planning	Particle Swarm Optimization	No	-
Kuhn [113]	2017	Global Planning	Geometric methods	No	-
Dal Bianco et al. [45]	2018	Global Planning	Optimal Control (Min. time)	No	-
Heilmeier et al. [78]	2019	Global Planning	Optimization (Min. Curvature)	Yes	Roborace
Herrmann et al. [84], [85]	2019	Global Planning	Optimization (Min time+energy)	Yes	Roborace
Herrmann et al. [86]	2020	Global Planning	Optimization (velocity)	Yes	Roborace
Pagot et al. [143]	2020	Global Planning	NMPC (Min. time)	Yes	-
Vazquez et al. [196]	2021	Global Planning	Optimization (Min. time) + NMPC	Yes	FSD
Lovato et al. [127]	2021	Global Planning	Apex-finding + Optimal control	No	-
Butz et al. [32]	2009	Local Planning	CMA	No	-
Jeong et al. [94]	2013	Local Planning	RRT*	No	-
Liniger et al. [119]	2014	Local Planning	NMPC	Yes	1:43
Liniger et al. [120]	2015	Local Planning	Viability theory + MPC	Yes	1:43
Anderson et al. [12]	2016	Local Planning	MPC	No	-
Kapania et al. [102]	2016	Local Planning	Convex Optimization	Yes	-
Williams et al. [206]	2016	Local Planning	MPPI	Yes	Autorally
Buyal et al. [33]	2017	Local Planning	Nonlinear MPC (NMPC)	Yes	Roborace
Funke et al. [63]	2017	Local Planning	MPC	Yes	-
Arslan et al. [14]	2017	Local Planning	CL-RRT$^{\#}$	No	-
You et al. [219]	2018	Local Planning	Trail braking	No	-
Subosits et al. [183]	2019	Local Planning	Convex Optimization	Yes	-
Stahl et al. [178]	2019	Local Planning	Graph search + spline opt.	Yes	Roborace
Alcal et al. [9]	2020	Local Planning	LPV-MPC	No	-
Bulsara et al. [31]	2020	Local Planning	RRT+MPC	Yes	F1TENTH
Feraco et al. [57]	2020	Local Planning	RRT + Stanley	Yes	FSD
Srinivasan et al. [176]	2021	Local Planning	Holistically designed hierarchical controllers	Yes	FSD
Evans et al. [54], [55]	2021	Local Planning	TD3	No	F1TENTH
Kalaria et al. [233]	2021	Local Planning	NMPC	No	-
Reiter et al. [155]	2021	Local Planning	NMPC + MILP	Yes	Roborace
Brüdigam et al. [236]	2021	Local Planning	MPC + Gaussian	No	-
Bhargav et al. [234]	2021	Local Planning	MPC	No	-
You et al. [220]	2021	Local Planning	Trail braking	No	-
Wang et al. [237]	2021	Local Planning	MPC + Deep-Koopman	Yes	F1TENTH
Jung et al. [227]	2021	Local Planning	MPC + Game Theory	No	IAC
Loiacona et al. [126]	2010	Behavioral Planning	Reinforcement Learning	No	-
Kloeser et al. [108]	2020	Global+Local Planning	NMPC	No	-
O'Kelly et al. [141]	2020	Global+Local Planning	CMA-ES + Spline opt.	Yes	F1TENTH
Williams et al. [207]	2017	Local+Behavior Planning	Best response + MPPI	Yes	Autorally
Notomista et al. [137]	2020	Local+Behavior Planning	Iterated Best Response + CBF	No	-
Wang et al. [202]	2020	Local+Behavior Planning	Iterated Best Response	No	-
Sinha et al. [171]	2020	Local+Behavior Planning	EXP3 + Dist. robust opt.	Yes	F1TENTH
Liniger et al. [124]	2020	Local+Behavior Planning	Non-cooperative game	Yes	1:43
Wang et al. [201], [203]	2021	Local+Behavior Planning	Iterated Best Response + Spline opt.	Yes	-
Schwarting et al. [170]	2021	Local+Behavior Planning	Belief-space Planning	No	-

A first category of global planning approaches is the usage of variations of Evolutionary Algorithms (EAs) to optimize for lap times. This approach is used in [25], [141], [154] with different parameterizations of the search spaces. In these category, an individual "gene" models a complete configuration of the racing environment, and sometimes vehicle hardware and software. The algorithms also require evaluation functions to gauge the performance of an individual; here, this is the simulated lap time given a configuration. Initially, a pool of genes (referred to as a population) is created by sampling the search space. Then, in each iteration of the algorithm, genes are evaluated, and mutations are performed following different strategies depending on the specific algorithm. Eventually, the individuals in the population should converge, and a best configuration for the global optimal raceline is found.

Another popular approach when optimizing for lap times as the objective function is to form an optimal control problem (OCP), usually non-linear. The OCP uses the race vehicle's lap times as the objective function, and respects

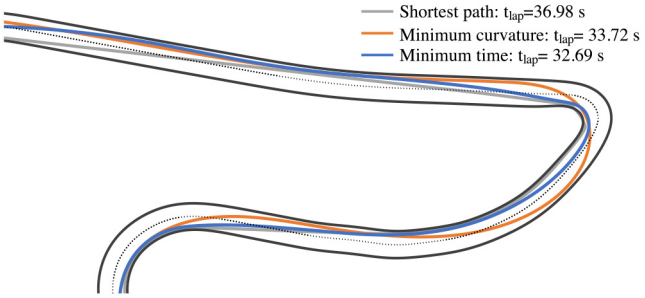

FIGURE 5. Comparison of global optimal raceline algorithms based on shortest path, minimum-curvature [78] and minimum-time [43] optimization, which lead to different lap times t_{lap} and trajectories.

constraints of the geometry and friction limits μ_{max} of the race track as well as the dynamics and control limits of the racecar.Proposed approaches usually choose different vehicle dynamic models, and solvers to solve the optimization problem. Metz and Williams [130] formulate an OCP where the objective is the lap time, and solutions are found by quasi-linearization with integral penalty functions, and splicing of constrained and unconstrained arcs to form a two-point boundary value problem. Kelly and Sharp [106] use Sequential Quadratic Programming (SQP) to solve the non-linear programming problem where lap time is the objective. Rucco *et al.* [163] formulate an OCP where the objective is the lap time. By reformulating the objective with eliminating explicit time step terms, the problem becomes a fixed-horizon free-endpoint problem. Projection operator-based Newton's method is ultimately used for the trajectory optimization. Theodosis and Gerdes [189] initialize the optimization problem with a path created by connecting center line on the straights and clothoids between the center lines. The sequential gradient based, non-linear optimization then uses the lap time as the cost function to find an improved global race line. Pagot *et al.* [143] present a non-linear model-predictive framework to formulate a optimal control problem where time is the objective. Vazquez *et al.* [196] formulate a minimum-time optimal control problem by using the centerline of the race track and discretize the continuous space dynamics. Two regularization terms with slip angle cost and a control input rate of change cost are also included in the objective function for reducing the lap time. Hermann *et al.* [84]–[86] formulate an optimal control problem where the lap time, the path, the velocity profile and the energy consumption is included in the objective function. By formulating a multi-parametric SQP it is possible to find the optimal velocity plan along a race line while not violating dynamic and energy requirements of the drivetrain. Finally, the authors of [127] devise a apex-finding method and calculates the optimal time global race line by solving an OCP.

Some researchers show approaches of calculating the race line by satisfying certain geometric properties. Usually, the assumption that vehicles experience lateral acceleration that results in lateral tire forces is made when it comes to autonomous racing. It is then often desirable to minimize the lateral acceleration to minimize the possibility of side slip. In these cases, the aim is to find a race line with the *minimum-curvature* overall. However, it is widely known that the least curvature path is not the ideal path for a racecar which is asymmetric in braking and acceleration and is operated in a combined slip range. Braghin *et al.* [26] create design and solve a dynamic problem to find the best compromise between the shortest track and the least curvature track based on the vehicle's dynamic behavior. Similarly, Cardamone *et al.* [38] also try to find the compromise between the same conflicting objectives, but use a Genetic Algorithm (GA) to find the best weighting parameter between the two objectives. Heilmeier *et al.* [78] extends the work of Braghin *et al.* [26] to solve a quadratic optimization problem where constraints on vehicle dynamics limits are set up to find the minimum-curvature path around the race track. This approach is compared to the minimum-time optimization of [43]. For the same racetrack [78] achieves a laptime of $t_{lap} = 86.13$ s while [43] is even faster with $t_{lap} = 84.90$ s. The advantage of the minimum-curvature planning is obviously the reduced set of parameters for the vehicle dynamics model and the faster calculation time.

Lastly, some approaches also choose to mimic the geometric properties of a race line driven by human drivers. These approaches often decomposes a turn into different segments and require race lines to satisfy different properties in each sections. Kuhn [113] mimics the behavior of a racecar driver by defining the same important decision making points on the track. It calculates the race line by first fixing the locations of the following points: the braking points, turn-in points, apex points, turn-out points, and accelerations points. Then a piecewise rational spline function is used to interpolate all the points and create a race line. Theodosis and Gerdes [188] mimic the three phase cornering technique used by professional drivers. It first finds all straights along the track, and linking the straights with curve structures. Then, each connecting curve is found by combining clothoids and a circular arc. The parameters are adjusted to minimize the overall curvature and ensure the path is tangent to the following straight.

(ii) Local Planning: In local planning, the main objective is to plan the cars motion for a fixed horizon by avoiding collisions with either the environment or adversaries (Figure 6). There are three main strategies:

1) Modifying the global plan via optimization.
2) Sample multiple dynamically feasible trajectories and select the best one around obstacles.
3) Sample in the free space around obstacles to find a feasible trajectory.

From the state of the art we could deduce that many authors use Model Predictive Control (MPC) methods for local trajectory planning. Although this method is control technique, it is also suitable for planning a local trajectory.

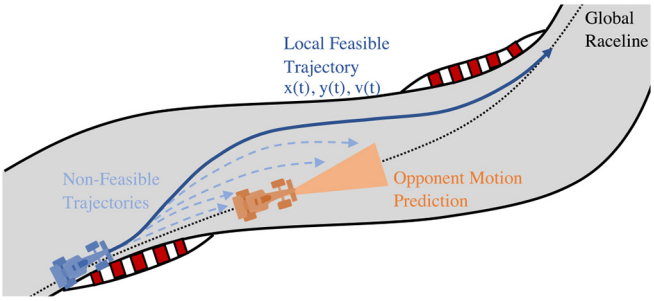

FIGURE 6. Local trajectory planning on the racetrack: The vehicle needs to prediction the opponents motion, plans a feasible trajectory around the opponent and stick closely to the global raceline.

In this section, we present work that is on the very thin and fluid boundary between planning and control, but mainly addresses the planning problem. Pure path tracking with MPC is described later in Section II-C.

In the first category, the global plan is modified to allow for obstacle avoidance. In these types of formulations, model predictive controllers are usually utilized to optimize the global plan. Upon encountering an obstacle or opponent vehicle, the constraints or cost functions of the optimization problem is modified, and a new motion plan is formulated. Anderson *et al.* [12] switch between two MPC modes to optimize for minimum-time objective or maximum velocity objective to mimic a professional driver. Kapania *et al.* [102] first find the optimal velocity profile given a reference path, and then updates the given path with the fixed velocity profile to find the minimum-curvature path by solving a convex optimization problem. Williams *et al.* [206] propose a sampling based MPC that relies on path integral control for entropy minimization. In [63] a planner is presented that is capable of mediating between conflicting objectives when performing collision avoidance, vehicle stabilization, and path tracking. Subosits and Gerdes [183] present a real-time trajectory planning algorithm by approximating a re-planning problem as a convex quadratically constrained quadratic problem (QCQP) with a simplified point-mass model. Alcalá *et al.* [9] reformulate the non-linear vehicle dynamics in a Linear Parameter Varying (LPV) form. This can then be used to create a convex optimization problem which is easier to solve for search for a local path. Kalaria *et al.* [233] use a nonlinear-MPC (NMPC) for local planning where the objective consists of progress along the race line, avoiding collision, and use drafting (reduce drag) to make progress. Brüdigam *et al.* [236] use a Gaussian Process (GP) to predict the opponent's maneuver. These stochastic information is then used in a Stochastic MPC to plan efficient overtaking maneuvers.

In the second category, multiple motion primitives, or prototype motion plans, are generated by forward simulating the vehicle dynamics given the current state of the vehicle using multiple different actuation input sequences. This usually results in multiple splines or arcs to select from. In addition, cost functions are used to give each primitive an attached value. With a search for the best (lowest/highest) cost in these primitives a final trajectory can then be chosen. Liniger *et al.* [119], [120] generate a library of trajectories by forward simulating the vehicle using a grid of vehicle velocities and steering angles up to a certain horizon. Stahl *et al.* [178] propagate the race track with a graph that covers the entire space. The nodes are first placed equidistantly along cross sections of the race track, then edges connecting nodes are created by optimizing for cubic clothoids. This planner was tested on the Roborace vehicle and achieved $v_{max} = 223$ km/h with an update rate of 16.8 Hz on and NVIDIA Arm electrical control unit (ECU). O'Kelly *et al.* [141] use a uniform grid of points along the global race line as local goal points. Afterwards cubic clothoids are optimized to connect the vehicle's current state and the grid points which leads ultimately to planning a local trajectory. Finally in [171] a set of local goal points in front of the vehicle is sampled with the help of a normalizing flow method. Again, cubic clothoids are optimized here to connect the vehicle's current state and the grid points to derive a driveable trajectory.

In the third category, sampling-based methods are used. These approaches randomly sample the free space around the current state of the vehicle for goal states. Once an available goal state is found, a motion plan is then generated connecting the current state of the vehicle with the selected goal state. By introducing randomness in the sampling process, these algorithms are usually efficient, but do not provide guarantees on their optimality. Jeon *et al.* [94] combine the rapidly-exploring random tree (RRT*) method with a local steering algorithm utilizing the dynamic model of the vehicle. Arslan *et al.* [14] combine RRT with closed-loop prediction based on the vehicle model, and incorporate relaxation methods for efficient construction of a tree that guarantees asymptotic optimality. Feraco *et al.* [57] combine RRT with Dubins curve to generate dynamically feasible local plans. Finally, Bulsara *et al.* [31] use RRT to find collision free reference paths in the free space.

(iii) Behavioral Planning: In behavior planning, the focus is usually on high-level decision making on tasks such as selecting an appropriate weighting of different objectives, or selecting plans that impedes the progress of opponents. The research in this area mainly focuses on two different strategies:

1) Assigning multiple cost functions with weighting and selecting the plan with the lowest overall cost.
2) Combine the local planner with game theoretic methods.

In the first category, cost functions are used that represent specific racing values like progress along the track, proximity to the obstacles, effort for control inputs and the deviation from optimal global plan. An overall cost is then found by combining all cost functions for each candidate local trajectory. Cost functions could also incorporate hard constraints by eliminating unqualified plans. Finally, the trajectory with

the minimum overall cost is chosen to be the local trajectory. Liniger *et al.* [119] use the prototype trajectory without collision and makes the largest progress along the track. This approach is extended in [120] by applying viability kernels on the track which only generates viable trajectories. O'Kelly *et al.* [141] assign cost functions representing proximity to the global plan, collision with the environment to prototype trajectories and select the best one. Finally, Sinha *et al.* [171] assign cost functions for progress along the track, overall curvature, maximum velocity, and collision with predicted opponent motion to all prototype trajectories.

In the second category, game theoretic approaches are used to usually find the best action in a two or multiple player game. The continuous motion planning problem is usually transformed into a step-by-step game where each player is allowed to make a "move" one by one. These approaches usually incorporate the concept of regret to try to find the best response for winning the racing game either immediately or in the long run. Williams *et al.* [207] combine a best response model of the opponent behavior with variation of MPC by including predicted opponent trajectories into the cost of other vehicles. Notomista *et al.* [137] propose sensitivity-enhanced Nash equilibrium seeking, which uses iterated best response algorithm to optimize for a trajectory in a two car racing game. In [202] a iterated best response with Nash equilibrium approximation is used to plan receding horizon trajectories. This technique helps to maximally advance the racecar along the track while taking into account opponent's intentions and responses. Sinha *et al.* [171] build a library of opponent prototypes offline and uses the EXP3 algorithm to solve for a multi-armed bandit problem to approximate the current opponent's driving policy by using the library. Liniger and Lygeros [124] repeat the multi-player game in a receding horizon fashion, which results in a sequence of coupled games. With this non-cooperative game approach the authors could show that the vehicles create blocking maneuvers although the risk of a collision gets higher. Wang *et al.* [203] use sensitivity enhanced iterated best response to seek convergence to the Nash equilibrium in the joint trajectory space for all agents. Finally, Schwarting *et al.* [170] use local iterative Dynamic Programming (DP) in belief space to solve a continuous Partially Observable Markov Decision Process (POMDP).

C. CONTROL

In the previous subsection, we discussed how to compute either a global or local trajectory on the racetrack. The trajectory includes both a path ($x(t)$, $y(t)$ - position) and velocity profile $v(t)$ which provides the reference information for the *lateral and longitudinal control*. In what follows, we provide control methodologies that leverage such a reference trajectory to compute control actions to navigate the car along the waypoints. The goal is to reduce the lateral and heading error to stay as close to the reference line and to reduce the velocity error to be as fast as possible (Figure 7).

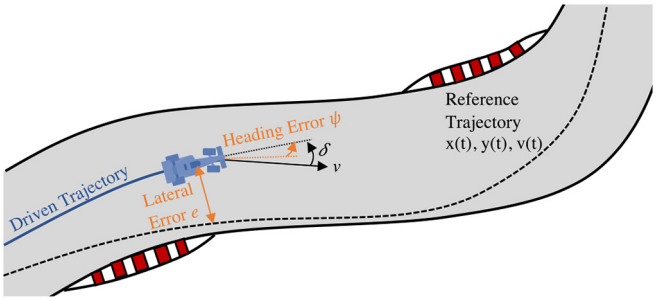

FIGURE 7. The goal of the control task in the autonomous racecar is to reduce lateral and heading error while following a global optimal raceline as a reference trajectory.

At this level of abstraction, the control actions are usually the steering angle δ and a throttle or brake request (acceleration commands a_{long}) that are sent to low-level controllers for actuating the motor and the brakes. We denote $x^{\text{ref}}(t)$ as the state associated with the reference trajectory at time t, and our goal is to design a controller policy π that, given the current state of the vehicle at time t, denoted as $x(t)$, and the reference state $x^{\text{ref}}(t)$, computes the control input $u(t)$.

Table 3 gives an overview of the papers that address the control problem for autonomous racing. The control papers try to address the issue of "handling at the limits" and follow the raceline/reference trajectory as accurate as possible. We define the fundamental problems for autonomous racing control as the following:

- Highly accurate path tracking for low lateral errors.
- Highly accurate path tracking for low heading errors.
- Highly accurate velocity tracking for fast lap times.
- Stable vehicle behavior at high accelerations.
- Exact modeling of the nonlinear vehicle behavior.
- High control frequency for real-time high speed driving.

Since the accurate control of the vehicle is relies heavily on precise vehicle dynamics modeling, we include the vehicle dynamics models used in each case. Authors use either a *point mass* model, a *bicycle* (single-track) model, or a *four-wheel* (double-track) model. Furthermore, to capture the lateral behavior, the authors use the bicycle model either with only the *kinematic* behavior (no lateral forces), *linear* behavior (linear tire forces based on cornering stiffness), or a *nonlinear* behavior (nonlinear tire forces based on the Pacejka Magic Formula or Fiala Tire). For a better overview and understanding of the different types of control research we categorized the papers into six subsections.

(i) Classic Control: First, we survey papers that cover classical and well-known principles in the field of path and velocity tracking. Ni and Hu [133] present a path following controller for their FSD vehicle. Their overall controller architecture consists of longitudinal, lateral and yaw controllers that operate the vehicle on a predefined G-G diagramm (maximum lateral and longitudinal acceleration.) In both [185] and [186] we see the usage of a simple lookahead controller that provides path tracking at

TABLE 3. Overview and categorization of research paper in the field of control for autonomous racing.

Name and Reference	Year	Control Category	Method	Vehicle Model	Tested on Hard-ware	Racing Series
Kritayakirana et al. [110]	2010	Classic	Feedforward Longitudinal Controller	Bicycle linear	Yes	-
Talvala et al. [186]	2011	Classic	Lookahead control	Bicycle nonlinear	Yes	-
Kritayakirana et al. [111], [112]	2012	Classic	COP feedforward and feedback steering	Bicycle nonlinear	Yes	-
Kapania et al. [100]	2015	Classic	Feedback-feedforward steering controller	Bicycle nonlinear	Yes	-
Park et al. [146]	2015	Classic	Convex optimization	Four Wheel	Yes	-
Laurense et al. [114], [115]	2017, 2018	Classic	Slip angle-based control strategy	Bicycle linear	Yes	-
Ni et al. [133], [134]	2017, 2019	Classic	Controller Framework	Bicycle kinematic	Yes	FSD
Fu et al. [59]	2018	Classic	Maximize GG Diagram	Four Wheel	Yes	FSD
Chatzikomis et al. [40]	2018	Classic	Torque Vectoring	Bicycle linear	Yes	Roborace
Wachter et al. [199]	2020	Classic	Controller Analysis	Four Wheel	No	-
Pedone et al. [148]	2020	Classic	Model-free nonlinear control	Bicycle linear	No	-
O Kelly et al. [141]	2020	Classic	Parameter Optimization with CMA-ES	Bicycle linear	No	F1TENTH
Sukhil et al. [185]	2021	Classic	Adaptive Lookahead for Pure Pursuit	-	Yes	F1TENTH
Beal et al. [19]	2012	MPC	Model Predictive Envelope Control	Bicycle linear; AFI	Yes	-
Williams et al. [206]	2015	MPC	Model Predictive Path Integral (MPPI)	Model agnostic	Yes	AutoRally
Carrau et al. [39]	2016	MPC	sparse Randomized MPC	Bicycle nonlinear	Yes	1:43 car
Verschueren et al. [197]	2016	MPC	Nonlinear MPC (NMPC)	Bicycle nonlinear	No	-
Drews et al. [51]	2017	MPC	MPPI + CNN	Model agnostic	Yes	AutoRally
Liniger et al. [121]	2017	MPC	sparse Randomized MPC	Bicycle nonlinear	No	1:43 car
Williams et al. [207], [209]–[211]	2017, 2018	MPC	MPC+ Game Theory, MPC + RL, Sampling based MPC	Model agnostic	Yes	AutoRally
Alrifaee et al. [10], [11]	2018, 2021	MPC	Sequential Convex Programming	Point Mass	No	-
Novi et al. [138]	2019	MPC	Hierarchial Nonlinear MPC (NMPC)	Four Wheel	No	-
Liniger et al. [123]	2019	MPC	MPC + Viability Theory	Bicycle nonlinear	Yes	1:43
Brown et al. [27]	2020	MPC	Nonlinear MPC (NMPC)	Bicycle nonlinear	Yes	-
Liu et al. [125]	2020	MPC	Standard MPC	Bicycle kinematic	No	FSD
Alcal et al. [8]	2020	MPC	Linear Parameter Varying MPC (LPV-MPC)	Bicycle linear	Yes	F1TENTH
Gandhi et al. [64]	2021	MPC	Robust MPPI (RMPPI)	Model agnostic	No	AutoRally
Pour et al. [152]	2021	MPC	Linear Parameter Varying MPC (LPV-MPC)	Bicycle nonlinear	No	-
Wischnewski et al. [216]	2021	MPC	Tube MPC (TMPC)	Point Mass	No	Roborace
Li et al. [228]	2021	MPC	NMPC + MIQP	Bicycle nonlinear	No	-
Kapania et al. [101]	2015	Learning	PD-ILC, Q-ILC	Bicycle nonlinear	Yes	-
Brunner et al. [30]	2017	Learning	Learning MPC (LMPC)	Bicycle linear	No	1:10
Rosolia et al. [159]–[162]	2017, 2020	Learning	Learning MPC (LMPC)	Bicycle nonlinear	No	-
Ji et al. [95]	2018	Learning	DNN + backstepping variable structure	Bicycle kinematic	Yes	FSD
Hewig et al. [88]	2018	Learning	NMPC + Gaussian Process (GP)	Bicycle nonlinear	No	1:43 car
Wagener et al. [200]	2019	Learning	Dynamic Mirror Descent (DMD) MPC	Model agnostic	No	AutoRally
Kabzan et al. [98]	2019	Learning	NMPC + Gaussian Process	Bicycle nonlinear	Yes	FSD
Wischnewski et al. [214] [215]	2019	Learning	Controller + Gaussian Process; recursive Least-Mean-Squares algorithm	Model agnostic	Yes	Roborace
Vallon et al. [193]	2020	Learning	Learning MPC (LMPC)	Bicycle nonlinear	No	-
Jain et al. [92]	2020	Learning	MPC + Gaussian Process	Bicycle nonlinear	No	F1TENTH
Kapania et al. [103]	2020	Learning	PD-ILC, Q-ILC	Bicycle nonlinear	Yes	-
Van Niekerk et al. [195]	2020	Learning	Receding Horizon Control + GP	Model agnostic	No	-
Xiao et al. [232]	2021	Learning	Controller DNN +	Model agnostic	No	F1TENTH
Voser et al. [198]	2010	Drifting	Controller Analysis	Bicycle nonlinear	Yes	-
Hindiyeh et al. [89]	2014	Drifting	Controller Framework	Bicycle nonlinear	Yes	-
Goh Jet al. [65]–[67]	2016-2019	Drifting	Controller Framework	Four Wheel; Bicycle nonlinear	Yes	-
Zubov et al. [226]	2019	Drifting	Controller Framework	Bicycle nonlinear	No	Roborace
Joa et al. [96]	2020	Drifting	Controller Framework	Bicycle nonlinear	No	-
Cai et al. [34]	2020	Drifting	Deep Learning	Model agnostic	No	-

high speeds event at the limits of tire adhesion. In [111] an autonomous racing controller is presented that uses the vehicle's centre of percussion (COP) to design a feedforward and feedback steering. This showed how to simplify the equations of motion and highlights the challenge of controlling a vehicle with highly saturated tires. Furthermore, a special focus on path tracking at the tire/vehicle limits is presented in the work of [59], [100], [112], [114]. While the research of [59] and [112] displays the usage of a G-G diagram to display controllers that operate the vehicle at the limits, Laurense and Gerdes [115] presents a slip angle-based control strategy to maintain the front tires at a certain slip angle to create the maximum tire forces. A special focus on longitudinal control (speed control) is shown in [110], [114], [148]. Laurense *et al.* [114] presents a control framework for full tire-force utilization with slip-angle based steering control, combined with explicit control of the path-tracking dynamics through longitudinal speed feedback to achieve a better path tracking. In [148] a model-free nonlinear controller for longitudinal speed control is presented.

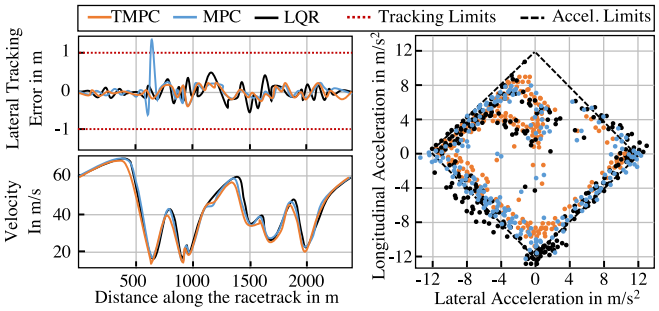

FIGURE 8. Qualitative comparison of an LQR, MPC and Tube MPC controller in an autonomous racing setup based on [216]. While the MPC is outperforming the LQR controller, it can be seen the Tube-MPC is not violating any dynamical constraints, e.g., the maximum lateral and longitudinal acceleration.

In this approach, a dynamic reconstruction of information on the vehicle's motion concerning the inputs acting on the system with sensor data is displayed. With this its possible to reconstruct the maximum longitudinal tire forces for current states which can be used for accurate speed tracking. Finally, in both [198] and [199] additional sensitivity analysis of path controlling at the limits and high sideslip maneuvers are displayed.

(ii) Model Predictive Control: The second and most-popular strategy used to realize an autonomous racing controller is *Model Predictive Control*. In MPC a sequence of control actions is computed by forecasting the future trajectory of the vehicle over a short time window. In particular, given the state of the system x_t, an MPC solves a *Finite Time Optimal Control Problem (FTOCP)* to compute an optimal sequence of states $\{x_t^*, \ldots, x_{t+N}^*\}$ and inputs $\{u_t^*, \ldots, u_{t+N-1}^*\}$ over a fixed horizon N. In autonomous racing the objective of the optimal control problem is to either track a global reference trajectory or to minimize the lap time. Upon computing such sequence of optimal states and actions, the first control action u_t^* is applied to the system and the process is repeated at the next time step based on the updated state x_{t+1}. MPC-based methodologies are the main method behind several autonomous racing controller which have been implemented on real vehicles. The advantages of MPC are that 1) forecast is used to act proactively and to 2) feedback is naturally incorporated in the controller that repeatedly updates the optimal trajectory. Notice that when the planning horizon N is short, the planned trajectory may not account for the future behavior of the system and as a result the controller may take shortsighted control actions. However, computing such quantities that exactly approximate the cost and constraint beyond the prediction is challenging. In practical applications it is preferred either to use a long prediction horizon [121] or to approximate these quantities based on historical data [162].

In [206] a sampling based MPC algorithm is derived. This so called *Model Predictive Path Integral Control (MPPI)* algorithm is using the methodology of path integral control that derives an optimal control based on stochastic sampling of trajectories. It is demonstrated that this approach explicitly provides a formula for the controls over the entire time horizon and that it relaxes the usual condition between control authority and noise required in path integral control. The authors use this fundamental control approach and enhance it with additional decision maker [64], game theory [209], DNNs [51] and reinforcement learning [208] methods to derive further improvements.

Carrau *et al.* [39] present at *sparse Randomized MPC* (SRMPC) approach that is based on a *Stochastic MPC*. This approach is used to deal with model uncertainty at high speeds and high accelerations. While driving with the vehicle it collects data along the track which is then used to identify the model uncertainty probabilistically. This tightens the constraints for the MPC automatically while still having the size and structure of a standard MPC problem. The optimization problem is solved in 20 ms for a 1:43 and the results show that for a desired violation the controller achieves faster lap times and fewer constraints violations than a standard MPC algorithm. Reference [121] is building on top of this approach and enhancing it with disturbance feedback policies to optimize over the state feedback matrices.

In order to capture the nonlinear dynamics of the vehicle and tires a *nonlinear MPC* (NMPC) can be designed and modeled [27]. Although this is computationally expensive, with the help of nonlinear optimization solver like FORCES PRO [244] these type of optimizations can be solved in real-time on the vehicle. In [138] a *hierarchical NMPC* is presented that consists of two controllers: Firstly, a high-level NMPC with point-mass model that simplifies the vehicle dynamics and is constraint by the tire G–G diagram. Secondly, a low-level NMPC with a high-fidelity model uses the output (velocity profile) of the first NMPC as a terminal constraint. This method helps to reduce the prediction horizon and therefore calculate the vehicle dynamics in real-time. Furthermore, this approach was improved in [196] with a simpler vehicle model to run on a FSD vehicle.

Li *et al.* [228] use a NMPC with a minimum-time objective and a collision-avoidance constraint. By applying a Mixed Integer Quadratic Programming (MIQP) method a control strategy is created that is optimized regarding the safety and the laptime. The authors conclude that for such an approach the prediction horizon needs to be large enough for creating feasible results although this leads to a higher computation time and is therefore non real-time capable. The authors of [123] combine a low-level MPC with a viability kernel that efficiently generates finite look-ahead trajectories to maximize the progress of the car. At the same time the viability kernel creates trajectories that remaining recursively feasible with respect to static obstacles. The authors apply this algorithm to both a simulation where the effects of various design choices and parameters are identified and showed that a hierarchical controller can be improved by

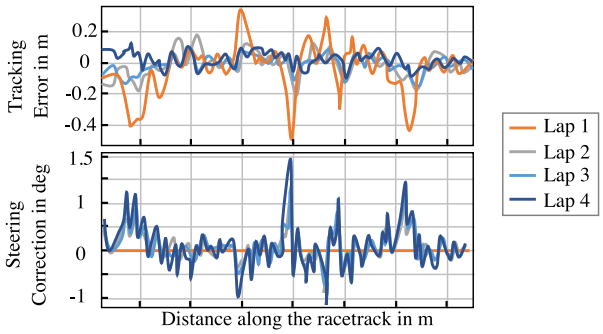

FIGURE 9. Qualitative demonstration of a learning-based control approach based on [101]. For each new lap the car is decreasing the tracking error by applying a higher correction steering angle for each vehicle movement.

incorporating the viability kernel in the trajectory planning phase.

Beal and Gerdes [19] use estimations of the friction coefficient and vehicle sideslip to define state constraint and unstable vehicle behaviors. This information is utilized in an model predictive envelope controller to create a region of stable vehicle motions. With this approach it is possible to operate the vehicle on the handling and stability limits. Similarly, Wischnewski *et al.* [216] and Williams *et al.* [211] present a *Tube-MPC* (TMPC) approach where nonlinear effects and external disturbances are taken into account of the MPC design. By approximating a tube of reachable sets over the prediction horizon the vehicle guarantees a space of constraint satisfaction. Finally, because tuning the parameters of a controller is time-consuming and needs experience from experts some researches try to automatically tune the parameters with optimization techniques. In [141] the super-optimization toolchain *TUNERCAR* is presented which is using a Covariance Matrix Adaptation Evolution Strategy (CMA-ES) to optimize both vehicle hardware parameters (center of gravity, mass) and control parameters (P-, I-, D-parameters) of the car. The evaluation is based on the laptime of the car and the algorithm shows the capability of reducing the laptime driven by the car by optimizing these parameters.

(iii) Learning Based Control: An additional control strategy to improve the tracking of a reference trajectory is to leverage *Iterative Learning Control (ILC)* based methodologies [6]. ILC methods are useful for applying them to an autonomous race vehicle, since the vehicle is running on the race track repeatedly for multiple laps. In this case the tracking error and vehicle data from previous laps can be used to compute a feed-forward correction term that improves the path and velocity tracking performance significantly (Figure 9).

ILC-based strategies for autonomous racing have been successfully implemented on full-size vehicles [101], [103]. In [30], [159]–[162] a *Learning MPC (LMPC)* is proposed which is an optimization and data-driven framework to make the car faster every lap and therefore reduce the lap time. The optimal control problem of the MPC is enhanced in a way so it tries to compute a solution by solving at time t of each lap the finite time constrained optimal control problem. This creates a convex optimization problem which can be solved with respective solvers. The authors show that the LMPC finds a faster trajectory for each new lap while maintaining the set constraints.

Hewing *et al.* [88] present an *learning-based cautious NMPC* which aims to learn from vehicle sensor data with Gaussian Processes to improve the vehicle dynamics model. The GP model is used for regression to identify uncertainties and a mismatch in the vehicle dynamics model parameters based on measurement data. The NMPC is extended with this learning model and reformulated in a stochastic setting which improves the performance and safety of the vehicle at the same time. Furthermore, the authors implement this approach on a FSD vehicle in [98] and demonstrate the implementation and experimental validation of this kind of learning-based control approach. Finally, Jain *et al.* [92] use a similar approach but only with an extended kinematic vehicle model for the MPC to prove that this type of learning-based control can also leverage the usage of simplified vehicle models.

Because of the control system quality and unmodeled effects, it is well known that there is a gap between the planned and the driven trajectory. This gap is unknown and depended on the environment the vehicle is driving in. To mitigate this gap, [214] presents a learning control approach on the method of Gaussian Process for a nonlinear regression. This GP learns online, while driving, how big this gap is and then tries close it over the time by using a so called scale-factor. This scale-factor serves as an optimization variable that tries to maximize longitudinal and lateral accelerations each lap.

Finally, the authors of [95] proposed a control scheme that consists of a robust steering controller and a DNN. While the path tracking is done via *backstepping variable structure control (BVSC)* the DNN is integrated to estimate nonlinear functions, e.g., the uncertainty of tire cornering stiffness.

(iv) Drifting Control: Although it is not following an optimal raceline, racing head-to-head or striving for the fastest laptime, the field of autonomous drifting is a special subcategory for autonomous racing. Here, the researchers show algorithms that are able to maneuver the car autonomously beyond the stable handling of limits and stabilize the car in a point of high slip angle. In [89] a successive loop structure is presented as a controller that tracks only the vehicles sideslip based on the yaw rate as a control input. Goh *et al.* [65]–[67] present a controller framework that is able to drift autonomously with the vehicle while tracking a predefined reference path. This enables drifting maneuvers at special references path, e.g., a circle or figure of 8. The authors using a single track vehicle model [65] and experiment with different variations of control values (sideslip [65], rotation of the vehicle's velocity vector to track the lateral error [66], [67]) to reach a sideslip angle of up to -40 degree with speeds up to 45 km/h on a real

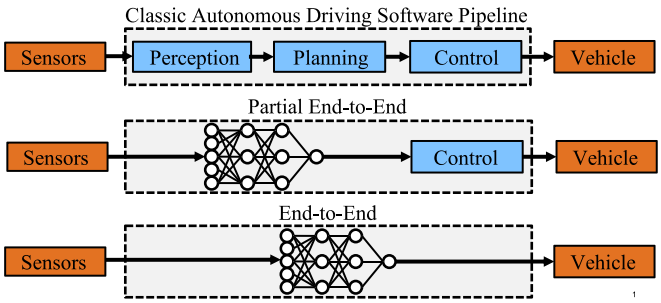

FIGURE 10. Classic autonomous driving software pipeline in comparison to partial and full end-to-end software pipeline.

test vehicle. Finally, Joa *et al.* [96] present a 3-level structure for a drift controller. First, they designed a supervisor which determines rate and rear longitudinal slip ratio for the drift maneuver. Second, a upper-level controller calculates lateral force (front) and longitudinal force (rear) for tracking the planned vehicle motion. Finally, a low-level controller converts the commands (forces) defined by the upper-level controller into control inputs for the vehicles (throttle, steering angle). With this setup the authors derived a steady state drifting on a real test vehicle.

D. END-TO-END DRIVING FOR AUTONOMOUS RACING

The previous subsections described in detail which research efforts in the fields of perception, planning and control have been achieved for autonomous racing. Besides that, researchers have tried to set up partial or full *end-to-end* approaches to master the autonomous racing task. As displayed in Figure 10, in the context of autonomous driving end-to-end means that either partial modules or all software modules are completely replaced with data-driven approaches like a DNN.

The partial end-to-end approach aims towards replacing or combining modules with a DNN. This has the advantage that the DNN provides a low-dimensional intermediate representation of the racetrack (e.g., a trajectory) that can be then used in a classic control systems (e.g., PID-Controller). In contrast to a full end-to-end system the final actuator output (steering angle, throttle position) is not predicted directly. On the one hand the field of autonomous racing provides a perfect proving ground for end-to-end approaches: clear driveable area, no signage (e.g., traffic lights), one class of objects, clear objective for training (fastest laptime). On the other hand, when using end-to-end systems large data sets are needed to train the DNNs. This data must contain a wide variety of situations so that the DNN achieves a good level of performance. Similar to other DNN applications the generalization and performance of these systems are the biggest issues. In addition to these common known issues, we define the following main problems for end-to-end autonomous racing:

- Open to question system architecture design: partial vs. full end-to-end.
- Difficulty to learn the vehicle dynamics parameter - especially the nonlinear vehicle and tire dynamics.
- Training purely in simulation environments lead to *simulation-to-reality* gap.
- High amount of various data necessary for training the artificial networks.
- Out of distribution events can cause drastic failure cases: Driving at high speeds is rare and thus learning how to correctly react is difficult

In the following we present research efforts that use end-to-end approaches for autonomous racing which are summarized in categories in Table 4.

Perez *et al.* [150] derives the control commands based on a rule-based evolutionary strategy. Although the car is driving successfully around the racetrack and follows a raceline the controller is only able to handle low speeds. In both [164], [166] the authors propose two fuzzy controllers for calculating the steering angle and computing the target speed of the car based on sensor information in the TORCS simulator [241]. In Oliveira *et al.* [142] Bayesian optimization (BO) is used to find a control policy that minimizes the time per lap while keeping the vehicle on the racetrack. The BO helps to search more efficiently over high-dimensional policy-parameter spaces an outperforms other evolutionary algorithms.

A solution for a partial end-to-end approach is presented in [204], [205], and [230]. The *DeepRacing Framework* is an end-to-end simulation environment and virtual testbed for training and evaluating algorithms especially for autonomous racing. In [205] three versions to control the racecar are presented: Pixel to control, pixel to waypoints, pixel to curves. It was shown that a partial end-to-end approach that provides parameterized trajectories based on a DNN outperforms a full end-to-end approach in terms of laptime and failures. Reference [116] shows the combination of MPC and CNNs to create a *perceptual attention-based predictive control algorithm*. With this, the MPC learns how to place attention on relevant areas of a visual input, which allows the vehicle to detect unsafe conditions faster. Drews *et al.* [52] are providing a framework that combines DNN based road detection as well as MPC to drive aggressively using only the sensor data from a monocular camera, IMU, and wheel speed sensors on the AutoRally vehicle. By combining CNNs and a Long Short Term Memory (LSTM) network the car is able to learn a local cost map representation of the track based on the camera input. This enables a global position estimation with a particle filter against a schematic map at high speeds. The local trajectory planning is afterwards done with the MPC. The authors of [128] provide an evaluation about the image sizes for a full end-to-end approach on a 1:10 scale vehicle. Based on their experiments the authors show that by decreasing the image size as an input for the end-to-end pipeline the car can drive faster and has a higher response time. Another evaluation for the usage of end-to-end algorithms is done by Wadekar *et al.* [235] in an

TABLE 4. Overview and categorization of additional software in the field of end-to-end approaches for autonomous racing.

Name and Reference	Year	End-to-End Category	Topic	Method	Tested on Hardware	Racing Series
Perez et al. [150]	2008	Optimization	Optimal Control Policy	Evolutionary Algorithm	No	-
Salem et al. [164]–[166]	2017, 2018	Optimization	Optimal Control Policy	Fuzzy Logic	No	-
Korkmaz et al. [109]	2018	Optimization	Optimal Control Policy	Fuzzy Logic	No	-
Oliveira et al. [142]	2018	Optimization	Vision based Planning	Bayesian Optimization	No	-
Lee et al. [116]	2019	Deep Learning	Vision based Planning	MPC + CNN	Yes	AutoRally
Weiss et al. [204], [205]	2020	Deep Learning	Vision based Planning	CNN, RNN	No	-
Tatulea et al. [187]	2020	Deep Learning	trajectory planning	NMPC & DNN	No	F1TENTH
Weiss et al. [230]	2021	Deep Learning	Trajectory Prediction	RNN	No	-
Drews et al. [52]	2019	Deep Learning	Localization	CNN, LSTM, + MPC	Yes	AutoRally
Mahmoud et al. [128]	2020	Deep Learning	Vision based Planning	CNN, LSTM	Yes	Donkey Car
Wadeka et al. [235]	2021	Deep Learning	Vision based Planning	CNN	No	IAC
Perot et al. [151]	2017	Reinforcement Learning	Vision based Planning	Advantage actor-critic	No	-
Jaritz et al. [93]	2018	Reinforcement Learning	Vision based Planning	Advantage actor-critic	No	-
De Bruin et al. [46]	2018	Reinforcement Learning	Vision based Planning	Q-Learning+ State representation Learning	No	-
Remonda et al. [156]	2019	Reinforcement Learning	Vision based Planning	DDPG	No	-
Niu et al. [135]	2020	Reinforcement Learning	Vision based Planning	DDPG	No	-
Gückiran et al. [72]	2019	Reinforcement Learning	Vision based Planning	SAC, Rainbow DQN	No	-
Fuchs et al. [60]	2021	Reinforcement Learning	Vision based Planning	SAC	No	-
Chisari et al. [42]	2021	Reinforcement Learning	Vision based Planning	SAC + policy output regularization	Yes	1:43 car
Lee et al. [117]	2021	Reinforcement Learning	Vision based Planning	Bayesian Deciscion Making	Yes	AutoRally
Pan et al. [145]	2021	Reinforcement Learning	Vision based Planning	Imitation Learning	Yes	AutoRally
Cai et al. [35]	2021	Reinforcement Learning	Vision based Planning	Imitation Learning	Yes	1:20 car
Schwarting et al. [169]	2021	Reinforcement Learning	Vision based Planning	Model based RL	No	-
Brunnbauer et al. [29]	2021	Reinforcement Learning	Vision based Planning	Model based RL	Yes	F1TENTH
Song et al. [172]	2021	Reinforcement Learning	Vision based Planning + Overtaking	SAC + 3-stage curriculum learning	No	-
Gundu et al. [74]	2019	Reinforcement Learning	Model-free optimal control	Q Learning + Soft-Actor Critic	No	-
Ivanov et al. [91]	2020	Reinforcement Learning	Verification	Variation of Algorithms	Yes	F1TENTH

simulation environment. The authors explored different data collection strategies with the goal of reaching high speeds and stable driving with the racecar. They conclude that even in the racetrack setup a high diversity and high amount of training data is necessary to achieve decent results.

Beside these plain usage of DNNs in the software pipeline, additional research efforts were done in the field of reinforcement learning. The autonomous racecar is seen as an agent that interacts with its environment in a continuous form. At each timestep t the agent fulfills an action a_t that leads to a reward r_t as well as an observations of all environment states s_t. Based on the reward r_t the agent tries to maximize the sum of the rewards over time and therefore can *learn* a specific behavior in this environment. Autonomous racing researchers that develop RL algorithms are using either the F1TENTH Gym [139], the Roborace Simulator [80], the SVL Simulator [75] or TORCS [241]. All these simulation environments have an openAI Gym [243] interface that was created for the setup of RL developments. Both [93] and [151] address the problem of autonomous racing by applying the method of Advantage Actor-Critic (A3C) to a simulation rally racing game. A complete framework for the training of the autonomous agents in a distributed system with different tracks and road conditions is presented as well as the RL method for achieving the end-to-end driving. They generate reasonable results with a fast and reliable vehicle maneuver, especially on different road conditions, but the approach fails to generalize well. To create a better generalization for different racetracks de Bruin *et al.* [46] provide a combination of Q-Learning and state representation learning to display that this combination learns policies quicker and generalize better to new racetracks then single RL. Both [135] and [156] use the method of Deep Deterministic Policy Gradient (DDPG) to explore the usage of RL in autonomous racing in the TORCS simulation. In both experimental setups DDPG is specially enhanced for the usage on the racetrack and shows good learning and execution results. The application of Soft Actor Critic (SAC) with enhancement and variations is displayed in [42], [72], [172]. While [72] is only using a simulation, Chisari *et al.* [42] are applying this approach to 1:43 small-scale vehicles and compare the SAC method to a MPC path planner – the MPC outperforms the RL method. The work from Fuchs *et al.* [60], [172] is using the racing game *Gran Turismo* as both a training and evaluation environment. While in [60] the framework for training the RL agents is presented Song *et al.* [172] uses and enhances this approach not only to drive with a single vehicle but also with multiple agents. They show that their RL agent is capable of driving fast, following the raceline and overtaking other agents without crashing. Finally, Schwarting *et al.* [169] and Brunnbauer *et al.* [29] present model-based reinforcement learning approaches which can learn competitive visual control policies through self-play in imagination (World Models idea [245]). Especially in [29]

it is shown that model-based RL approaches outperform non-model based methods. In addition, the authors display the sim-to-real transfer by testing the trained agent on a F1TENTH vehicle. The vehicle shows good generalization on unknown tracks but no high performance (e.g., low laptime) because of oscillating steering.

E. APPLIED AUTONOMOUS RACING STUDIES

In the final subsection all applied autonomous racing studies are displayed that clearly do not belong in the previous Sections II-A–II-D. These research papers provide *Evaluations* that run either simulator studies or overall analysis in the field of autonomous racing. In addition, the efforts of *Complete Software Stack* developments are displayed here. In order to achieve the vehicle's driving dynamics limits, there must be in-depth knowledge of the driving dynamics behavior and thus sufficiently good vehicle dynamic modeling. In the category of *modeling* the research that shows all vehicle dynamic modeling efforts for later usage in either trajectory planner or control algorithms is displayed. Finally, we present *Simulation* efforts and environments for autonomous racecars. A summary and overview of the research in those categories can be found in Table 5.

(i) Evaluations: To gain more knowledge in the field of racing different researchers conducted studies with race drivers or racecars. Kegelman *et al.* [105] did a study with real (vintage) racing cars and collected vehicle and position data from their runs on the racetracks. By examining the statistical dispersion of the vehicles race lines, the author displayed a quantification of the repeatability of professional racecar driver performance. In addition, they concluded that different driving styles (combination of path and velocity) can lead to similar lap times. In [167] the raceline trajectory information (dGPS data) from a test vehicle is collected to derive a path fitting algorithm that is describing a raceline. Based on this setup the race lines can be analyzed in-depth and results for autonomous raceline planning can be derived. A direct comparison between autonomous racecar against a human race driver is both done in [157] and [82]. Remonda *et al.* [157] conduct this study in a simulator environment and compared the lap times, telemetry data and the performance level of a human race driver against a pure autonomous racing software based on RL. By doing this the researchers were able to analyze which features have the most impact on the drivers performance. Those features where used afterwards to enhance the RL approach. In contrast, Hermansdorfer *et al.* [82] conduct a real world study by comparing an autonomous racing stack on the Roborace vehicle against a professional Formula 2 driver to find indications where the autonomous car fails to meet the performance level of the human race driver. The main reasons are that the human driver is detecting the vehicle limits (tire limits) more accurate, bringing the vehicle more often beyond the limit (higher slip angle) and is applying both brakes later and throttle earlier. Finally in [24] an evaluation about a crash of an autonomous racecar is displayed.

(ii) Complete Software Stack: Although many researchers are just deploying a single algorithm for testing and evaluation, to fully run an autonomous vehicle a complete software stack consisting of perception, planning and control algorithms is necessary. Therefore, many publications aim towards designing a holistic autonomous software stack that describes the individual software components, the methods, the transfer of messages from one module to the other and a final evaluation on real hardware or simulation. In [7], [41], [44], [61], [98], [132], [191], [221] the efforts of developing an autonomous racing software stack for FSD vehicles are presented. Based on the tasks in the FSD competitions (Section III) these cars need to map the environment, localize themselves, plan the path on the fly and follow the path fast and reliable. The teams provide different concepts to solve those individual tasks and display at the same time the underlying computation hardware of the their autonomous race vehicles. In addition, the teams provide insights in the middleware (e.g., ROS) as well as computations times of their algorithms. In contrast to the FSD efforts the publications [22], [23], [37] show their autonomous racing software stacks for running the Roborace vehicle. In [23] the research is aiming towards a software that can operate in a multi-vehicle scenario and therefore displays a dynamic local trajectory planner as a main component. In addition, to achieve high dynamic trajectory planning maneuvers the team displays a *Performance Assessment Module* that is observing the controller and the tires while adjusting parameters accordingly. Caporale *et al.* [37] display a holistic planning and control stack that has a real-time NMPC as main backbone to track a pre-planned racing line as well as a mapping and localization approach for high speed driving.

(iii) Modeling: The modeling of the vehicle dynamics behavior of the racecar is an essential part in the field of autonomous racing. Either these models are used in the simulation environments or model-based trajectory planning/control design approaches. The current state of the art provides many variations of vehicle dynamics modeling such as single track model, double track model or full vehicle model. The more complicated the vehicle dynamics model, the more parameters are needed. Unfortunately not all of those parameters are available in detail for a vehicle and so different methods for estimating these parameters are proposed - especially for nonlinear vehicle parameters like the tires. In [218] the design of the standard joint-state Unscented Kalman Filter (UKF) is presented to estimated vehicle dynamics parameters of a model car both in simulation and with experimental data. The experimental results show satisfactory estimates of the model parameters. Unfortunately the tuning process of this algorithm is time consuming and can only be implemented offline. Park and Gerdes [147] describe the region of feasible tire forces mathematically with constraints on the limits of actuation. They conclude in their work that with reasonable assumptions, the border of feasible tire forces can be displayed by an ellipse and circle for a wheel with steering and braking actuators.

TABLE 5. Overview and categorization of applied research and development studies in the field of autonomous racing.

Name and Reference	Year	Applied Studies Category	Topic/Methods	Tested on Hardware	Racing Series
Samper et al. [167]	2014	Evaluation	Path Analyzation	Yes	-
Kegelman et al. [105]	2016	Evaluation	Simulator Study	Yes	-
Remonda et al. [157]	2021	Evaluation	Simulator Study	No	-
Betz et al. [24]	2019	Evaluation	Crash Analysis	-	Roborace
Hermansdorfer et al. [82]	2020	Evaluation	Race Driver vs. Car	-	Roborace
Bak et al. [16]	2020	Evaluation	Path Planner Stress Tester	No	F1TENTH
Funke et al. [62]	2019	Complete Software Stack	Framework	Yes	Racecar
Culley et al. [44]	2019	Complete Software Stack	Framework	Yes	FSD
Funk et al. [61]	2017	Complete Software Stack	Framework	Yes	FSD
Tian et al. [190]	2018	Complete Software Stack	Framework	Yes	FSD
Chen et al. [41]	2019	Complete Software Stack	Framework	Yes	FSD
Zadok et al. [221]	2019	Complete Software Stack	Simulation	No	FSD
Kabzan et al. [98]	2020	Complete Software Stack	Framework	Yes	FSD
Nekkah et al. [132]	2020	Complete Software Stack	Framework	Yes	FSD
Tian et al. [191]	2020	Complete Software Stack	Framework	Yes	FSD
Betz et al. [22], [23]	2019	Complete Software Stack	Framework	Yes	Roborace
Caporale et al. [37]	2019	Complete Software Stack	Framework	Yes	Roborace
You et al. [218]	2017	Modeling	UKF	Yes	AutoRally
Williams et al. [212]	2019	Modeling	Deep Learning	Yes	AutoRally
Park et al. [147]	2017	Modeling	modeling	Yes	-
Spielberg et al. [173]	2019	Modeling	Deep Learning	Yes	-
Ignat et al. [90]	2020	Modeling	Deep Learning	Yes	Roborace
Hermansdorfer et al. [83]	2021	Modeling	Deep Learning	Yes	Roborace
Wymann et al. [241]	2005	Simulation	TORCS Simulator	No	-
Jiang et al. [4]	2021	Simulation	Carla Simulator	No	-
Guodong et al. [75]	2020	Simulation	SVL Simulator	Yes	F1TENTH, IAC
Babu et al. [15]	2020	Simulation	F1TENTH Simulator	Yes	F1TENTH
Stahl et al. [180]	2020	Simulation	Scenario Creation	-	Roborace
Herman et al. [80]	2021	Simulation	Roborace Simulator	Yes	Roborace

The papers [83], [90], [173], [212] are using learning based approaches by applying DNNs to identify the model parameters. All of these works show that DNNs can learn and identify the vehicle parameters more accurately than a purely parametric model. In addition, the researchers showed that the DNNs can generalize better than a purely non-parametric model especially when it comes to capturing the unknown dynamics. This makes the usage of DNNs ideal for real-world applications where collecting data from the full state space for a vehicle is not feasible and when different environment dynamics (e.g., icy road) need to be captured in the model.

(iv) Simulation: The final paragraph in this subsection displays all simulation efforts that have been done in the field of autonomous racing. Espié *et al.* [241] are the authors of *TORCS - The Open Racing Car Simulator*. This lightweight 3D simulator provides different race tracks, different cars, NPC opponents as well as a sophisticated vehicle physics model. This simulator is used for research in the field of control, trajectory planning, game theory and RL and is therefore providing a solution for autonomous race engineers. Roborace released its own simulation environment that is enhanced with an OpenAI Gym interface especially for classical control or RL tasks [80] which they called *Learn-to-Race (L2R)*. The simulation environment provides a racetrack, sophisticated vehicle physics simulation as well as a wide variety of sensors. The SVL Simulator [75] is a 3D end-to-end autonomous vehicle simulation platform that provides different maps, vehicles, sensor modeling, weather simulation, APIs to well-known open source software stacks (e.g., Autoware.Auto, Baidu Apollo) and the possibility of a distributed simulation. The SVL Simulator is offering both a 3D-vehicle model of the F1TENTH and the Indy Autonomous Challenge vehicle with different racetracks (Figure 11).

FIGURE 11. F1TENTH vehicle in the SVL simulator [75] on a 1:10 scale version of the Indianapolis Motor Speedway.

For the F1TENTH vehicle different simulation environment exist. Babu and Behl [15] present a ROS and Gazebo based autonomous racing simulator that is providing different maps, visualizations and model physics. The advantage here is access to the ROS community that enables the integration of robotics packages. Another F1TENTH simulator is the F1TENTH Gym [140] that provides a lightweight, 2D-simulation with an openAI Gym interface. Based upon

the Carla Simulator [242] the authors of [238] present an Autonomous System Operations (AutOps) and continuous integration (CI) and testing framework to evaluate the software in the context of autonomous racing. Especially for the evaluation of trajectory planning maneuvers in a multi vehicle environment Stahl and Betz [180] display an open-source graphical user interface that allows the fast generation of multi vehicle race scenarios. These dedicated scenarios (e.g., overtaking maneuvers) can be used to evaluate the trajectory planner or safety assessment algorithms in an autonomous racing stack.

III. AUTONOMOUS RACING HARDWARE: VEHICLES AND COMPETITIONS

The previous section gave an overview on the efforts in the field of algorithm and software development for autonomous racing vehicles. Almost all of the papers provided an evaluation of their proposed methods in an specific simulation environment. About half of the papers did additional evaluations on real vehicle hardware. This hardware is ranging from (powerful) passenger sports cars, specific research vehicle prototypes, small-scale race vehicles or real racing cars. In the following section we provide an overview of currently available hardware and racing competitions (Table 7) that are available for researchers.

A. SMALL-SCALE AUTONOMOUS RACING VEHICLES

The first type of autonomous racing vehicles are so called *small-scale* or *reduced-scale* vehicles. These racecars were mainly developed for the purpose of testing the new developed autonomous racing software. Those racecars are normally derived from remote controlled (RC) cars and therefore provide an electrical engine and a battery as a main powertrain unit. Those vehicles are then modified with additional hardware (sensors, ECUs), are constructed and maintained by a team of students and researchers and usually costs a few hundred to a few thousands of dollars. Although these are small-scale vehicles, they reach high speeds and accelerations for their size and therefore can be compared to real racecars.

(i) 1:43 vehicles: In the ORCA (Optimal RC Racing)[1] project researchers from the ETH Zurich developed a test bed consisting of a race track, an infrared camera based tracking system and modified 1:43 cars, in order to apply research in the field of MPC algorithms at high speeds and in real-time. A vision system captures the cars on the track and estimates both positions and velocity of each car. This information is then sent to a specific control platform where the MPC controller calculates the control inputs for the cars. This information is then sent via Bluetooth to the embedded cars where the control input is actuated. The research published with this 1:43 cars is heavily in the field of planning and control. As a result the researchers displayed new developments in the field of MPC [39], [88], [119]–[122], game theory [124] and reinforcement learning [42].

(ii) 1:10 vehicles: In the next bigger size researchers use modified 1:10 scale RC cars for their autonomous racing research. In the last years different institutions released their documentation for both hardware and setup of these 1:10 vehicles and so currently versions like *the Berkeley Autonomous Racecar*,[2] the *MIT Racecar* [104], the *MuSHR racecar* [174], the *RoSCAR* [76] or the *F1TENTH* [139], [140] vehicle from the University of Pennsylvania exists. The sensor setup on these cars is interchangeable and so it is possible to apply monocular cameras (e.g., Raspberry Pi, OpenCV OAK-1), stereo cameras (e.g., ZED, ZED2, Intel Realsense d435i, OpenCV OAK-D), 2D LiDARs (Hokuyo 10LX, Hokoyu 20LX), IMU, indoor GPS or wheelspeed sensors. As a main computation platform these vehicles use embedded GPU systems like the Nvidia Jetson (Models: TX1, TX2, NX, AGX Xavier, Nano). This gives the possibility to speed up the inference of DNNs. With the F1TENTH vehicle an additional, annual autonomous racing competition was launched where students, researchers and hobbyists can race against each other. The competitions consists of a single vehicle time trial and a head-to-head two vehicle race with knockout phase. In addition to these in-person competitions virtual competitions are organized to test the software of the developers. Similar competitions, where those kind of racecars or variations of it are used, are the *DiYRobocar* events or the Amazon *DeepRacer* [17], [18] competitions. The research published with these 1:10 cars is spread completely over all topics in perception [28], [70], planning [97], [232] and control [30], [31], [91], [92], [143], [162], [187], [237]. In recent years these type of vehicles got more important for optimization pipelines [141], [171], the application of RL techniques [29], [54], [55] and the evaluation of game theory methods [201], [203]. In addition, the 1:10 vehicles are used for education purposes [7], [15], [53], [104] to teach hands-on fundamentals of autonomous driving.

(iii) 1:5 vehicles: A special version of an autonomous small-scale vehicle is the so called *AutoRally* [68] vehicle, a 1:5 scale autonomous racecar developed by a team of researchers from Giorgia Tech. The AutoRally autonomous vehicle platform is based on a RC trophy truck (length: 1 m, width: 0.6 m, mass: 22 kg) with a top speed of $\sim$ 90 km/h. The AutoRally vehicle uses two monocular cameras (Point Grey Flea3 FL3-U3-13E4C-C color) as a main sensors to perceive the environment, has an IMU for acceleration measurements and hall-effect sensors to measure the wheel speeds. In addition, this vehicle has a GPS receiver (Hemisphere P307) integrated which provides an absolute position at 20 Hz with an accuracy of approximately 2 cm under ideal conditions with Real-Time Kinematic (RTK) corrections that are derived from a GPS base station. The main computation unit consists of standard consumer

1. https://control.ee.ethz.ch/research/team-projects/autonomous-rc-car-racing.html

2. www.barc-project.com/projects

TABLE 6. Autonomous racing hardware: overview over different available hardware and racing competitions available for researchers.

	F1TENTH[3]	EV Grand Prix Autonomous[4]	Formula Student Driverless[5]	Indy Autonomous Challenge[6]	Roborace[7]
Vehicle Image					
Vehicle Type	Small Scale 1:10	Reduced Scale 1:3	Reduced Scale 1:1.5	Real Racecar Indy Light Chassis	Real Racecar LMP Chassis
Vehicle Parameters	Length: 0.53 m Width: 0.28 m Mass: 3.5 kg	Length: 1.5 m Width: 1.4 m Mass: 110 kg	Vehicle parameters are based on the teams design choices	Length: 4.9 m Width: 1.9 m Mass: 750 kg	Length: 4.7 m Width: 2.0 m Mass: 1200 kg
Powertrain	Electrical Engine AWD Engine: 230 W Battery: 55 Wh	Electrical Engine RWD Engine: Teams choice Battery: Teams choice	Electrical Engine AWD/RWD Engine: Teams choice Battery: Teams choice	Combustion Engine RWD Engine: 335 kW 6 speed sequential	Electrical Engine RWD Engine: 270 kW Battery: 40 kWh
Maximum Speed	∼72 km/h	∼ 100 km/h Depends on components choices	∼ 120 km/h Depends on components choices	∼290 km/h	∼250 km/h
Sensor Setup	Monocular camera Stereocamera 2D LiDAR Indoor GPS	Sensor setup based on team choice: Monocular camera Stereocamera Radar 2D LiDAR 3D LiDAR (RTK) GPS	Sensor setup based on team choice: Monocular camera Stereocamera Radar 2D LiDAR 3D LiDAR (RTK) GPS	6x Monocular camera 4x Radar 3x 3D LiDAR (RTK) GPS	4x Monocular camera 2x Long Range Radar 2x Short Range Radar 5x 3D LiDAR (RTK) GPS
Computation Unit	Nvidia Jetson Nano Nvidia Jetson NX Nvidia Jetson AGX	Teams choice	Teams choice	Intel Xeon E 2278 GE – 3.30 GHz, 1x Nvidia Quadro RTX 8000, 64GB Ram	Nvidia Drive PX2 Speedgoat Mobile McLaren ECU
Software	ROS ROS2 Autoware.Auto	Teams choice	Teams choice	ROS2 Autoware.Auto	Teams choice
Competitons	Several competitions a year, competitions in different countries	One Race, USA only	Several competitions a year, competitions in different countries	Two Races, USA only	One championship with several races, USA & UK
Single/Multi Vehicle Race	Multi Vehicle 2 Cars	Single Vehicle	Single Vehicle	Single Vehicle Multi Vehicle (2 Cars)	Single Vehicle Multi Vehicle (2 Cars)
Real Race Competition Type	Time Trial Head to Head Racing	Time Trials	Acceleration Skidpad Autocross Trackdrive Efficiency Business, Cost, Design	Time Trial Overtaking Competition	Time Trial (with virtual objects)
Virtual Race	Yes	No	No	Yes	No
Simulation Environments	F1TENTH Gym F1TENTH Simulator SVL Simulator	-	Formula Student Driverless Simulator	Ansys Simulator SVL Simulator	Roborace Simulator
Related Papers	[7], [15], [28], [29], [31], [54], [55], [70], [91], [92], [97], [139]–[141], [171], [187], [203], [232]	-	[13], [41], [47]–[49], [57], [61], [69], [95], [98], [99], [118], [125], [132]–[134], [153], [176], [182], [190]–[192], [194], [196], [221], [222]	[168], [217], [227], [231], [235]	[3], [22]–[24], [33], [36], [37], [40], [43], [78], [80]–[86], [90], [129], [136], [144], [158], [177]–[181], [213]–[216], [224]–[226], [229]

computer components (Intel i7-6700 -3.4 GHz quad-core, 32 GB DDR4, Nvidia GTX-750ti SC) which are modular and reconfigurable and are all connected on a Mini-ITX motherboard.

This brings the AutoRally setup closer to real passenger vehicles and allows a high computation power. The research published with the AutoRally car is in the field of planning [14], [117], [207], [219], [220] and control [52], [64], [116], [200], [206], [208]–[212], state estimation [58], [218] and the application of deep neural networks for perception and planning [52], [145] with an overall special focus on low friction surfaces.

(iv) eV Grand Prix Autonomous: In 2021 a new racing competition called *eV Grand Prix Autonomous* started in the

3. www.f1tenth.org
4. www.evgrandprix.org/autonomous/
5. www.fsaeonline.com
6. www.indyautonomouschallenge.com
7. www.roborace.com

USA. Student teams need to acquire a standardized electric go-kart chassis and are then allowed to modify the vehicle. The teams can change the complete electric drivetrain and integrate new components, e.g., battery, electrical engine, inverter. In addition, the teams can choose their own sensor setup (camera, radar, LiDAR, (RTK) GPS, IMU) to create an autonomous vehicle. Furthermore, the teams develop the software that drives the autonomous go-kart around the racetrack. The current race setup consists of single vehicle time trial laps. Because the eV Grand Prix autonomous is still in its early stage there was no research published with these kind of vehicles so far.

(v) Formula Student Driverless: Since 2017 student teams can develop a driverless vehicle for the Formula Student Driverless competition. The students can choose on their own how to design and equip the vehicle with both powertrain or autonomous driving hardware. Therefore different setups with different computation platforms (e.g., consumer hardware, Nvidia Drive PX hardware, Nvidia Jetson hardware) and sensors setups (monocular cameras, 3D LiDAR) exist. The cars compete in different single vehicle competitions: Acceleration (driving 75m straight with standing start), skid pad (two congruent circles with a diameter of 18.25m), autocross (racing on closed loop track with unknown layout), track drive and efficiency (racing 10 laps on a track with additional efficiency scoring based on the consumed energy). In addition to these pure driving competitions the cars are then judged in an additional business plan (business idea of the vehicle), design (judgement of hardware and software) and cost (financial planning an manufacturing) competition.

The research published with the Formula Student cars is spread completely over all topics in object detection [47]–[49], [153], [182], localization [13], [69], [118], [175], planning [57], [176], [196] and control [59], [95], [98], [125], [133], [134] with a focus on holistic software pipelines [41], [50], [61], [99], [132], [190], [191], [194], [221], [222] with adjustments for the specialities of the FSD competition.

B. FULL-SCALE AUTONOMOUS RACE VEHICLES

The small-scale vehicles offer a low-cost and easy to set up platform for researchers. In addition, only a small space is needed to run the vehicles and therefore these kind of small scale vehicles are very attractive for research in the field of autonomous racing. Unfortunately because of the scaling there is still a mismatch between those vehicles and real racecars. This has not only to do with the performance (v_{max}, $a_{long,max}$, $a_{lat,max}$) but also with the kind of sensors or computation units these vehicles equipped with. Furthermore, a real racecar has a different dynamic behavior because of the stiffness of the chassis. Based on this some companies/institutions decided do develop real autonomous race vehicles which are explored in further detail. There was an additional development of an autonomous dragster [20], [21], the application of which, apart from in these papers, has not taken place elsewhere and is therefore not be considered in the further discussion.

(i) Roborace: Roborace is a U.K. based company that developed different autonomous racecars (Devbot 1.0, Devbot 2.0, Robocar). The Robocar was only used for internal Roborace events and both Devbot 1.0 and 2.0 where provided to interested university teams and companies for their research. The Devbot 2.0 is based on a Le Mans Prototype (LMP) chassis and is a rear wheel drive, fully electric racecar. The vehicle is equipped with camera, LiDAR and radar sensors and two main ECUs (Nvidia PX2, Speedgoat Mobile Target Machine) to run the autonomous software. The goal of this vehicle development efforts is to provide both a vehicle platform and an annual competition where teams can compete against each other. The teams only need to develop the software, the hardware setup is equal for all teams. In 2018 single vehicle time trials were executed, in 2019 the so called *Season Alpha* provided different race formats (single vehicle, multi vehicle, localization) on racetracks in Europe. In 2020/2021 Roborace *Season Beta* started with seven university teams competing against each other in single-vehicle races (time trials). A special software from Roborace called *Metaverse* provides virtual static and dynamic objects on the track that needed to be avoided by the teams - otherwise they get time penalties for hitting these objects. The university teams used the Roborace vehicles for their research and provided plenty of published papers in the field of localization and motion estimation [129], [158], [177], [213], [224], [229], mapping [136], [144], planning [36], [43], [78], [81], [85], [86], [177], [179], [181] and control [33], [40], [214]–[216], [226] as well as energy management [84], [85] and holistic software stack development [3], [22]–[24]. In addition, the Roborace vehicle was used to derive some new simulation [77], [79], [80] and scenario environments [180], vehicle dynamics modeling [83], [90] and autonomous racing benchmarks [82].

(ii) Indy Autonomous Challenge: In 2020 the *Indy Autonomous Challenge* (IAC) was launched as a successor of the DARPA Grand Challenge and DARPA Urban Challenge. The IAC racecar is based on an Indy Lights chassis and is a rear wheel drive racecar powered by a combustion engine with 6 gear sequential transmission. The IAC vehicle is equipped with camera, LiDAR and radar sensors for perception and has one main ECU to run the autonomous software. The IAC provides universities both the vehicle platform and several competition types. The teams only need to develop the software, the hardware setup is equal for all teams. As a main middleware ROS2 is used. The IAC challenge consists of a single vehicle race around the Indianapolis Motor Speedway in October 2021 and a two vehicle head-to-head race on the Las Vegas Motor Speedway in January 2022. The aim is to drive 290 km/h with those vehicles and therefore the teams need to develop a high performance autonomous software stack that executes perception, planning and control. Since the IAC competition just finished its competition

only a few papers [168], [217], [227], [231], [235] have been published so far.

IV. DISCUSSION AND CONCLUSION

In this section, we present short discussions on each area of research we have presented above. We try to point out the most predominant methods in each field and explain why these methods are popular and preferable for autonomous racing applications from the authors' perspective. Since not all the works surveyed here explicitly present their results numerically, and even when they do they are rarely on the same metrics. Furthermore, hardware used (i.e., small-scale v.s. full-scale vehicles) differ tremendously between each publication, and some approaches were only evaluated in simulation. Thus, the authors think that an explicit comparison of numerical values would lead to the reader's misunderstanding of each algorithm's overall performance. Instead of showcasing numerical analysis and direct comparison of results between each approach, we only compare approaches by displaying their advantages and disadvantages on a high level.

A. SOFTWARE PERFORMANCE

(i) Perception: In Autonomous Racing, we saw authors adapting mostly well-known algorithms from SLAM, computer vision, and deep learning for perception. These algorithms are modified and enhanced for usage on the racetrack. A common thread we see in perception for Autonomous Racing is that LiDARs and odometry are still preferred for almost all SLAM approaches. Although there is plenty of research on camera-only perception, and it is widely used on Tesla vehicles worldwide, it's effectiveness has yet to be demonstrated in autonomous racing. Similarly, for object detection, well-known methods such as YOLO and its variations are used. From the authors' perspective there is no preferable method in this field since most approaches are adaptation from software used on road vehicles. On contrary, as pointed out in Section V, we see the trend to develop high-speed localization and object detection methods. And with the recent development of Event-based Vision [5], it would a great candidate as a perception sensor in racing scenarios.

(ii) Planning: For global planning, the predominant approach is to set up an optimal control problem (OCP). The OCP not only considers the objective for the global planner (minimum lap time), but also considers hard constraints on the vehicle's dynamic behavior. The optimization produces competent racelines has a choice of several efficient solvers for fast computation. Since the global plan is generated offline, the higher computation time of these methods can be neglected.

For local planning, we recommend decoupling local trajectory generation and local trajectory tracking. In trajectory generation, one can use a nominal dynamic model that describe the vehicle's behavior under most conditions to create multiple dynamically viable local plans with different characteristics. Requirements and cost objectives can be baked into these trajectories thus path selection can be done efficiently. E.g., sampling-based trajectory generation. Then, when tracking the selected trajectory, an MPC could be used with up-to-date operating condition of the vehicle. The decoupling of trajectory generation, selection, and tracking also allows integration with RL and game-theoretic methods during trajectory selection.

(iii) Control: With the decoupling of planning and control, the predominant approach amongst surveyed work is to focus on high precision path and velocity tracking with a dedicated algorithm. Although classic controllers (e.g., PID controllers) can achieve a good level of control precision, MPC approaches captures and considers the vehicle's dynamic behavior in the control loop, and thus usually achieve higher control precision and accuracy. With the development of high performance solvers, (e.g., [244]), we see that Non-linear MPC can be solved in real time on real vehicle hardware with satisfying efficiency. Furthermore, learning-based approaches can address the limitation of traditional MPCs where the underlying predictive model can be updated online. However, it is yet to be shown the effectiveness of these approaches in a multi-vehicle competitive scenario.

(iv) Deep and Reinforcement Learning: In all the work surveyed in this field, only a small portion of algorithms are tested on real vehicles in realistic high speed environments. Among these algorithms implemented on the real vehicle, none of them consider safety and robustness constraints, which are crucial in the adversarial racing scenario. Nevertheless, RL and Deep Learning approaches show promising results in encoding the highly non-linear interaction and dynamics of a racing scenario. Together with the development of high-performance perception pipelines, the viability of these algorithms for multi-agent interaction is expected to improve significantly.

B. HARDWARE PERFORMANCE

This survey displays five different types of autonomous racing hardware. Additionally, researchers are using modified sports cars to apply their developed algorithms. Because the research in the field of autonomous racing is very much at the edge of the dynamic driving limits, we can state that the research work with actual vehicle applications better covers the potential of the respective algorithms. Furthermore, we conclude that the robustness of the algorithms demonstrated on real hardware reaches higher speeds and accelerations and displays higher robustness in contrast to the research that was shown purely in simulation.

Since all autonomous racing hardware displayed in this paper differs in their purpose, hardware setup, vehicle dynamics capabilities, and racing competition type, no conclusion about the most favorable hardware can be drawn. We are concluding that especially the small scale and reduced scale vehicle sizes are interesting for research labs since both their setup and operation are demanding less personal and financial resources.

C. FROM PASSENGER CARS TO RACING CARS

In all the software sections displayed, we saw researchers are using and improving algorithms from the field of robotics and regular passenger cars. This concludes that the transfer from normal autonomous passenger cars to autonomous racing cars is manageable and straightforward. One will see impressive results in the car's behavior by applying state-of-the-art autonomy algorithms. For example, the open-source autonomous driving software Autoware.Auto is used in different research papers as a baseline software stack. Unfortunately, none of the state-of-the-art algorithms is capable of driving fast near the vehicle's dynamic limits or executing high-overtaking maneuvers. To achieve this, an extension of the software is necessary with the methods surveyed in this paper.

D. FROM RACING CARS TO PASSENGER CARS

A direct transfer of software from autonomous race cars to autonomous passenger cars has not been done yet. This is because this field is relatively young and emerging and researchers have been focusing on driving on the racetrack only so far. This is mainly due to the fact that the context and objectives for an autonomous racing scenario are too different from a street scenario. For example, some areas like the global optimal trajectory planning are not entirely of interest for passenger cars with their current definition. Therefore, we recommend focusing on the transfer of software from the named subcategories. Currently, we see the most substantial transfer possibilities in control and planning. For example, the knowledge gained in the field of MPC is beneficial for regular passenger cars to create a highly accurate path and velocity tracking on our road networks. Additionally, the displayed work in the field of behavioral planning, e.g., with game-theoretic approaches, has a high potential for being integrated into passenger cars since modeling the interaction between different road users is of high research interest.

V. OPEN RESEARCH QUESTIONS AND CHALLENGES

In the previous sections we provided a detailed overview of all the efforts that have been made in the field of autonomous racing for both software and hardware. The goal of these research efforts is to contribute to the development of safer autonomous passenger vehicles and the possibility to derive knowledge for the development of new and advanced autonomous driving algorithms. Although the state of the art is quite extensive, there are still some open and unsolved research questions where the field of autonomous racing can support, help and leverage the development of future autonomous driving algorithms. Based on additional discussions with leading researchers in the field we present a list of challenges in the field that determine open research questions:

Challenge 1 - Autonomous high speed perception: None of the previous work addresses high speed object detection or provides detailed insights into different fusion techniques for high speed localization. The current state of the art presents standard SLAM or object detection methods that are then adapted to the field of autonomous racing. We are currently missing methods, techniques and algorithms that are especially made for high speed driving where increased motion blur occurs and sensor synchronization becomes more important. A reliable detection distance above 100m is required. This can be achieved by decreasing the computational delay, an enhancement in the sensor fusion performance (camera + Radar + LiDAR) and with an increase of the object detection quality. When it comes to vision-based localization we see successful research in the field of drone racing which can be adapted and applied to the field of autonomous racing. Besides that there is currently no public dataset for high speed driving. However, the availability of rich data is essential for the development of comprehensive perception algorithms for this speed range.

Challenge 2 - Multi-vehicle trajectory planning: Most of the papers surveyed focus on a single vehicle racing scenario and only a handful of researchers tried to address multi-vehicle scenarios (>3 vehicles). Dynamic local trajectory planning at high speeds with multiple vehicles (e.g., for overtaking) is difficult and not covered completely in the state of the art and displays therefore a grand challenge for future research. The trajectory planning method must be capable of finding a path in a non-convex environment that is collision free, recursive feasible, incorporating dynamic vehicle constraints and is executable in real-time. Both the path and the velocity must be planned while taking the vehicle dynamics into account to leverage the current tire performance of the vehicle. We see this either as a chance for creating new types of methods and algorithms for trajectory planning or as a test environment for new heuristics that decrease the computational heavy calculations.

Challenge 3 - Multi-vehicle interaction: The interaction with other vehicles is an essential part of racing especially when it comes to head-to-head racing (e.g., overtaking, blocking). This interaction is covered with game theory approaches in some of the work but is not explored extensively. This interaction provides the need for new prediction algorithms that can deal with the high uncertainty of the opponents movements/behavior in the less structured environment of the racetrack. Hence, the prediction of surrounding objects can not rely on lane information or traffic rules, but has to be based on a comprehensive understanding of interactive scenarios. Besides that, a fundamental aspect of future state prediction is that it is inherently stochastic, as agents cannot know each other's motivations, so multiple modalities have to be considered. We seek a model of the future that can provide both (1) a weighted set of discrete trajectories that covers the space of likely outcomes and (2) a closed-form evaluation of the likelihood of any trajectory. In addition, there are no sophisticated behavior planners that can derive critical interaction based maneuvers for competing in a race environment. The goal is to enable a tight coupling between reason about the influence of the surrounding agents on the ego systems trajectory

and maintaining full capabilities of the ego systems vehicle dynamics.

Challenge 4 - Adversarial driving: The racetrack enables the testing of the capabilities of an adversarial vehicle that is exploring and evaluating the risk of future actions by planning and performing high risk maneuvers. This kind of research enables knowledge for autonomous cars that need to operate in highly crowded multi vehicle and multi passenger scenarios while minimizing the possibility of a freezing robot problem. This research heavily includes the calculation of risk for a perceived environment, a risk evaluation as well as new high precision local behavioral and trajectory planning algorithms.

Challenge 5 - Real-time vehicle dynamics modeling: Autonomous vehicles that operate on the limits of handling need to have an exact knowledge about the current vehicle dynamics state. One crucial factor here is the tire which creates the road-vehicle contact (friction value) which is changing drastically with aerodynamics (downforce), road conditions (tarmac), weather conditions (rain, snow) and the current vehicle maneuver (load shift due to braking, acceleration). The high degree of model uncertainty due to external influences combined with strongly non-linear effects in tire and vehicle dynamics represents a challenge for both motion planning and control (e.g., MPC) algorithms. Available models approximate the vehicle dynamics to a certain degree but are computationally demanding, especially when it comes to tire models. Current research efforts try to calculate the dynamical behavior of the vehicle with the help of artificial neural networks to be computationally faster than classical physical models.

Challenge 6 - Balancing safety and performance: The current work is heavily exploring the limits of an autonomous vehicle from a software and hardware perspective with the goal to drive fast. When it comes to a racing scenario we have to make a trade off between safety (not crashing the car), high performance (staying close to the opponent), energy management and making decisions while not violating the handling limits (stay behind opponent, overtake in particular turn). This setup creates the need for software that explores the trade-off between safety and performance. This software is then coupled with motion and behavioral planners and decides which actions to take next. In addition, the current state of the art does not cover the safety aspects of the autonomous racecar in particular and therefore we have an open research area where algorithms need to be derived that evaluate and balance both safety and performance of the vehicle.

Challenge 7 - Autonomous racing regulations and rulebook: Although the community currently consists of many different racing series with different cars we have no agreement on the driving rules for autonomous racecars. Although this can be declared as less research and more a community effort, when it comes to autonomous driving a rulebook based definition for racecars could be helpful. This leads to general guidelines researchers can rely on when developing their algorithms. Ultimately, this leads to software that is compliant for specific racecar competitions.

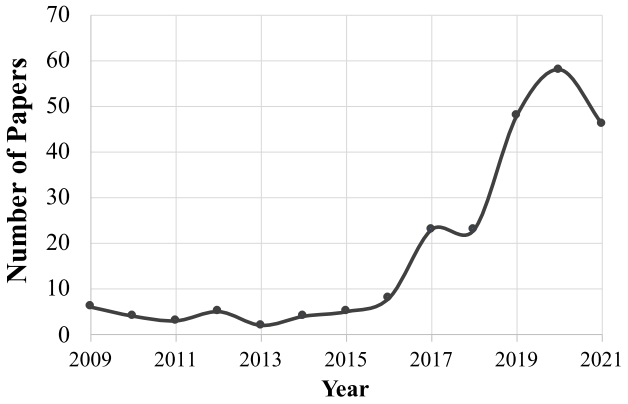

FIGURE 12. Evaluation: Published paper in the field autonomous racing from 2009 until the end of 2021.

Challenge 8 – Overall system software performance: Besides the development of particular algorithms for autonomous racing in each part of the software stack, the in-depth analysis of the overall software performance is a research field that is rarely covered. The synchronization of modules and the application of real-time conditions can reduce the overall latency significantly. Additionally, the delay of sensors and actuators influence both reaction time and vehicle performance. Unfortunately, these need to be considered in the development of a full software stack for autonomous racing. The optimal scheduling of software execution steps and an efficient management of the CPU and GPU usage by orchestration and hypervisor methods are additional research fields for the software development.

Challenge 9 - Autonomous high speed hardware: All platforms discussed in this survey rely on standard consumer hardware (e.g., ECUs, sensors). There is currently no specialized hardware that aims towards high speed driving or that was made particularly for autonomous racecars regarding computational demand as well as for vibration and shock resistance. Especially when we have a closer look to execution times displayed in the listed papers we see that some of the algorithms can be executed faster if particular hardware would exist.

VI. SUMMARY

This survey paper presents a comprehensive overview of the current state of the art in the field of autonomous vehicle racing. By discussing the previous and ongoing research efforts in this field we were able to demonstrate what kind of algorithms were developed to derive autonomous high speed driving on the racetrack. By splitting this paper into different sections for perception, planning and control we showed the individual achievements by researchers to establish the autonomous driving task for a racecar. Furthermore, we displayed and categorized research in the field of end-to-end algorithms, vehicle dynamics modeling and simulation

TABLE 7. List of abbreviations.

Abbreviation	Definition
A3C	Advantage Actor-Critic
AMCL	Adaptive Monte Carlo Localization
AWD	All Wheel Drive
BVSC	Backstepping Variable Structure Control
BO	Bayesian Optimization
CI	Continuous Integration
CMA	Covariance Matrix Adapation
CMA-ES	Covariance Matrix Adapation Evolutionary Strategy
CNN	Convolutional Neural Network
CPU	Central processing unit
COP	Center of Percussion
DDPG	Deep Deterministic Policy Gradient
DMD-MPC	Dynamic Mirror Descent MPC
DNN	Deep Neural Network
DP	Dynamic Programming
EA	Evolutionary Algorithm
ECU	Electrical Control Unit
EHF	H∞ Filter
EKF	Extended Kalman Filter
ES	Evolutionary Strategy
FSD	Formula Student Driverless
G-G	Maximal Lateral and Longitudinal Accelerations
GA	Genetic Algorithm
GP	Gaussian Process
GPS	Global Positioning System
GPU	Graphical Processing Unit
HD	High Definition
IAC	Indy Autonomous Challenge
ILC	Iterativ Learning Control
IMU	Inertial Measurement Unit
KF	Kalman Filter
L2R	Learn-to-Race
LMP	Le Mans Prototype
LMPC	Learning Model Predictive Control
LSTM	Long Short Term Memory
LPV	Linear Parameter-Varying
MIQP	Mixed Integer Quadratic Programming
MPC	Model Predictive Control
MPPI	Model Predictive Path Integral Control
NDT	Normal Distribution Transform
NMPC	Nonlinear Model Predictive Control
NPC	Non Playable Character
OCP	Optimal Control Problem
PID	Proportional Integral Derivative Controller
POMDP	Partially Observable Markov Decision Process
QCQP	Convex Quadratically Constrained Quadratic Problem
RC	Remote Controlled
ROS	Robot Operating System
RL	Reinforcement Learning
RRT	Rapidly-exploring random tree
RTK	Real-Time Kinematic
RNN	Recurrent Neural Network
RWD	Rear Wheel Drive
SAC	Soft Actor Critic
SQP	Sequential Quadratic Programming
SRMPC	sparse Randomized MPC
SSD	Single Shot Detection
UKF	Unscented Kalman Filter
TMPC	Tube MPC
SLAM	Simultaneous Localization and Mapping
YOLO	You Only Look Once (DNN)

environments. This survey aims towards a holistic review in the field of autonomous racing so additionally all hardware developments and autonomous racing platforms that are available are explained in detail. Some of these vehicles are associated with annual competitions that provide opportunities for researchers to test and evaluate the performance of their software. In total this survey is covering 237 papers in the field of autonomous vehicle racing. Furthermore, in the last four years we saw an increase of papers in this field (Figure 12).

Undoubtedly, the field of autonomous racing is an emerging field in intelligent vehicles, robotics and transportation system that is attracting more interest from researchers. Although we see an emphasis of research in the fields of planning & control, emerging fields like reinforcement learning for physical systems are applicable to autonomous racing. Based on these results we listed open research challenges for autonomous racing. This list can be used as a guideline for future researchers who can participate in the autonomous racing competitions. Finally, the list of papers surveyed in here are uploaded to a Github repository and updated on a regular basis so other researchers have an easy, open-source and structured access to the papers in the field of autonomous racing.

CONTRIBUTIONS AND ACKNOWLEDGMENT

Johannes Betz initiated the idea of this paper, created the overall structure and contributed to all sections of this survey paper. Hongrui Zheng contributed to the path planning section. Alex Liniger and Ugo Rosolia contributed to the path planning and control section. Phillip Karle contributed to the path planning section and open research challenges. Madhur Behl and Venkat Krovi revised the paper critically. Rahul Mangharam contributed to the overall structure of the paper, the open research challenges and revised the paper critically.

The authors would like to thank Todd Murphy (Northwestern University), Davide Scaramuzza (University of Zurich), Chris Gerdes (Stanford University), Markus Lienkamp (Technical University of Munich), Panagiotis Tsiotras (Georgia Institute of Technology) and Sertac Karaman (Massachusetts Institute of Technology) for their talks at 2021 IEEE ICRA 1st Workshop "Opportunities and Challenges with Autonomous Racing."[8] which contributed to the creation of Section V in this survey paper.

APPENDIX

See Table 7.

REFERENCES

[1] F. Doubek, F. Salzmann, and J. de Winter, "What makes a good driver on public roads and race tracks? An interview study," *Transp. Res. F Traffic Psychol. Behav.*, vol. 80, pp. 399–423, Jul. 2021. [Online]. Available: https://doi.org/10.1016/j.trf.2021.04.019

[2] I. R. Lima, S. Haar, L. D. Grassi, and A. A. Faisal, "Neurobehavioural signatures in race car driving: A case study," *Sci. Rep.*, vol. 10, no. 1, Jul. 2020, Art. no. 11537. [Online]. Available: https://doi.org/10.1038/s41598-020-68423-2

8. https://linklab-uva.github.io/icra-autonomous-racing

[3] J. Betz *et al.*, "What can we learn from autonomous level-5 motorsport?" in *Proc. Chassis.Tech*, Sep. 2018, pp. 123–146. [Online]. Available: https://doi.org/10.1007/978-3-658-22050-1_12

[4] A. Jiang, "Research on the development of autonomous race cars and impact on self-driving cars," *J. Phys. Conf.*, vol. 1, no. 1, Mar. 2021, Art. no. 012009. [Online]. Available: https://doi.org/10.1088/1742-6596/1824/1/012009

[5] G. Gallego *et al.*, "Event-based vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 154–180, Jan. 2022.

[6] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Syst. Mag.*, vol. 26, no. 3, pp. 96–114, Jun. 2006.

[7] A. Agnihotri, M. O'Kelly, R. Mangharam, and H. Abbas, "Teaching autonomous systems at 1/10th-scale," in *Proc. 51st ACM Tech. Symp. Comput. Sci. Educ.*, Feb. 2020, pp. 657–663. [Online]. Available: https://doi.org/10.1145/3328778.3366796

[8] E. Alcalá, V. Puig, J. Quevedo, and U. Rosolia, "Autonomous racing using linear parameter varying-model predictive control (LPV-MPC)," *Control Eng. Pract.*, vol. 95, Feb. 2020, Art. no. 104270. [Online]. Available: https://doi.org/10.1016/j.conengprac.2019.104270

[9] E. Alcalá, V. Puig, and J. Quevedo, "LPV-MP planning for autonomous racing vehicles considering obstacles," *Robot. Auton. Syst.*, vol. 124, Feb. 2020, Art. no. 103392. [Online]. Available: https://doi.org/10.1016/j.robot.2019.103392

[10] B. Alrifaee and J. Maczijewski, "Real-time trajectory optimization for autonomous vehicle racing using sequential Linearization," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun. 2018, pp. 476–483. [Online]. Available: https://doi.org/10.1109/ivs.2018.8500634

[11] B. Alrifaee, P. Scheffe, M. Kloock, and T. M. Henneken. "Sequential convex programming methods for real-time trajectory optimization in autonomous vehicle racing." Sep. 2021. [Online]. Available: https://doi.org/10.36227/techrxiv.16635205.v1

[12] J. R. Anderson, B. Ayalew, and T. Weiskircher, "Modeling a professional driver in ultra-high performance maneuvers with a hybrid cost MPC," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2016, pp. 1981–1986. [Online]. Available: https://doi.org/10.1109/acc.2016.7525209

[13] L. Andresen *et al.*, "Accurate mapping and planning for autonomous racing," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 4743–4749. [Online]. Available: https://doi.org/10.1109/iros45743.2020.9341702

[14] O. Arslan, K. Berntorp, and P. Tsiotras, "Sampling-based algorithms for optimal motion planning using closed-loop prediction," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 4991–4996. [Online]. Available: https://doi.org/10.1109/icra.2017.7989581

[15] V. S. Babu and M. Behl, "f1tenth.dev—An open-source ROS based F1/10 autonomous racing simulator," in *Proc. IEEE 16th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2020, pp. 1614–1620. [Online]. Available: https://doi.org/10.1109/case48305.2020.9216949

[16] S. Bak, J. Betz, A. Chawla, H. Zheng, and R. Mangharam, "Stress testing autonomous racing overtake maneuvers with RRT," 2021, *arxiv.abs/2110.01095*.

[17] B. Balaji *et al.*, "DeepRacer: Educational autonomous racing platform for experimentation with Sim2Real reinforcement learning," 2019, *arxiv.abs/1911.01562*.

[18] B. Balaji *et al.*, "DeepRacer: Autonomous racing platform for experimentation with Sim2Real reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 2746–2754. [Online]. Available: https://doi.org/10.1109/icra40945.2020.9197465

[19] C. E. Beal and J. C. Gerdes, "Model predictive control for vehicle stabilization at the limits of handling," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 4, pp. 1258–1269, Jul. 2013. [Online]. Available: https://doi.org/10.1109/tcst.2012.2200826

[20] J. Bell, V. Donikian, and G. Washington, "Modeling and design of an autonomous dragster," in *Proc. SAE*, Apr. 2020, p. 6. [Online]. Available: https://doi.org/10.4271/2020-01-0711

[21] J. Bell, "Modeling and design of an autonomous high speed 1/4 mile vehicle," Ph.D. dissertation, Dept. Comput. Sci., UC Irvine, San Francisco, CA, USA, 2020.

[22] J. Betz *et al.*, "A software architecture for an autonomous Racecar," in *Proc. IEEE 89th Veh. Technol. Conf. (VTC-Spring)*, Apr. 2019, pp. 1–6. [Online]. Available: https://doi.org/10.1109/vtcspring.2019.8746367

[23] J. Betz *et al.*, "A software architecture for the dynamic path planning of an autonomous Racecar at the limits of handling," in *Proc. IEEE Int. Conf. Connected Veh. Expo (ICCVE)*, Nov. 2019, pp. 1–8. [Online]. Available: https://doi.org/10.1109/iccve45908.2019.8965238

[24] J. Betz, A. Heilmeier, A. Wischnewski, T. Stahl, and M. Lienkamp, "Autonomous driving—A crash explained in detail," *Appl. Sci.*, vol. 9, no. 23, p. 5126, Nov. 2019. [Online]. Available: https://doi.org/10.3390/app9235126

[25] M. Bevilacqua, A. Tsourdos, and A. Starr, "Particle swarm for path planning in a racing circuit simulation," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf. (I2MTC)*, May 2017, pp. 1–6. [Online]. Available: https://doi.org/10.1109/i2mtc.2017.7969735

[26] F. Braghin, F. Cheli, S. Melzi, and E. Sabbioni, "Race driver model," *Comput. Struct.*, vol. 86, nos. 13–14, pp. 1503–1516, Jul. 2008. [Online]. Available: https://doi.org/10.1016/j.compstruc.2007.04.028

[27] M. Brown and J. C. Gerdes, "Coordinating tire forces to avoid obstacles using nonlinear model predictive control," *IEEE Trans. Intell. Veh.*, vol. 5, no. 1, pp. 21–31, Mar. 2020. [Online]. Available: https://doi.org/10.1109/tiv.2019.2955362

[28] A. Brunnbauer and M. Bader. "Traffic cone based self-localization on a 1:10 race car," 2019. [Online]. Available: https://openlib.tugraz.at/download.php?id=5d09dba11efec&location=medra

[29] A. Brunnbauer *et al.*, "Model-based versus model-free deep reinforcement learning for autonomous racing cars," 2021, *arxiv.abs/2103.04909*.

[30] M. Brunner, U. Rosolia, J. Gonzales, and F. Borrelli, "Repetitive learning model predictive control: An autonomous racing example," in *Proc. IEEE 56th Annu. Conf. Decis. Control (CDC)*, Dec. 2017, pp. 2545–2550. [Online]. Available: https://doi.org/10.1109/cdc.2017.8264027

[31] A. Bulsara, A. Raman, S. Kamarajugadda, M. Schmid, and V. N. Krovi, "Obstacle avoidance using model predictive control: An implementation and validation study using scaled vehicles," in *Proc. SAE*, Apr. 2020, p. 6. [Online]. Available: https://doi.org/10.4271/2020-01-0109

[32] M. V. Butz and T. D. Lonneker, "Optimized sensory-motor couplings plus strategy extensions for the TORCS car racing challenge," in *Proc. IEEE Symp. Comput. Intell. Games*, Sep. 2009, pp. 317–324. [Online]. Available: https://doi.org/10.1109/cig.2009.5286458

[33] A. Buyval, A. Gabdulin, R. Mustafin, and I. Shimchik, "Deriving overtaking strategy from nonlinear model predictive control for a race car," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 2623–2628. [Online]. Available: https://doi.org/10.1109/iros.2017.8206086

[34] P. Cai, X. Mei, L. Tai, Y. Sun, and M. Liu, "High-speed autonomous drifting with deep reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1247–1254, Apr. 2020. [Online]. Available: https://doi.org/10.1109/lra.2020.2967299

[35] P. Cai, H. Wang, H. Huang, Y. Liu, and M. Liu, "Vision-based autonomous car racing using deep imitative reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 7262–7269, Oct. 2021.

[36] D. Caporale *et al.*, "A planning and control system for self-driving racing vehicles," in *Proc. IEEE 4th Int. Forum Res. Technol. Soc. Ind. (RTSI)*, Sep. 2018, pp. 1–6. [Online]. Available: https://doi.org/10.1109/rtsi.2018.8548444

[37] D. Caporale *et al.*, "Towards the design of robotic drivers for full-scale self-driving racing cars," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 5643–5649. [Online]. Available: https://doi.org/10.1109/icra.2019.8793882

[38] L. Cardamone, D. Loiacono, P. L. Lanzi, and A. P. Bardelli, "Searching for the optimal racing line using genetic algorithms," in *Proc. IEEE Conf. Comput. Intell. Games*, Aug. 2010, pp. 388–394. [Online]. Available: https://doi.org/10.1109/itw.2010.5593330

[39] J. V. Carrau, A. Liniger, X. Zhang, and J. Lygeros, "Efficient implementation of randomized MPC for miniature race cars," in *Proc. Eur. Control Conf. (ECC)*, Jun. 2016, pp. 957–962. [Online]. Available: https://doi.org/10.1109/ecc.2016.7810413

[40] C. Chatzikomis, A. Sorniotti, P. Gruber, M. Zanchetta, D. Willans, and B. Balcombe, "Comparison of path tracking and torque-vectoring controllers for autonomous electric vehicles," *IEEE Trans. Intell. Veh.*, vol. 3, no. 4, pp. 559–570, Dec. 2018. [Online]. Available: https://doi.org/10.1109/tiv.2018.2874529

[41] T. Chen, Z. Li, Y. He, Z. Xu, Z. Yan, and H. Li, "From perception to control: An autonomous driving system for a formula student driverless car," 2019, *arxiv.abs/1909.00119*.

[42] E. Chisari, A. Liniger, A. Rupenyan, L. V. Gool, and J. Lygeros, "Learning from simulation, racing in reality," 2021. *arXiv:2011.13332*.

[43] F. Christ, A. Wischnewski, A. Heilmeier, and B. Lohmann, "Time-optimal trajectory planning for a race car considering variable tyre-road friction coefficients," *Veh. Syst. Dyn.*, vol. 59, no. 4, pp. 588–612, Dec. 2019. [Online]. Available: https://doi.org/10.1080/00423114.2019.1704804

[44] J. Culley *et al.*, "System design for a driverless autonomous racing vehicle," in *Proc. 12th Int. Symp. Commun. Syst. Netw. Digit. Signal Process. (CSNDSP)*, Jul. 2020, pp. 1–6. [Online]. Available: https://doi.org/10.1109/csndsp49049.2020.9249626

[45] N. D. Bianco, E. Bertolazzi, F. Biral, and M. Massaro, "Comparison of direct and indirect methods for minimum lap time optimal control problems," *Veh. Syst. Dyn.*, vol. 57, no. 5, pp. 665–696, Jun. 2018. [Online]. Available: https://doi.org/10.1080/00423114.2018.1480048

[46] T. de Bruin, J. Kober, K. Tuyls, and R. Babuska, "Integrating state representation learning into deep reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1394–1401, Jul. 2018. [Online]. Available: https://doi.org/10.1109/lra.2018.2800101

[47] N. D. Rita, A. Aimar, and T. Delbruck, "CNN-based object detection on low precision hardware: Racing car case study," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun. 2019, pp. 647–652. [Online]. Available: https://doi.org/10.1109/ivs.2019.8814001

[48] A. Dhall, D. Dai, and L. V. Gool, "Real-time 3D traffic cone detection for autonomous driving," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun. 2019, pp. 494–501. [Online]. Available: https://doi.org/10.1109/ivs.2019.8814089

[49] N. Vödisch, D. Dodel, and M. Schötz, "FSOCO: The formula student objects in context dataset," in *Proc. SAE Int.*, vol. 5, 2022, p. 6.

[50] T. H. Drage, J. Kalinowski, and T. Bräunl, "Development of an autonomous formula SAE car with laser scanner and GPS," *IFAC Proc. Vol.*, vol. 47, no. 3, pp. 2652–2657, 2014. [Online]. Available: https://doi.org/10.3182/20140824-6-za-1003.01386

[51] P. Drews, G. Williams, B. Goldfain, E. A. Theodorou, and J. M. Rehg, "Aggressive deep driving: Model predictive control with a CNN cost model," 2017, *arxiv.abs/1707.05303*.

[52] P. Drews, G. Williams, B. Goldfain, E. A. Theodorou, and J. M. Rehg, "Vision-based high-speed driving with a deep dynamic observer," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1564–1571, Apr. 2019. [Online]. Available: https://doi.org/10.1109/lra.2019.2896449

[53] S. Eken *et al.*, "A reproducible educational plan to teach mini autonomous race car programming," *Int. J. Elect. Eng. Educ.*, vol. 57, no. 4, pp. 340–360, Feb. 2020. [Online]. Available: https://doi.org/10.1177/0020720920907879

[54] B. Evans, H. A. Engelbrecht, and H. W. Jordaan, "From navigation to racing: Reward signal design for autonomous racing," 2021, *arxiv.abs/2103.10098*.

[55] B. Evans, H. W. Jordaan, and H. A. Engelbrecht, "Autonomous obstacle avoidance by learning policies for reference modification," 2021, *arxiv.abs/2102.11042*.

[56] D. Falanga, S. Kim, and D. Scaramuzza, "How fast is too fast? The role of perception latency in high-speed sense and avoid," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1884–1891, Apr. 2019. [Online]. Available: https://doi.org/10.1109/lra.2019.2898117

[57] S. Feraco, S. Luciani, A. Bonfitto, N. Amati, and A. Tonoli, "A local trajectory planning and control method for autonomous vehicles based on the RRT algorithm," in *Proc. AEIT Int. Conf. Elect. Electron. Technol. Autom. (AEIT AUTOMOTIVE)*, Nov. 2020, pp. 1–6. [Online]. Available: https://doi.org/10.23919/aeitautomotive50086.2020.9307439

[58] A. Foris, N. Wagener, B. Boots, and A. Mazumdar, "Exploiting singular configurations for controllable, low-power friction enhancement on unmanned ground vehicles," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3556–3563, Apr. 2020. [Online]. Available: https://doi.org/10.1109/lra.2020.2977266

[59] M. Fu, J. Ni, X. Li, and J. Hu, "Path tracking for autonomous race car based on G–G diagram," *Int. J. Autom. Technol.*, vol. 19, no. 4, pp. 659–668, Jun. 2018. [Online]. Available: https://doi.org/10.1007/s12239-018-0063-7

[60] F. Fuchs, Y. Song, E. Kaufmann, D. Scaramuzza, and P. Durr, "Super-human performance in Gran Turismo sport using deep reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 4257–4264, Jul. 2021. [Online]. Available: https://doi.org/10.1109/lra.2021.3064284

[61] N. Funk *et al.*, "Autonomous electric race car design," 2017, *arxiv.abs/1711.00548*.

[62] J. Funke *et al.*, "Up to the limits: Autonomous Audi TTS," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2012, pp. 1–6. [Online]. Available: https://doi.org/10.1109/ivs.2012.6232212

[63] J. Funke, M. Brown, S. M. Erlien, and J. C. Gerdes, "Collision avoidance and stabilization for autonomous vehicles in emergency scenarios," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 4, pp. 1204–1216, Jul. 2017. [Online]. Available: https://doi.org/10.1109/tcst.2016.2599783

[64] M. S. Gandhi, B. Vlahov, J. Gibson, G. Williams, and E. A. Theodorou, "Robust model predictive path integral control: Analysis and performance guarantees," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1423–1430, Apr. 2021. [Online]. Available: https://doi.org/10.1109/lra.2021.3057563

[65] J. Y. Goh and J. C. Gerdes, "Simultaneous stabilization and tracking of basic automobile drifting trajectories," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun. 2016, pp. 597–602. [Online]. Available: https://doi.org/10.1109/ivs.2016.7535448

[66] J. Goh, T. Goel, J. C. Gerdes, "A controller for automated drifting along complex trajectories," in *Proc. 14th Int. Symp. Adv. Veh. Control (AVEC)*, 2018. [Online]. Available: https://ddl.stanford.edu/sites/g/files/sbiybj9456/f/marty_avec2018_fullpaper.pdf

[67] J. Goh, T. Goel, and J. C. Gerdes, "Toward automated vehicle control beyond the stability limits: Drifting along a general path," *J. Dyn. Syst. Meas. Control*, vol. 142, no. 2, Nov. 2019, Art. no. 021004. [Online]. Available: https://doi.org/10.1115/1.4045320

[68] B. Goldfain *et al.*, "AutoRally: An open platform for aggressive autonomous driving," *IEEE Control Syst.*, vol. 39, no. 1, pp. 26–55, Feb. 2019. [Online]. Available: https://doi.org/10.1109/mcs.2018.2876958

[69] N. Gosala *et al.*, "Redundant perception and state estimation for reliable autonomous racing," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 6561–6567. [Online]. Available: https://doi.org/10.1109/icra.2019.8794155

[70] A. Gotlib, K. Lukojc, and M. Szczygielski, "Localization-based software architecture for 1:10 scale autonomous car," in *Proc. Int. Interdiscipl. Ph.D. Workshop (IIPhDW)*, May 2019, pp. 1–8. [Online]. Available: https://doi.org/10.1109/iiphdw.2019.8755418

[71] F. Gottmann, H. Wind, and O. Sawodny, "On the influence of rear axle steering and modeling depth on a model based racing line generation for autonomous racing," in *Proc. IEEE Conf. Control Technol. Appl. (CCTA)*, Aug. 2018, pp. 846–852. [Online]. Available: https://doi.org/10.1109/ccta.2018.8511508

[72] K. Guckiran and B. Bolat, "Autonomous car racing in simulation environment using deep reinforcement learning," in *Proc. Innov. Intell. Syst. Appl. Conf. (ASYU)*, Oct. 2019, pp. 1–6. [Online]. Available: https://doi.org/10.1109/asyu48272.2019.8946332

[73] I. Gundlach and U. Konigorski, "Modellbasierte online-Trajektorienplanung Für zeitoptimale Rennlinien," *Automatisierungstechnik*, vol. 67, no. 9, pp. 799–813, Sep. 2019. [Online]. Available: https://doi.org/10.1515/auto-2019-0032

[74] P. K. Gundu, K. G. Vamvoudakis, and R. M. Gerdes, "An intermittent learning algorithm for high-speed autonomous driving in unknown environments," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2019, pp. 4286–4292. [Online]. Available: https://doi.org/10.23919/acc.2019.8814462

[75] G. Rong *et al.*, "SVL simulator: A high fidelity simulator for autonomous driving," May 2020, *arXiv:2005.03778*.

[76] K. Hart *et al.*, "RoSCAR," in *Proc. Workshop Mobile Augmented Reality Robot. Technol. Syst. (MARS)*, 2014, pp. 3–8. [Online]. Available: https://doi.org/10.1145/2609829.2609837

[77] A. Heilmeier, M. Graf, and M. Lienkamp, "A race simulation for strategy decisions in circuit motorsports," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2986–2993. [Online]. Available: https://doi.org/10.1109/itsc.2018.8570012

[78] A. Heilmeier, A. Wischnewski, L. Hermansdorfer, J. Betz, M. Lienkamp, and B. Lohmann, "Minimum curvature trajectory planning and control for an autonomous race car," *Veh. Syst. Dyn.*, vol. 58, no. 10, pp. 1497–1527, Jun. 2019. [Online]. Available: https://doi.org/10.1080/00423114.2019.1631455

[79] A. Heilmeier, M. Geisslinger, and J. Betz, "A quasi-steady-state lap time simulation for electrified race cars," in *Proc. 14th Int. Conf. Ecol. Veh. Renew. Energies (EVER)*, May 2019, pp. 1–10. [Online]. Available: https://doi.org/10.1109/ever.2019.8813646

[80] J. Herman *et al.*, "Learn-to-race: A multimodal control environment for autonomous racing," in *Proc. ICCV*, 2021, pp. 9773–9782.

[81] L. Hermansdorfer, J. Betz, and M. Lienkamp, "A concept for estimation and prediction of the tire-road friction potential for an autonomous Racecar," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 1490–1495. [Online]. Available: https://doi.org/10.1109/itsc.2019.8917024

[82] L. Hermansdorfer, J. Betz, and M. Lienkamp, "Benchmarking of a software stack for autonomous racing against a professional human race driver," in *Proc. 15th Int. Conf. Ecol. Veh. Renew. Energies (EVER)*, Sep. 2020, pp. 1–8. [Online]. Available: https://doi.org/10.1109/ever48776.2020.9242926

[83] L. Hermansdorfer, R. Trauth, J. Betz, and M. Lienkamp, "End-to-end neural network for vehicle dynamics modeling," in *Proc. 6th IEEE Congr. Inf. Sci. Technol. (CiSt)*, Jun. 2020, pp. 407–412. [Online]. Available: https://doi.org/10.1109/cist49399.2021.9357196

[84] T. Herrmann, F. Christ, J. Betz, and M. Lienkamp, "Energy management strategy for an autonomous electric Racecar using optimal control," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 720–725. [Online]. Available: https://doi.org/10.1109/itsc.2019.8917154

[85] T. Herrmann, F. Passigato, J. Betz, and M. Lienkamp, "Minimum race-time planning-strategy for an autonomous electric Racecar," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–6. [Online]. Available: https://doi.org/10.1109/itsc45102.2020.9294681

[86] T. Herrmann, A. Wischnewski, L. Hermansdorfer, J. Betz, and M. Lienkamp, "Real-time adaptive velocity optimization for autonomous electric cars at the limits of handling," *IEEE Trans. Intell. Veh.*, vol. 6, no. 4, pp. 665–677, Dec. 2021. [Online]. Available: https://doi.org/10.1109/tiv.2020.3047858

[87] T. Herrmann, F. Sauerbeck, M. Bayerlein, J. Betz, and M. Lienkamp, "Optimization-based real-time-capable energy strategy for autonomous electric race cars [SAEIndexTestSampleV2 2022-01-17T10:06:32.798262Z]," *SAE Int. J. Connected Autom. Veh.*, vol. 5, no. 1, p. 10, Jan. 2022.

[88] L. Hewing, A. Liniger, and M. N. Zeilinger, "Cautious NMPC with Gaussian process dynamics for autonomous miniature race cars," in *Proc. Eur. Control Conf. (ECC)*, Jun. 2018, pp. 1341–1348. [Online]. Available: https://doi.org/10.23919/ecc.2018.8550162

[89] R. Y. Hindiyeh and J. C. Gerdes, "A controller framework for autonomous drifting: Design, stability, and experimental validation," *J. Dyn. Syst. Meas. Control*, vol. 136, no. 5, Jul. 2014, Art. no. 051015. [Online]. Available: https://doi.org/10.1115/1.4027471

[90] I. Georgiev, C. Chatzikomis, T. Völkl, J. Smith, and M. N. Mistry. "Iterative semi-parametric dynamics model learning for autonomous racing." 2020. [Online]. Available: https://arxiv.org/abs/2011.08750

[91] R. Ivanov, T. J. Carpenter, J. Weimer, R. Alur, G. J. Pappas, and I. Lee, "Case study," in *Proc. ACM 23rd Int. Conf. Hybrid Syst. Comput. Control*, Apr. 2020, pp. 1–7. [Online]. Available: https://doi.org/10.1145/3365365.3382216

[92] A. Jain, M. O'Kelly, P. Chaudhari, and M. Morari, "BayesRace: Learning to race autonomously using prior experience," in *Proc. CORL*, 2020, pp. 1918–1929.

[93] M. Jaritz, R. de Charette, M. Toromanoff, E. Perot, and F. Nashashibi, "End-to-end race driving with deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2070–2075. [Online]. Available: https://doi.org/10.1109/icra.2018.8460934

[94] J. H. Jeon *et al.*, "Optimal motion planning with the half-car dynamical model for autonomous high-speed driving," in *Proc. Amer. Control Conf.*, Jun. 2013, pp. 188–193. [Online]. Available: https://doi.org/10.1109/acc.2013.6579835

[95] X. Ji, X. He, C. Lv, Y. Liu, and J. Wu, "Adaptive-neural-network-based robust lateral motion control for autonomous vehicle at driving limits," *Control Eng. Pract.*, vol. 76, pp. 41–53, Jul. 2018. [Online]. Available: https://doi.org/10.1016/j.conengprac.2018.04.007

[96] E. Joa, H. Cha, Y. Hyun, Y. Koh, K. Yi, and J. Park, "A new control approach for automated drifting in consideration of the driving characteristics of an expert human driver," *Control Eng. Pract.*, vol. 96, Mar. 2020, Art. no. 104293. [Online]. Available: https://doi.org/10.1016/j.conengprac.2019.104293

[97] A. Joglekar, B. Deshpande, M. Basuthakur, and V. N. Krovi, "Fusing Offline and online trajectory optimization techniques for goal-to-goal navigation of a scaled autonomous vehicle," in *Proc. SAE*, Apr. 2021, p. 10. [Online]. Available: https://doi.org/10.4271/2021-01-0097

[98] J. Kabzan, L. Hewing, A. Liniger, and M. N. Zeilinger, "Learning-based model predictive control for autonomous racing," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3363–3370, Oct. 2019. [Online]. Available: https://doi.org/10.1109/lra.2019.2926677

[99] J. Kabzan *et al.*, "AMZ driverless: The full autonomous racing system," *J. Field Robot.*, vol. 37, no. 7, pp. 1267–1294, Aug. 2020. [Online]. Available: https://doi.org/10.1002/rob.21977

[100] N. R. Kapania and J. C. Gerdes, "Design of a feedback-feedforward steering controller for accurate path tracking and stability at the limits of handling," *Veh. Syst. Dyn.*, vol. 53, no. 12, pp. 1687–1704, Jun. 2015. [Online]. Available: https://doi.org/10.1080/00423114.2015.1055279

[101] N. R. Kapania and J. C. Gerdes, "Path tracking of highly dynamic autonomous vehicle trajectories via iterative learning control," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2015, pp. 2753–2758. [Online]. Available: https://doi.org/10.1109/acc.2015.7171151

[102] N. R. Kapania, J. Subosits, and J. C. Gerdes, "A sequential two-step algorithm for fast generation of vehicle racing trajectories," *J. Dyn. Syst. Meas. Control*, vol. 138, no. 9, Jun. 2016, Art. no. 091005. [Online]. Available: https://doi.org/10.1115/1.4033311

[103] N. R. Kapania and J. C. Gerdes, "Learning at the racetrack: Data-driven methods to improve racing performance over multiple laps," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8232–8242, Aug. 2020. [Online]. Available: https://doi.org/10.1109/tvt.2020.2998065

[104] S. Karaman *et al.*, "Project-based, collaborative, algorithmic robotics for high school students: Programming self-driving race cars at MIT," in *Proc. IEEE Integr. STEM Educ. Conf. (ISEC)*, 2017, pp. 195–203. [Online]. Available: https://doi.org/10.1109/isecon.2017.7910242

[105] J. C. Kegelman, L. K. Harbott, and J. C. Gerdes, "Insights into vehicle trajectories at the handling limits: Analysing open data from race car drivers," *Veh. Syst. Dyn.*, vol. 55, no. 2, pp. 191–207, Nov. 2016. [Online]. Available: https://doi.org/10.1080/00423114.2016.1249893

[106] D. P. Kelly and R. S. Sharp, "Time-optimal control of the race car: A numerical method to emulate the ideal driver," *Veh. Syst. Dyn.*, vol. 48, no. 12, pp. 1461–1474, Aug. 2010. [Online]. Available: https://doi.org/10.1080/00423110903514236

[107] T. Kessler, K. Esterle, and A. Knoll, "Linear differential games for cooperative Behavior planning of autonomous vehicles using mixed-integer programming," in *Proc. 59th IEEE Conf. Decis. Control (CDC)*, Dec. 2020, pp. 4060–4066. [Online]. Available: https://doi.org/10.1109/cdc42340.2020.9304495

[108] D. Kloeser, T. Schoels, T. Sartor, A. Zanelli, G. Prison, and M. Diehl, "NMPC for racing using a singularity-free path-parametric model with obstacle avoidance," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 14324–14329, 2020. [Online]. Available: https://doi.org/10.1016/j.ifacol.2020.12.1376

[109] B. Korkmaz, U. B. Etlik, A. Beke, and T. Kumbasar, "Fuzzy logic based self-driving racing car control system," in *Proc. 6th Int. Conf. Control Eng. Inf. Technol. (CEIT)*, Oct. 2018, pp. 1–6. [Online]. Available: https://doi.org/10.1109/ceit.2018.8751900

[110] K. Kritayakirana and J. C. Gerdes, "Autonomous cornering at the limits: Maximizing a 'G–G' diagram by using feedforward trail-braking and throttle-on-exit," *IFAC Proc. Vol.*, vol. 43, no. 7, pp. 548–553, Jul. 2010. [Online]. Available: https://doi.org/10.3182/20100712-3-de-2013.00060

[111] K. Kritayakirana and J. C. Gerdes, "Using the centre of percussion to design a steering controller for an autonomous race car," *Veh. Syst. Dyn.*, vol. 50, no. S1, pp. 33–51, Jan. 2012. [Online]. Available: https://doi.org/10.1080/00423114.2012.672842

[112] K. Kritayakirana and J. C. Gerdes, "Autonomous vehicle control at the limits of handling," *Int. J. Veh. Auton. Syst.*, vol. 10, no. 4, p. 271, 2012. [Online]. Available: https://doi.org/10.1504/ijvas.2012.051270

[113] P. W. Kuhn, "Methodology for the numerical calculation of racing lines and the virtual assessment of driving behavior for training circuits for the automobile industry," *Transp. Res. Procedia*, vol. 25, pp. 1416–1429, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352146517304593

[114] V. A. Laurense, J. Y. Goh, and J. C. Gerdes, "Path-tracking for autonomous vehicles at the limit of friction," in *Proc. Amer. Control Conf. (ACC)*, May 2017, pp. 5586–5591. [Online]. Available: https://doi.org/10.23919/acc.2017.7963824

[115] V. Laurense and C. Gerdes, "Speed control for robust path-tracking for automated vehicles at the tire-road friction limit," in *Proc. AVEC 14th Int. Symp. Adv. Veh. Control*, 2018, pp. 1–8. [Online]. Available: https://ddl.stanford.edu/publications/speed-control-robust-path-tracking-automated-vehicles-tire-road/-friction-limit

[116] K. Lee, G. N. An, V. Zakharov, and E. A. Theodorou, "Perceptual attention-based predictive control," in *Proc. CoRL*, 2019, pp. 220–232.

[117] K. Lee, Z. Wang, B. I. Vlahov, H. K. Brar, and E. A. Theodorou, "Ensemble Bayesian decision making with redundant deep perceptual control policies," in *Proc. ICMLA*, 2019, pp. 831–837.

[118] N. LE Large, F. Bieder, and M. Lauer, "Comparison of different SLAM approaches for a driverless race car," *Technisches Messen*, vol. 88, no. 4, pp. 227–236, Mar. 2021. [Online]. Available: https://doi.org/10.1515/teme-2021-0004

[119] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1:43 scale RC cars," *Opt. Control Appl. Methods*, vol. 36, no. 5, pp. 628–647, Jul. 2014. [Online]. Available: https://doi.org/10.1002/oca.2123

[120] A. Liniger and J. Lygeros, "A viability approach for fast recursive feasible finite horizon path planning of autonomous RC cars," in *Proc. 18th Int. Conf. Hybrid Syst. Comput. Control*, Apr. 2015, pp. 1–10. [Online]. Available: https://doi.org/10.1145/2728606.2728620

[121] A. Liniger, X. Zhang, P. Aeschbach, A. Georghiou, and J. Lygeros, "Racing miniature cars: Enhancing performance using stochastic MPC and disturbance feedback," in *Proc. Amer. Control Conf. (ACC)*, May 2017, pp. 5642–5647. [Online]. Available: https://doi.org/10.23919/acc.2017.7963833

[122] A. Liniger, "Path planning and control for autonomous racing," Ph.D. dissertation, Dept. Comput. Sci., ETH Zürich, Zürich, Switzerland, 2018. [Online]. Available: http://hdl.handle.net/20.500.11850/302942

[123] A. Liniger and J. Lygeros, "Real-time control for autonomous racing based on viability theory," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 2, pp. 464–478, Mar. 2019. [Online]. Available: https://doi.org/10.1109/tcst.2017.2772903

[124] A. Liniger and J. Lygeros, "A noncooperative game approach to autonomous racing," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 3, pp. 884–897, May 2020. [Online]. Available: https://doi.org/10.1109/tcst.2019.2895282

[125] Y. Liu, L. Shi, W. Xu, X. Xiong, W. Sun, and L. Qu, "Design of driverless racing chassis based on MPC," in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2020, pp. 1–9. [Online]. Available: https://doi.org/10.1109/cac51589.2020.9326482

[126] D. Loiacono, A. Prete, P. L. Lanzi, and L. Cardamone, "Learning to overtake in TORCS using simple reinforcement learning," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2010, pp. 1–8. [Online]. Available: https://doi.org/10.1109/cec.2010.5586191

[127] S. Lovato and M. Massaro, "Three-dimensional fixed-trajectory approaches to the minimum-lap time of road vehicles," *Veh. Syst. Dyn.*, pp. 1–18, Aug. 2021. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/00423114.2021.1969024?journalCode=nvsd20

[128] Y. Mahmoud, Y. Okuyama, T. Fukuchi, T. Kosuke, and I. Ando, "Optimizing deep-neural-network-driven autonomous race car using image scaling," in *Proc. SHS Web Conf.*, vol. 77, 2020, Art. no. 04002. [Online]. Available: https://doi.org/10.1051/shsconf/20207704002

[129] F. Massa, L. Bonamini, A. Settimi, L. Pallottino, and D. Caporale, "LiDAR-based GNSS denied Localization for autonomous racing cars," *Sensors*, vol. 20, no. 14, p. 3992, Jul. 2020. [Online]. Available: https://doi.org/10.3390/s20143992

[130] D. Metz and D. Williams, "Near time-optimal control of racing vehicles," *Automatica*, vol. 25, no. 6, pp. 841–857, Nov. 1989. [Online]. Available: https://doi.org/10.1016/0005-1098(89)90052-6

[131] M. Montani, L. Ronchi, R. Capitani, and C. Annicchiarico, "A hierarchical autonomous driver for a racing car: Real-time planning and tracking of the trajectory," *Energies*, vol. 14, no. 19, p. 6008, Sep. 2021. [Online]. Available: https://doi.org/10.3390/en14196008

[132] S. Nekkah *et al.*, "The autonomous racing software stack of the KIT19d," 2020, *arxiv.abs/2010.02828*.

[133] J. Ni and J. Hu, "Path following control for autonomous formula racecar: Autonomous formula student competition," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun. 2017, pp. 1835–1840. [Online]. Available: https://doi.org/10.1109/ivs.2017.7995972

[134] J. Ni, J. Hu, and C. Xiang, "Robust path following control at driving/handling limits of an autonomous electric racecar," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 5518–5526, Jun. 2019. [Online]. Available: https://doi.org/10.1109/tvt.2019.2911862

[135] J. Niu, Y. Hu, B. Jin, Y. Han, and X. Li, "Two-stage safe reinforcement learning for high-speed autonomous racing," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2020, pp. 3934–3941. [Online]. Available: https://doi.org/10.1109/smc42975.2020.9283053

[136] F. Nobis, J. Betz, L. Hermansdorfer, and M. Lienkamp, "Autonomous racing: A comparison of SLAM algorithms for large scale outdoor environment," in *Proc. ACM 3rd Int. Conf. Virtual Augmented Real. Simulat.*, Feb. 2019, pp. 82–89. [Online]. Available: https://doi.org/10.1145/3332305.3332319

[137] G. Notomista, M. Wang, M. Schwager, and M. Egerstedt, "Enhancing game-theoretic autonomous car racing using control barrier functions," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 5393–5399. [Online]. Available: https://doi.org/10.1109/icra40945.2020.9196757

[138] T. Novi, A. Liniger, R. Capitani, and C. Annicchiarico, "Real-time control for at-limit handling driving on a predefined path," *Veh. Syst. Dyn.*, vol. 58, no. 7, pp. 1007–1036, Apr. 2019. [Online]. Available: https://doi.org/10.1080/00423114.2019.1605081

[139] M. O'Kelly *et al.*, "F1/10: An open-source autonomous cyber-physical platform," 2019, *arxiv.abs/1901.08567*.

[140] M. O'Kelly, H. Zheng, D. Karthik, and R. Mangharam, "F1TENTH: An open-source evaluation environment for continuous control and reinforcement learning," in *Proc. NeurIPS Competition Demonstration Track*, vol. 123, Dec. 2020, pp. 77–89. [Online]. Available: http://proceedings.mlr.press/v123/o-kelly20a.html

[141] M. O'Kelly, H. Zheng, A. Jain, J. Auckley, K. Luong, and R. Mangharam, "TUNERCAR: A Superoptimization toolchain for autonomous racing," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 5356–5362. [Online]. Available: https://doi.org/10.1109/icra40945.2020.9197080

[142] R. Oliveira, F. H. M. Rocha, L. Ott, V. Guizilini, F. Ramos, and V. Grassi, "Learning to race through coordinate descent Bayesian optimisation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 6431–6438. [Online]. Available: https://doi.org/10.1109/icra.2018.8460735

[143] E. Pagot, M. Piccinini, and F. Biral, "Real-time optimal control of an autonomous RC car with minimum-time maneuvers and a novel kineto-dynamical model," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 2390–2396. [Online]. Available: https://doi.org/10.1109/iros45743.2020.9340640

[144] P. R. Palafox, J. Betz, F. Nobis, K. Riedl, and M. Lienkamp, "SemanticDepth: Fusing semantic segmentation and monocular depth estimation for enabling autonomous driving in roads without lane lines," *Sensors*, vol. 19, no. 14, p. 3224, Jul. 2019. [Online]. Available: https://doi.org/10.3390/s19143224

[145] Y. Pan *et al.*, "Agile autonomous driving using end-to-end deep imitation learning," in *Robotics: Science and Systems Foundation*. Cambridge, MA, USA: MIT Press, Jun. 2018. [Online]. Available: https://doi.org/10.15607/rss.2018.xiv.056

[146] H. Park and J. C. Gerdes, "Optimal tire force allocation for trajectory tracking with an over-actuated vehicle," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun. 2015, pp. 1032–1037. [Online]. Available: https://doi.org/10.1109/ivs.2015.7225820

[147] H. Park and J. C. Gerdes, "Analysis of feasible tire force regions for optimal tire force allocation with limited actuation," *IEEE Intell. Transp. Syst. Mag.*, vol. 9, no. 3, pp. 75–87, Jul. 2017, doi: 10.1109/MITS.2017.2709879. [Online]. Available: https://doi.org/10.1109/mits.2017.2709879

[148] S. Pedone and A. Fagiolini, "Racecar longitudinal control in unknown and highly-varying driving conditions," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12521–12535, Nov. 2020. [Online]. Available: https://doi.org/10.1109/tvt.2020.3023059

[149] W. Z. Peng, Y. H. Ao, J. H. He, and P. F. Wang, "Vehicle Odometry with camera-Lidar-IMU information fusion and factor-graph optimization," *J. Intell. Robot. Syst.*, vol. 101, no. 4, Apr. 2021. [Online]. Available: https://doi.org/10.1007/s10846-021-01329-x

[150] D. Perez, Y. Saez, G. Recio, and P. Isasi, "Evolving a rule system controller for automatic driving in a car racing competition," in *Proc. IEEE Symp. Comput. Intell. Games*, Dec. 2008, pp. 336–342. [Online]. Available: https://doi.org/10.1109/cig.2008.5035659

[151] E. Perot, M. Jaritz, M. Toromanoff, and R. D. Charette, "End-to-end driving in a realistic racing game with deep reinforcement learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 474–475. [Online]. Available: https://doi.org/10.1109/cvprw.2017.64

[152] F. K. Pour, D. Theilliol, V. Puig, and G. Cembrano, "Health-aware control design based on remaining useful life estimation for autonomous racing vehicle," *ISA Trans.*, vol. 113, pp. 196–209, Jul. 2021. [Online]. Available: https://doi.org/10.1016/j.isatra.2020.03.032

[153] P. Puchtler and R. Peinl, *Evaluation of Deep Learning Accelerators for Object Detection at the Edge* (Lecture Notes in Computer Science). Cham, Switzerland: Springer Int., 2020, pp. 320–326. [Online]. Available: https://doi.org/10.1007/978-3-030-58285-2_29

[154] J. Quadflieg, M. Preuss, and G. Rudolph, "Driving faster than a human player," in *Applications of Evolutionary Computation*. Berlin, Germany: Springer, 2011, pp. 143–152. [Online]. Available: https://doi.org/10.1007/978-3-642-20525-5_15

[155] R. Reiter, M. Kirchengast, D. Watzenig, and M. Diehl, "Mixed-integer optimization-based planning for autonomous racing with obstacles and rewards," *IFAC-PapersOnLine*, vol. 54, no. 6, pp. 99–106, 2021. [Online]. Available: https://doi.org/10.1016/j.ifacol.2021.08.530

[156] A. Remonda, S. Krebs, E. Veas, G. Luzhnica, and R. Kern, "Formula RL: Deep reinforcement learning for autonomous racing using telemetry data," Unpublished, 2019. [Online]. Available: http://rgdoi.net/10.13140/RG.2.2.30678.09283

[157] A. Remonda, E. Veas, and G. Luzhnica, "Comparing driving behavior of humans and autonomous driving in a professional racing simulator," *PLoS ONE*, vol. 16, no. 2, Feb. 2021, Art. no. e0245320. [Online]. Available: https://doi.org/10.1371/journal.pone.0245320

[158] T. Renzler, M. Stolz, M. Schratter, and D. Watzenig, "Increased accuracy for fast moving LiDARS: Correction of distorted point clouds," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf. (I2MTC)*, May 2020, pp. 1–6. [Online]. Available: https://doi.org/10.1109/i2mtc43012.2020.9128372

[159] U. Rosolia, A. Carvalho, and F. Borrelli, "Autonomous racing using learning model predictive control," in *Proc. Amer. Control Conf. (ACC)*, May 2017, pp. 5115–5120. [Online]. Available: https://doi.org/10.23919/acc.2017.7963748

[160] U. Rosolia and F. Borrelli, "Learning model predictive control for iterative tasks: A computationally efficient approach for linear system," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 3142–3147, Jul. 2017. [Online]. Available: https://doi.org/10.1016/j.ifacol.2017.08.324

[161] U. Rosolia, X. Zhang, and F. Borrelli, "Robust learning model predictive control for iterative tasks: Learning from experience," in *Proc. IEEE 56th Annu. Conf. Decis. Control (CDC)*, Dec. 2017, pp. 1157–1162. [Online]. Available: https://doi.org/10.1109/cdc.2017.8263812

[162] U. Rosolia and F. Borrelli, "Learning how to autonomously race a car: A predictive control approach," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 6, pp. 2713–2719, Nov. 2020. [Online]. Available: https://doi.org/10.1109/tcst.2019.2948135

[163] A. Rucco, G. Notarstefano, and J. Hauser, "An efficient minimum-time trajectory generation strategy for two-track car vehicles," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 4, pp. 1505–1519, Jul. 2015. [Online]. Available: https://doi.org/10.1109/tcst.2014.2377777

[164] M. Salem, A. M. Mora, J. J. Merelo, and P. García-Sánchez, "Driving in TORCS using modular fuzzy controllers," in *Applications of Evolutionary Computation*. Cham, Switzerland: Springer Int., 2017, pp. 361–376. [Online]. Available: https://doi.org/10.1007/978-3-319-55849-3_24

[165] M. Salem, A. M. Mora, and J. J. Merelo, "The evolutionary race: Improving the process of evaluating car controllers in racing simulators," in *Proc. IEEE Conf. Comput. Intell. Games (CIG)*, Aug. 2018, pp. 1–8. [Online]. Available: https://doi.org/10.1109/cig.2018.8490364

[166] M. Salem, A. M. Mora, and J. J. Merelo, "Beating uncertainty in racing bot evolution through enhanced exploration and pole position selection," in *Proc. IEEE Conf. Games (CoG)*, Aug. 2019, pp. 1–8. [Online]. Available: https://doi.org/10.1109/cig.2019.8847998

[167] J. P. S. Mejia, P. A. Theodosis, and J. C. Gerdes, "Using a path-fitting algorithm to Analyze the racing techniques of a skilled driver," in *Proc. Dyn. Syst. Control Conf.*, Oct. 2013, p. 10. [Online]. Available: https://doi.org/10.1115/dscc2013-4106

[168] F. Sauerbeck, L. Baierlein, J. Betz, and M. Lienkamp, "A combined LiDAR-camera localization for autonomous race cars [SAEIndexTestSampleV2 2022-01-18T10:06:44.605496Z]," *SAE Int. J. Connected Autom. Veh.*, vol. 5, no. 1, pp. 61–71, Jan. 2022. [Online]. Available: https://doi.org/10.4271/12-05-01-0006

[169] W. Schwarting *et al.*, "Deep latent competition: Learning to race using visual control policies in latent space," 2021, *arxiv.abs/2102.09812*.

[170] W. Schwarting, A. Pierson, S. Karaman, and D. Rus, "Stochastic dynamic games in belief space," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1–16, Dec. 2021. [Online]. Available: https://doi.org/10.1109/tro.2021.3075376

[171] A. Sinha, M. O'Kelly, H. Zheng, R. Mangharam, J. Duchi, and R. Tedrake, "FormulaZero: Distributionally robust online adaptation via offline population synthesis," in *Proc. 37th Int. Conf. Mach. Learn.*, vol. 119, Jul. 2020, pp. 8992–9004. [Online]. Available: http://proceedings.mlr.press/v119/sinha20a.html

[172] Y. Song, H. Lin, E. Kaufmann, P. Duerr, and D. Scaramuzza, "Autonomous overtaking in Gran Turismo sport using curriculum reinforcement learning," in *Proc. ICRA*, 2021, pp. 9403–9409.

[173] N. A. Spielberg, M. Brown, N. R. Kapania, J. C. Kegelman, and J. C. Gerdes, "Neural network vehicle models for high-performance automated driving," *Sci. Robot.*, vol. 4, no. 28, Mar. 2019, Art. no. eaaw1975. [Online]. Available: https://doi.org/10.1126/scirobotics.aaw1975

[174] S. S. Srinivasa *et al.*, "MuSHR: A low-cost, open-source robotic Racecar for education and research," 2019, *arxiv.abs/1908.08031*.

[175] S. Srinivasan, I. Sa, A. Zyner, V. Reijgwart, M. I. Valls, and R. Siegwart, "End-to-end velocity estimation for autonomous racing," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 6869–6875, Oct. 2020. [Online]. Available: https://doi.org/10.1109/lra.2020.3016929

[176] S. Srinivasan, S. N. Giles, and A. Liniger, "A holistic motion planning and control solution to challenge a professional racecar driver," 2021, *arxiv.abs/2103.00358*.

[177] T. Stahl, A. Wischnewski, J. Betz, and M. Lienkamp, "ROS-based localization of a race vehicle at high-speed using LIDAR," in *Proc. E3S Web Conf.*, vol. 95, 2019, Art. no. 04002. [Online]. Available: https://doi.org/10.1051/e3sconf/20199504002

[178] T. Stahl, A. Wischnewski, J. Betz, and M. Lienkamp, "Multilayer graph-based trajectory planning for race vehicles in dynamic scenarios," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 3149–3154. [Online]. Available: https://doi.org/10.1109/itsc.2019.8917032

[179] T. Stahl, M. Eicher, J. Betz, and F. Diermeyer, "Online verification concept for autonomous vehicles—Illustrative study for a trajectory planning module," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–7. [Online]. Available: https://doi.org/10.1109/itsc45102.2020.9294703

[180] T. Stahl and J. Betz, "An open-source scenario architect for autonomous vehicles," in *Proc. 15th Int. Conf. Ecol. Veh. Renew. Energies (EVER)*, Sep. 2020, pp. 1–8. [Online]. Available: https://doi.org/10.1109/ever48776.2020.9243029

[181] T. Stahl and F. Diermeyer, "Online verification enabling approval of driving functions—Implementation for a planner of an autonomous race vehicle," *IEEE Open J. Intell. Transp. Syst.*, vol. 2, pp. 97–110, 2021. [Online]. Available: https://doi.org/10.1109/ojits.2021.3078121

[182] K. Strobel, S. Zhu, R. Chang, and S. Koppula, "Accurate, low-latency visual perception for autonomous racing: Challenges, mechanisms, and practical solutions," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 1969–1975. [Online]. Available: https://doi.org/10.1109/iros45743.2020.9341683

[183] J. K. Subosits and J. C. Gerdes, "From the racetrack to the road: Real-time trajectory replanning for autonomous driving," *IEEE Trans. Intell. Veh.*, vol. 4, no. 2, pp. 309–320, Jun. 2019. [Online]. Available: https://doi.org/10.1109/tiv.2019.2904390

[184] J. Subosits and J. C. Gerdes, "Impacts of model fidelity on trajectory optimization for autonomous vehicles in extreme maneuvers," *IEEE Trans. Intell. Veh.*, vol. 6, no. 3, pp. 546–558, Sep. 2021. [Online]. Available: https://doi.org/10.1109/tiv.2021.3051325

[185] V. Sukhil and M. Behl, "Adaptive lookahead pure-pursuit for autonomous racing," 2021, *arxiv.abs/2111.08873*.

[186] K. L. Talvala, K. Kritayakirana, and J. C. Gerdes, "Pushing the limits: From lanekeeping to autonomous racing," *Annu. Rev. Control*, vol. 35, no. 1, pp. 137–148, Apr. 2011. [Online]. Available: https://doi.org/10.1016/j.arcontrol.2011.03.009

[187] A. Tătulea-Codrean, T. Mariani, and S. Engell, "Design and simulation of a machine-learning and model predictive control approach to autonomous race driving for the F1/10 platform," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6031–6036, 2020. [Online]. Available: https://doi.org/10.1016/j.ifacol.2020.12.1669

[188] P. A. Theodosis and J. C. Gerdes, "Generating a racing line for an autonomous Racecar using professional driving techniques," in *Proc. ASME Dyn. Syst. Control Conf. Bath/ASME Symp. Fluid Power Motion Control*, vol. 2, Jan. 2011, pp. 853–860. [Online]. Available: https://doi.org/10.1115/dscc2011-6097

[189] P. A. Theodosis and J. C. Gerdes, "Nonlinear optimization of a racing line for an autonomous racecar using professional driving techniques," in *Proc. ASME*, 2012, pp. 235–241. [Online]. Available: https://doi.org/10.1115/dscc2012-movic2012-8620

[190] H. Tian, J. Ni, and J. Hu, "Autonomous driving system design for formula student driverless Racecar," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun. 2018, pp. 1–6. [Online]. Available: https://doi.org/10.1109/ivs.2018.8500471

[191] H. Tian, J. Ni, Z. Li, and J. Hu, "Autonomous formula racecar: Overall system design and experimental validation," 2020, *arxiv.abs/2009.00385*.

[192] E. Tramacere, S. Luciani, S. Feraco, A. Bonfitto, and N. Amati, "Processor-in-the-loop architecture design and experimental validation for an autonomous racing vehicle," *Appl. Sci.*, vol. 11, no. 16, p. 7225, Aug. 2021. [Online]. Available: https://doi.org/10.3390/app11167225

[193] C. Vallon and F. Borrelli, "Task decomposition for iterative learning model predictive control," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2020, pp. 2024–2029. [Online]. Available: https://doi.org/10.23919/acc45564.2020.9147625

[194] M. I. Valls *et al.*, "Design of an autonomous racecar: Perception, state estimation and system integration," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2048–2055. [Online]. Available: https://doi.org/10.1109/icra.2018.8462829

[195] B. van Niekerk, A. Damianou, and B. Rosman, "Online constrained model-based reinforcement learning," 2020, *arxiv.abs/2004.03499*.

[196] J. L. Vazquez, M. Bruhlmeier, A. Liniger, A. Rupenyan, and J. Lygeros, "Optimization-based hierarchical motion planning for autonomous racing," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 2397–2403. [Online]. Available: https://doi.org/10.1109/iros45743.2020.9341731

[197] R. Verschueren, M. Zanon, R. Quirynen, and M. Diehl, "Time-optimal race car driving using an online exact hessian based nonlinear MPC algorithm," in *Proc. Eur. Control Conf. (ECC)*, Jun. 2016, pp. 141–147. [Online]. Available: https://doi.org/10.1109/ecc.2016.7810277

[198] C. Voser, R. Y. Hindiyeh, and J. C. Gerdes, "Analysis and control of high sideslip manoeuvres," *Veh. Syst. Dyn.*, vol. 48, no. S1, pp. 317–336, Dec. 2010. [Online]. Available: https://doi.org/10.1080/00423111003746140

[199] E. Wachter, A. Schmeitz, F. Bruzelius, and M. Alirezaei, *Path Control in Limits of Vehicle Handling: A Sensitivity Analysis* (Lecture Notes in Mechanical Engineering). Cham, Switzerland: Springer Int., 2020, pp. 1089–1095. [Online]. Available: https://doi.org/10.1007/978-3-030-38077-9_126

[200] N. Wagener, C.-A. Cheng, J. Sacks, and B. Boots, "An online learning approach to model predictive control," 2019, *arxiv.abs/1902.08967*.

[201] M. Wang, Z. Wang, J. Talbot, J. C. Gerdes, and M. Schwager, "Game theoretic planning for self-driving cars in competitive scenarios," in *Proc. Robot. Sci. Syst. XV*, 2019, pp. 1–9. [Online]. Available: https://doi.org/10.15607/RSS.2019.XV.048

[202] Z. Wang, R. Spica, and M. Schwager, "Game theoretic motion planning for multi-robot racing," in *Proc. Distrib. Auton. Robot. Syst.*, vol. 9, 2020, pp. 225–238. [Online]. Available: http://link.springer.com/10.1007/978-3-030-05816-6_16

[203] M. Wang, Z. Wang, J. Talbot, J. C. Gerdes, and M. Schwager, "Game-theoretic planning for self-driving cars in multivehicle competitive scenarios," *IEEE Trans. Robot.*, vol. 37, no. 4, pp. 1313–1325, Aug. 2021, [Online]. Available: https://doi.org/10.1109/tro.2020.3047521

[204] T. Weiss and M. Behl, "DeepRacing: Parameterized trajectories for autonomous racing," 2020, *arxiv.abs/2005.05178*.

[205] T. Weiss and M. Behl, "DeepRacing: A framework for autonomous racing," in *Proc. Design Autom. Test Europe Conf. Exhibit. (DATE)*, Mar. 2020, pp. 1163–1168. [Online]. Available: https://doi.org/10.23919/date48585.2020.9116486

[206] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 1433–1440. [Online]. Available: https://doi.org/10.1109/icra.2016.7487277

[207] G. Williams, B. Goldfain, P. Drews, J. M. Rehg, and E. A. Theodorou, "Autonomous racing with autorally vehicles and differential games," 2017, *arxiv.abs/1707.04540*.

[208] G. Williams *et al.*, "Information theoretic MPC for model-based reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 1714–1721. [Online]. Available: https://doi.org/10.1109/icra.2017.7989202

[209] G. Williams, B. Goldfain, P. Drews, J. M. Rehg, and E. A. Theodorou, "Best response model predictive control for agile interactions between autonomous ground vehicles," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2403–2410. [Online]. Available: https://doi.org/10.1109/icra.2018.8462831

[210] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Information-theoretic model predictive control: Theory and applications to autonomous driving," *IEEE Trans. Robot.*, vol. 34, no. 6, pp. 1603–1622, Dec. 2018. [Online]. Available: https://doi.org/10.1109/tro.2018.2865891

[211] G. Williams, B. Goldfain, P. Drews, K. Saigol, J. Rehg, and E. Theodorou, "Robust sampling based model predictive control with sparse objective information," in *Proc. Robot. Sci. Syst. XIV*, Jun. 2018, pp. 1–9. [Online]. Available: https://doi.org/10.15607/rss.2018.xiv.042

[212] G. Williams, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Locally weighted regression pseudo-rehearsal for online learning of vehicle dynamics," 2019, *arxiv.abs/1905.05162*.

[213] A. Wischnewski, T. Stahl, J. Betz, and B. Lohmann, "Vehicle dynamics state estimation and Localization for high performance race cars," *IFAC-PapersOnLine*, vol. 52, no. 8, pp. 154–161, 2019. [Online]. Available: https://doi.org/10.1016/j.ifacol.2019.08.064

[214] A. Wischnewski, J. Betz, and B. Lohmann, "A model-free algorithm to safely approach the handling limit of an autonomous racecar," in *Proc. IEEE Int. Conf. Connected Veh. Expo (ICCVE)*, Nov. 2019, pp. 1–6. [Online]. Available: https://doi.org/10.1109/iccve45908.2019.8965218

[215] A. Wischnewski, J. Betz, and B. Lohmann, "Real-time learning of non-Gaussian uncertainty models for autonomous racing," in *Proc. 59th IEEE Conf. Decis. Control (CDC)*, Dec. 2020, pp. 609–615. [Online]. Available: https://doi.org/10.1109/cdc42340.2020.9304230

[216] A. Wischnewski, M. Euler, S. Gümüs, and B. Lohmann, "Tube model predictive control for an autonomous race car," *Veh. Syst. Dyn.*, pp. 1–23, Jun. 2021. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/00423114.2021.1943461

[217] A. Wischnewski *et al.*, "Indy autonomous challenge—Autonomous race cars at the handling limits," 2022, *arxiv.abs/2202.03807*.

[218] C. You and P. Tsiotras, "Vehicle modeling and parameter estimation using adaptive limited memory joint-state UKF," in *Proc. Amer. Control Conf. (ACC)*, May 2017, pp. 322–327. [Online]. Available: https://doi.org/10.23919/acc.2017.7962973

[219] C. You and P. Tsiotras, "Real-time trail-braking maneuver generation for off-road vehicle racing," in *Proc. Annu. Amer. Control Conf. (ACC)*, Jun. 2018, pp. 4751–4756. [Online]. Available: https://doi.org/10.23919/acc.2018.8431620

[220] C. You and P. Tsiotras, "High-speed cornering for autonomous off-road rally racing," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 2, pp. 485–501, Mar. 2021. [Online]. Available: https://doi.org/10.1109/tcst.2019.2950354

[221] D. Zadok, T. Hirshberg, A. Biran, K. Radinsky, and A. Kapoor, "Explorations and lessons learned in building an autonomous formula SAE car from simulations," in *Proc. 9th Int. Conf. Simulat. Model. Methodol. Technol. Appl. (SCITEPRESS)*, 2019, pp. 414–421. [Online]. Available: https://doi.org/10.5220/0008120604140421

[222] M. Zeillinger, R. Hauk, M. Bader, and A. Hofmann. "Design of an autonomous race car for the formula student driverless (FSD)." 2017. [Online]. Available: https://openlib.tugraz.at/download.php?id=5aaa45931188a&location=medra

[223] V. Zhang, S. Thornton, and C. Gerdes, "Tire modeling to enable model predictive control of automated vehicles from standstill to the limits of handling," in *Proc. AVEC 14th Int. Symp. Adv. Veh. Control*, 2018, pp. 1–9. [Online]. Available: https://ddl.stanford.edu/sites/g/files/sbiybj9456/f/Zhang_2018_avec.pdf

[224] J. Zubaca, M. Stolz, and D. Watzenig, "Extended H_∞ filter adaptation based on innovation sequence for advanced ego-vehicle motion estimation," in *Proc. IEEE 3rd Connected Autom. Veh. Symp. (CAVS)*, Nov. 2020, pp. 1–4. [Online]. Available: https://doi.org/10.1109/cavs51000.2020.9334568

[225] J. Zubaca, M. Stolz, and D. Watzenig, "Smooth reference line generation for a race track with gates based on defined borders," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Oct. 2020, pp. 604–609. [Online]. Available: https://doi.org/10.1109/iv47402.2020.9304722

[226] I. Zubov, I. Afanasyev, A. Gabdullin, R. Mustafin, and I. Shimchik, "Autonomous drifting control in 3D car racing simulator," in *Proc. Int. Conf. Intell. Syst. (IS)*, Sep. 2018, pp. 235–241. [Online]. Available: https://doi.org/10.1109/is.2018.8710588

[227] C. Jung, S. Lee, H. Seong, A. Finazzi, and D. H. Shim, "Game-theoretic model predictive control with data-driven identification of vehicle model for head-to-head autonomous racing," in *Proc. Int. Conf. Robot. Autom. (ICRA) Workshop Opportunities Challenges Auton. Racing*, 2021, pp. 1–9. [Online]. Available: https://linklab-uva.github.io/icra-autonomous-racing/contributed_papers/paper1.pdf

[228] N. Li, E. Goubault, L. Pautet, and S. Putot, "Autonomous racecar control in head-to-head competition using mixed-integer quadratic programming," in *Proc. Int. Conf. Robot. Autom. (ICRA) Workshop Opportunities Challenges Auton. Racing*, 2021, pp. 1–9. [Online]. Available: https://linklab-uva.github.io/icra-autonomous-racing/contributed_papers/paper2.pdf

[229] M. Schratter *et al.*, "LiDAR-based mapping and Localization for autonomous racing," in *Proc. Int. Conf. Robot. Autom. (ICRA) Workshop Opportunities Challenges Auton. Racing*, 2021, pp. 1–6. [Online]. Available: https://linklab-uva.github.io/icra-autonomous-racing/contributed_papers/paper4.pdf

[230] T. Weiss, J. Chrosniak, and M. Behl, "Towards multi-agent autonomous racing with the DeepRacing framework," in *Proc. Int. Conf. Robot. Autom. (ICRA) Workshop Opportunities Challenges Auton. Racing*, 2021, pp. 1–5. [Online]. Available: https://linklab-uva.github.io/icra-autonomous-racing/contributed_papers/paper5.pdf

[231] M. Schmid, Q. Zhu, A. Boncimino, R. Prucka, and C. Paredis, "Control informed design of the IAC autonomous Racecar for operation at the dynamic envelope," in *Proc. Int. Conf. Robot. Autom. (ICRA) Workshop Opportunities Challenges Auton. Racing*, 2021, pp. 1–7. [Online]. Available: https://linklab-uva.github.io/icra-autonomous-racing/contributed_papers/paper6.pdf

[232] X. Xiao, J. Biswas, and P. Stone, "Learning inverse Kinodynamics for accurate high-speed off-road navigation on unstructured terrain," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 6054–6060, Jul. 2021. [Online]. Available: https://doi.org/10.1109/lra.2021.3090023

[233] D. Kalaria *et al.*, "Local NMPC on global optimised path for autonomous racing," in *Proc. Int. Conf. Robot. Autom. (ICRA) Workshop Opportunities Challenges Auton. Racing*, 2021, pp. 1–6. [Online]. Available: https://linklab-uva.github.io/icra-autonomous-racing/contributed_papers/paper8.pdf

[234] J. Bhargav, J. Betz, H. Zheng, and R. Mangharam, "Track based offline policy learning for overtaking maneuvers with autonomous racecars," in *Proc. Int. Conf. Robot. Autom. (ICRA) Workshop Opportunities Challenges Auton. Racing*, 2021, pp. 1–6. [Online]. Available: https://linklab-uva.github.io/icra-autonomous-racing/contributed_papers/paper9.pdf

[235] S. Wadeka *et al.*, "Towards end-to-end deep learning for autonomous racing: On data collection and a unified architecture for steering and throttle prediction," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA) Workshop Opportunities Challenges Auton. Racing*, 2021, pp. 1–8. [Online]. Available: https://linklab-uva.github.io/icra-autonomous-racing/contributed_papers/paper10.pdf

[236] T. Brüdigam, A. Capone, S. Hirche, D. Wollherr, and M. Leibold, "Gaussian process-based stochastic model predictive control for overtaking in autonomous racing," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA) Workshop Opportunities Challenges Auton. Racing*, 2021, pp. 1–7. [Online]. Available: https://linklab-uva.github.io/icra-autonomous-racing/contributed_papers/paper11.pdf

[237] R. Wang, Y. Han, and U. Vaidya, "Deep Koopman data-driven control framework for Autonomous racing," in *Proc. Int. Conf. Robot. Autom. (ICRA) Workshop Opportunities Challenges Auton. Racing*, 2021, pp. 1–6. [Online]. Available: https://linklab-uva.github.io/icra-autonomous-racing/contributed_papers/paper12.pdf

[238] M. Jiang *et al.*, "Continuous integration and testing for autonomous racing software: An experience report from GRAIC," in *Proc. Int. Conf. Robot. Autom. (ICRA) Workshop Opportunities Challenges Auton. Racing*, 2021, pp. 1–5. [Online]. Available: https://linklab-uva.github.io/icra-autonomous-racing/contributed_papers/paper13.pdf

[239] S. D. Pendleton *et al.*, "Perception, planning, control, and coordination for autonomous vehicles," *Machines*, vol. 5, no. 1, p. 6, 2017. [Online]. Available: https://www.mdpi.com/2075-1702/5/1/6

[240] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2016, pp. 1271–1278.

[241] E. Espié *et al.*, "TORCS, the open racing car simulator," 2005. Accessed: Jun. 16, 2022.

[242] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. 1st Annu. Conf. Robot Learn.*, 2017, pp. 1–16.

[243] G. Brockman *et al.*, "Openai gym," 2016, *arXiv:1606.01540*.

[244] A. Domahidi and J. Jerez. (Embotech AG, Zürich, Switzerland). *FORCES Professional*. (2019). [Online]. Available: https://embotech.com/FORCES-Pro

[245] D. Ha and J. Schmidhuber, "Recurrent world models facilitate policy evolution," in *Advances in Neural Information Processing Systems*, vol. 31, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Red Hook, NY, USA: Curran, Curran Assoc., Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper/2018/file/2de5d16682c3c35007e4e92982f1a2ba-Paper.pdf

JOHANNES BETZ (Member, IEEE) received the B.Eng. and M.Sc. degrees in the field of Automotive Engineering 2011 and 2012, respectively, and the Ph.D. degree from the Technical University of Munich (TUM). He was a Postdoctoral Researcher with the Institute of Automotive Technology, TUM, where he founded the TUM Autonomous Motorsport Team. He is currently a Postdoctoral Researcher with the University of Pennsylvania, where he is working with the xLab for Safe Autonomous Systems. His research is focusing on a holistic software development for autonomous systems with extreme motions at the dynamic limits in extreme and unknown environments. By using modern algorithms from the field of artificial intelligence he is trying to develop new and advanced methods and intelligent algorithms. Based on his additional studies in philosophy he extends current path and behavior planners for autonomous systems with ethical theories

HONGRUI ZHENG (Student Member, IEEE) received the B.S. degrees in mechanical engineering and computer science from the Georgia Institute of Technology and the M.S. degree in robotics from the University of Pennsylvania, where he is currently pursuing the Ph.D. degree. He is working with the xLab for Safe Autonomous Systems, University of Pennsylvania. His research focuses on building the tools and theoretical foundations necessary to scale design, testing, and optimization of safe-autonomous systems.

ALEXANDER LINIGER received the B.Sc. and M.Sc. degrees in mechanical engineering from the Department of Mechanical and Process Engineering, ETH Zurich, Switzerland, in 2010 and 2013, respectively, and the Ph.D. degree from the Automatic Control Laboratory, ETH Zurich in May 2018. He is currently a Postdoctoral Researcher with the Computer Vision Lab, ETH Zurich, where he is part of Luc van Gool's Group working in the Toyta TRACE Project. During his Ph.D. his main research interests include model predictive control, viability theory as well as game theory and their application to autonomous driving and racing. He is currently investigating how control theory and computer vision can be combined to achieve end-to-end learning approaches with formal guarantees.

UGO ROSOLIA is a Postdoctoral Scholar with Caltech, working with Prof. A. Ames and Prof. Y. Yue. His current research focuses on high-level planning in partially observable environments and on designing control algorithms that allow autonomous systems to perform highly dynamical maneuvers while guaranteeing safety. During his Ph.D. with the University of California Berkeley, he worked with Prof. F. Borrelli with MPC Lab. He developed the Learning Model Predictive Control strategy, which is a model-based policy iteration strategy. This strategy was used to teach an autonomous vehicle how to race.

PHILLIP KARLE received the B.Sc. and M.Sc. degrees from the Technical University of Munich (TUM), Munich, Germany, in 2017 and 2019, respectively, where he is currently pursuing the Ph.D. degree in mechanical engineering with the Institute of Automotive Technology. His main research interests include data-mining, scenario understanding, motion prediction, and related applications for autonomous driving with the focus on unstructured environments.

MADHUR BEHL received the M.S. and Ph.D. degrees in electrical and systems engineering from the University of Pennsylvania in 2012 and 2015, respectively. He is an Assistant Professor with the Department of Computer Science, and Systems and Information Engineering, and a Member of the Cyber–Physical Systems Link Lab, University of Virginia. He conducts research at the confluence of machine learning, predictive control, and artificial intelligence with applications in cyber–physical systems, autonomous systems, robotics, and smart cities.

VENKAT KROVI (Senior Member, IEEE) received the Ph.D. degree in mechanical engineering and applied mechanics from the University of Pennsylvania in 1998. He is the Michelin Endowed Chair of Vehicle Automation with the Department of Automotive Engineering and Mechanical Engineering, Clemson University. The underlying theme of his research activities has been to take advantage of the "power of the many" (both autonomous-agents and humans) to extend the reach of human users in the dull, dirty, and dangerous environments. These efforts at analyzing and realizing the potential of distributed-autonomy and human–robot synergy has unlocked new opportunities in wide ranging applications in plant automation, consumer electronics, automobile, defense, and healthcare/surgical simulation arenas.

RAHUL MANGHARAM (Member, IEEE) received the B.S. and M.S. degrees and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University. He is an Associate Professor with the Department of Electrical and Systems Engineering, University of Pennsylvania. He is a Founding Member of the PRECISE Center and directs the xLab for Safe Autonomous Systems Lab, Penn. His research is at the intersection of formal methods, machine learning and control for medical devices, energy efficient buildings, and autonomous systems.