Indy Autonomous Challenge

Team Technical University of Munich

January 2020

1 Introduction

The cars of the future should not only be cleaner, but also safer and more efficient. Automated and connected driving will play a key role in this field. The potential of autonomous technology is enormous - for society, for safety and for business development worldwide.

There have been successes in research and development. Self-driving cars from well-known car manufacturers such as Daimler, BMW and VW are already on the road for tests. Toyota has just announced that it wants to build an experimental city of the future in Japan to increasingly test technologies such as autonomous driving in real environments. Experts place companies in the USA at the forefront for autonomous driving know-how- above all the US company Waymo. Waymo has been driving robot cars as taxis for a while, now they started to drive without a security driver behind the wheel. Customers can order their ride via app and are picked up wherever they are.

Unfortunately, there are still many unsolved technical questions: One of the biggest challenges for robot cars is to react correctly in all situations. To do this, the vehicle has to recognize their environment precisely and process the data quickly. The requirements differ depending on whether the car is on the country road, the highway or in the city. Another problem with autonomous driving is the weather and the often uncoordinated road traffic. Autonomous vehicles must function perfectly not only in sunny weather, but also in heavy rain, fog and snow. The same applies to asphalt roads with and without markings. In addition, the current development is based on constraints imposed by the road.

Our goal is to take part in the Indy Autonomous Challenge, because we think this special environment offers us the environment we need for the next level in our development of an autonomous driving software. The Indy Autonomous Challenge offers an autonomous level 5 vehicle which, due to its high speed, must process high-performance, robust and secure algorithms at high clock rates in real time. The absence of a driver also eliminates a limiting factor (too high G-forces) in the vehicle and thus creates the possibility to fully test the technical potential of all autonomous driving algorithms on safe and reproducible terrain (race track) in various scenarios. The racing vehicle therefore represents an extreme case, all the knowledge gained from the driving tests in this project can thus be transferred 1:1 to the development of conventional passenger cars.

2 The Team

The team members have previously participated in a competitive autonomous racing series called Roborace [1]. Their respective fields of expertise are presented in this section. The team members will continue their work in the presented fields during the Indy Autonomous Challenge.

- Prof. Dr. Markus Lienkamp holds the Institute of Automotive Technology (FTM) [2] at the Technical University of Munich since 2009. He graduated with a doctoral degree from the Technical University of Darmstadt. Following this, he continued his research in a variety of positions in the Volkswagen AG. Before his return to academia, he lead a team of 150 engineers and 20 PhDs in the field of Electronics and Vehicle Research. After his appointment to the Chair of Automotive Technology,

[1] https://roborace.com/
[2] http://www.ftm.mw.tum.de/en/home/

he initiated several large scale research projects in the areas of vehicle design, electric mobility, teleoperation and autonomous driving. In the research projects *Visio.M* and *MUTE* a compact electric vehicle concept has been developed. In 2017, professor Lienkamp initiated the cooperation with the Roborace racing series. In this environment, a team of the Technical University participates in what is the first racing series for autonomous vehicles. Professor Lienkamp established a team of a postdoctoral researcher and six Phd students for this task. He is looking for new challenges in autonomous racing in the context of the Indy Autonomous Challenge.

- Dr. Johannes Betz is a postdoctoral researcher at the Chair of Automotive Technology. He graduated with a doctoral degree from the Technical University of Munich about the topic of fleet disposition for electric vehicles. Since then, he continues his research in the field of autonomous driving. He is the lead of the TUM team that participates in the Roborace series. During the cooperation with Roborace, he published the approach of a holistic software design for autonomous racing vehicles [3, 2]: He has a strong knowledge in the planning, coordination, development and deployment of software for a competition format [4].

- Alexander Heilmeier focuses on the research topics of racing strategy and trajectory planning. He developed an open-source tool for the calculation of a racing trajectory for circuit racing [10]. This tool can be integrated into a lap-time simulation, to simulate the lap times of competitive racing drivers. He showed the applicability of this approach to simulate racing performance for Formula E and Formula 1 vehicles [8]. On top of the lap-time simulation, he builds a race simulation for motor sport [9]. These tools can be applied to the competitive racing during the Indy Autonomous Challenge.

- Alexander Wischnewski focuses on the control of vehicles at the handling limits. For the Roborace competition, he designed a controller that stabilized the vehicle running at speeds of up to 210 km/h [10] and the required sensor fusion module [20]. Currently he is developing a control strategy which automatically explores the handling limits of vehicles through a data-based learning strategy [19]. This strategy guarantees that the vehicle stays in a safe operating area, so that the vehicle can maximize its racing performance, without risking a crash.

- Tim Stahl is conducting research in cooperation with the TÜV Süd AG. His research deals with a plethora of topics for the safe operation of autonomous vehicles. For the Roborace team, he incorporated LIDAR features into the localization module to localize a vehicle safely on a race track [18]. Furthermore, he integrated a local trajectory planer which enabled the first autonomous overtake in a racing series [17]. In his cooperation with the TÜV Süd, he uses this trajectory planer to develop and validate safeguarding approaches for the trajectory planning for autonomous vehicles.

- Thomas Herrmann researches in the field of energy optimal trajectory planning for autonomous vehicles. To facilitate his research, he built up a driving simulation for race track environments with MATLAB/Simulink and the Unreal Engine. Here, he modeled the driving physics of the Roborace vehicles and the environment of the race tracks. In his current work, he is using optimization techniques to develop a trajectory planer which considers energy and power limitations while minimizing the lap-time of racing cars [12].

- Felix Nobis researches in the field of perception for autonomous driving. He developed the mapping solution for the Roborace project [14]. In his current work, he is focusing on a 3D object detection which is based on camera input data only. He is conducting research in cooperation with Continental Engineering Services, where he is developing deep learning techniques for the processing and fusion of radar data for objecte detection[15].

- Leonhard Hermansdorfer focuses on the estimation of the friction limits of vehicles. For the Roborace competition, he was responsible for map processing features to enable the global trajectory planning. Concurrently, he developed a detailed vehicle model of the research vehicle to estimate the handling limits along the racing trajectory. He extracts the maximum friction potential of the respective driving areas into a high resolution friction map [11]. Currently, he focuses to estimate the local friction potential without the need of such a fine-grained map.

Figure 1: The TUM Roborace Team

3 History of Automation in the TUM Team

First steps towards automated driving at the Institute of Automotive Technology at TUM were taken in the year 2011 when research on tele-operated driving started. The idea of tele-operated driving is based on the insight that automated driving is possible, but within a foreseeable horizon of time only in certain and less complex scenarios such as highway driving or stop-and-go traffic. Tele-operated driving fills the missing automation in more complex scenarios by a human ("operator") who remotely controls the vehicle from a kind of driving simulator [6, 7].

In the mid 2010's the topic of automated driving got a big boost mainly because of the progress in the perception area. The usage of graphic cards in combination with artificial neural networks accelerated the development of perception algorithms for complex scenarios massively. To keep track with this development, the FTM started another research project: autonomous motorsports. The project is conducted in cooperation with Roborace, a British company that builds the first autonomous and fully electric race car. Hereby, the teams only participate in the race series with their software. The cars themselves are provided by Roborace and are identical for all teams. The participating teams come from all over Europe: University of Pisa (Italiy), University of Graz (Austria), Innopolis University (Russia) and Arrival (a British company linked to Roborace). Our intention for the project was (and still is) the development of an entire software stack for an autonomous car in an extreme environment. The software therefore has to cover three major tasks: perception, planning and control. Hereby, the Roborace platform allows us to combine two important aspects: cutting edge research and real world application. For example, due to the high speeds of the car and the driving at the limits of the vehicle dynamics we have to use highly reliable and at the same time real-time capable algorithms for our software. This is highly relevant also for road vehicles [4].

Starting with five PhD students in the beginning of 2018, we developed a first working version of the required software stack until March 2018, were we started with the first tests in Upper Heyford, UK. On the perception side, we used SLAM to create a map of the race track while a real driver drove the car during a training session [18]. This map was then post-processed and handed into the trajectory planning module. It calculated our race line based on a quadratic optimization minimizing the summed discretized curvature along the race track [10]. For the control of the car we trusted a PD-controller using curvature and longitudinal acceleration as feed-forward input [10]. The whole concept proved to be fast and reliable and it was easy to adapt to new environments and race tracks. In May 2018, we then presented our software in the public for the fist time during the Formula E event in Berlin. Hereby, we have beaten the University of Pisa by a few seconds and almost reached the lap time of a human driver.

After the Berlin event, two additional PhD students joined the team. For the 2019 season ("Season Alpha"), we concentrated on improving the performance and energy efficiency. Therefore, research effort is put into several sub-projects:

- Multi-modal sensor fusion and 3D object detection [14, 15]
- Assessment and prediction of the tire-road friction potential [11]

- Safety assessment for the trajectory planning of an autonomous car [17]
- Optimization of the energy management for an autonomous race car [12]
- Race strategy optimization in circuit motorsports [9, 8]
- Self-learning controller for an autonomous vehicle at the limits of handling [20]

The first results of this research were successfully implemented and tested on the “Devbot 2.0” (see Figure 2) during several events of the 2019 season. The team experienced highs (faster-than-human lap time and first successful overtake in Modena, Italy and victory in Le-Croix, France) and lows (crash of the car in Modena, Italy [1]). Most important: we were able to gather lots of insights and data for our research.

Figure 2: The Roborace race car.

Apart from the pure software stack development, we have built a hardware-in-the-loop simulator at our institute which provides us the possibility to simulate our entire software stack on the target hardware with a setup identical to the real race car. The students benefited from the possibility to develop and program autonomous RC cars in their thesis’ and from a new lecture “Artificial Intelligence in Automotive Technology”. However, not only TUM students benefit from the research. Everybody can download and use huge parts of our software stack from Github[3]. The race line optimization is, for example, used by lots of teams in the “Formula Student Driverless” competition.

4 Technical Information

4.1 Autonomous Driving Software

The TUM Autonomous Racing Software is roughly structured into the well-known parts perception, planning and control (see Fig. 3). This provides a set of fairly stable interfaces while at the same time generating reasonable sub-team sizes for the development process.

The perception modules generates a race-track representation and an object list as an input to the planning module. The race-track is described in a reference coordinate system oriented along the track, together with its bounds. A key aspect during the development has been to develop technology agnostic interfaces, such that the planning module does not have to care about object fusion or any details of the mapping process.

The planning part is splitted into a trajectory planning and a behavior planning submodule. The former generates several options for the behavior planning using a point-mass dynamics model, such as driving the time-optimal raceline or to overtake the vehicle in front. The behavior planning selects the most suitable option and aligns it with long-term goals such as risk and race-strategy. It is furthermore allowed to set several hyperparameters of the trajectory planner, such as top-speed limit or acceleration

[3]https://github.com/TUMFTM

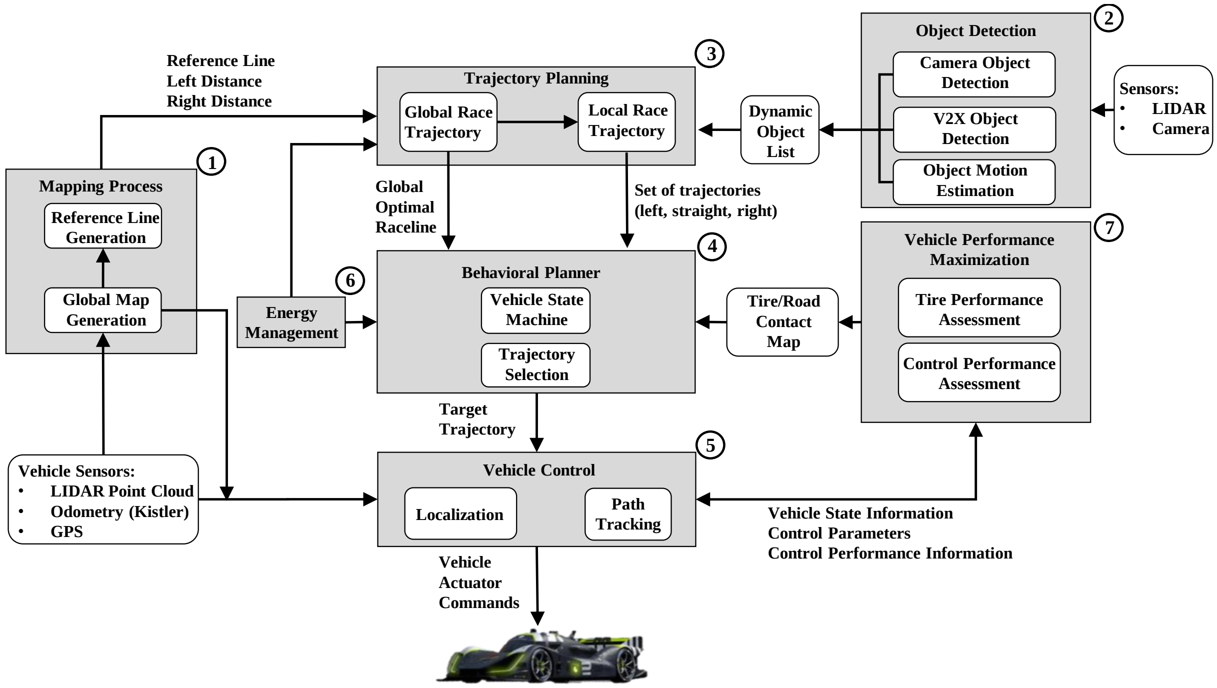


Figure 3: Architecture of TUM Autonomous Racing Software

limits and processes information from the energy management and the vehicle performance maximization submodules. This concept was introduced to allow for a separation of time-scales, such that the planning does not have to incorporate all high-level objectives.

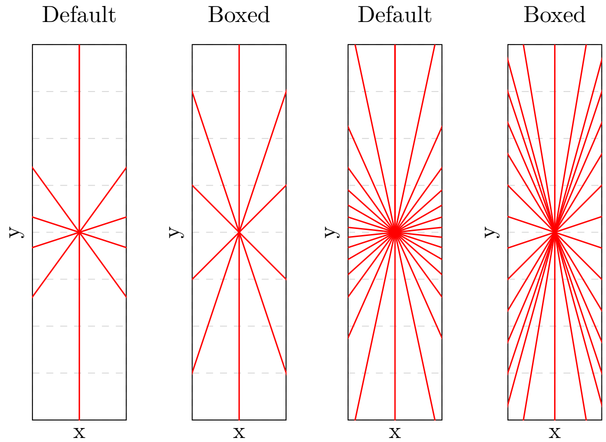
The controller receives the target trajectory and tracks it with a certain accuracy. This software module further manages the vehicle interface and therefore hides the vehicle complexity and its nonlinearites from the planning and perception modules. It further provides safety checks and handles the start up and shutdown procedures. The tracking capabilities of the controller re monitored via the control performance assessment module. This is augmented by a tire performance assessment algorithm, which allows to estimate the overall level of friction available at the track. Together they deliver an estimate for the achievable vehicle performance and allow to gradually decrease lap-time as the software learns from experience.

4.2 SLAM Knowledge

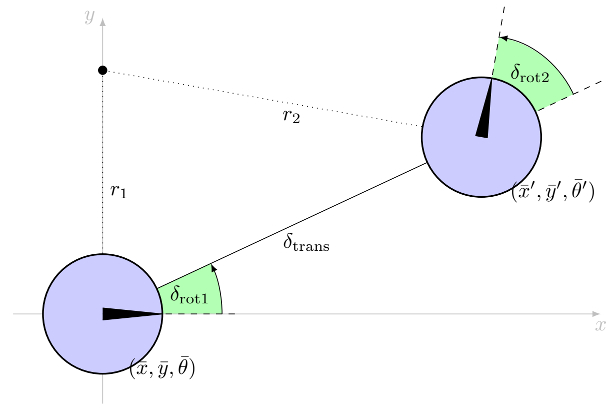
In order to drive fast and safe, an autonomous car needs to know accurately where it is located relative to its environment and other vehicles. In the related work, several techniques serve this purpose, each coupled to their individual advantages and disadvantages. In the scope of a race environment, extended and evaluated two approaches tackling pure localization as well as SLAM. The extension of a particle filter based approach for localization is elaborated in Section 4.2.1 and details on the application of a camera SLAM algorithm are given in Section 4.2.2.

4.2.1 AMCL in a Race Environment

The proposed method for high-speed LIDAR localization of a race vehicle is based on an adaptive Monte Carlo approach. The baseline approach uses a particle filter to estimate its position within an occupancy map. The maps were generated via gmapping or cartographer in separate mapping runs, but also offer the capability for SLAM execution. We improved the motion update and measurement step in the particle filter to support vehicles at high speeds. The modifications are outlined in the following, further details can be found in our paper [18].



(a) Comparison of default and introduced boxed approach for LIDAR beam extraction.



(b) Geometrical relations of the utilized odometry model consisting of two rotation and one translation step. The blue discs indicate the vehicles start and end pose.

Figure 4: Supporting visualization for the introduced modifications to the AMCL framework.

One of the most crucial issues when facing high speeds is the calculation time. Among the options allowing a reduction in computation burden, is to reduce the amount of LIDAR beams to be processed in each sensor update step. Commonly, only a representative subset of all the LIDAR measurements is evaluated. The default way is to sample a fixed amount of entities at an equal angular displacement from the full measurement set. Since a race vehicle is commonly driving in a sort of corridor with walls to the sides and free space ahead of the vehicle, this approach is not the most efficient. We introduced an approach to extract a race corridor optimized pattern of measurements (Figure 4a). The goal is to increase the information gain, while keeping the amount of extracted LIDAR beams constant. For this purpose, the angular displacement is calculated based on equal distances on the borderline of a surrounding rectangle with parameterizable aspect ratio.

One of the key goals of the motion update step (Figure 4b) is to translate all particles while at least covering or better overestimating the whole area of expected maximal error in the provided motion step. When moving a moderate speeds coupled with a proper parameter tuning, the publicly available algorithms perform well. However, when facing high velocities (i.e. larger motion steps δ_{trans}), more and more rotational noise is introduced). This spread in particles causes localization inaccuracies and a growing computational effort (due to the generation of additional particles due to the utilization of an adaptive filter). As the required lateral force rises quadratic with higher speeds, while following a fixed track radius, we reduce the introduced rotational noise according to the current speed.

The approach was implemented and evaluated on the DevBot up to speeds of 150km/h. We fused the localization estimate with other signal sources (e.g. GNSS) in an adjacent Kalman filter. Thereby, the lateral positioning on the track performed well with a mean absolute lateral error of 0.086m.

4.2.2 Camera SLAM

In the domain of vision based SLAM, we focused on the prominent ORB-SLAM2 [13] framework and investigated its performance when facing a race scenario. The algorithm generates keyframes for feature rich landmarks in the image flow and stores them in a map. Localization is based on a pose prediction coupled with a keyframe matching. Furthermore, the algorithm supports loop closure detection and correction. We successfully implemented and tested the approach on our RC-scale vehicle as well as on recorded data sets of the DevBot.

Within our tests, we concluded that vision based SLAM is only applicable when travelling at moderate speeds. In order to allow a vision based localization in a race environment, it is crucial to allow a moderate mapping lap before the actual race or load a map generated in previous runs. We extended the algorithm by the capability to import and export the visual map features in order to enable vision based localization for race events with high velocities right from the beginning of the race.

4.3 Testing

This section contains explanations of our software testing pipelines that help us to develop software for autonomous race cars [3]. In the past, these pipelines have been quite important within the Roborace project. Due to the limited testing time on track other means are required to test and validate our software functions such as

- the inter-component communication between the vehicle's control units.
- the behavior on static and dynamic obstacles.
- safety as well as emergency stops.
- the performance in case of sensor failures.

Additionally, we want to be able to tune the software modules as accurate as possible in our labs before going to the race track. By this, performance potentials regarding lap time or processor loads can be identified as early as possible.
The software modules "Vehicle Dynamics" and "Controller" are compiled and distributed regularly using GitLab's "CI/CD" functionality. By that, other software modules, e.g., the trajectory or behavior planner can be developed and tested conveniently on the developer's PCs. Still, the communication is realized via the same UDP-interfaces as in the real car in this Software-in-the-Loop approach.
To test the developed software on the DevBot's Electric Control Unit (ECU)s we set up a HIL-Simulator consisting of an Nvidia Drive PX2, a Speedgoat Mobile as well as a Performance real-time target machine and a GPU Server (Figure 5). The HIL-Simulator allows us to test our algorithms in a safe environment with conveninent debugging features. Additionally, we are able to generate dynamic sensor feedback.
The Nvidia Drive PX2 receives environment information and computes local trajectories. These are forwarded to the Speedgoat Mobile machine being responsible for the Sensor Fusion as well as for the transformation of the requested trajectories into steering angle and throttle commands. These signals are the input into the Vehicle and Environment Simulation. Here, the Speedgoat Performance machine replaces the real vehicle including an implementation of non-linear driving dynamics as well as low-level ECU models. The vehicle's position and motion state is visualized on the GPU Server rendering the scene and doing the sensor simulation.

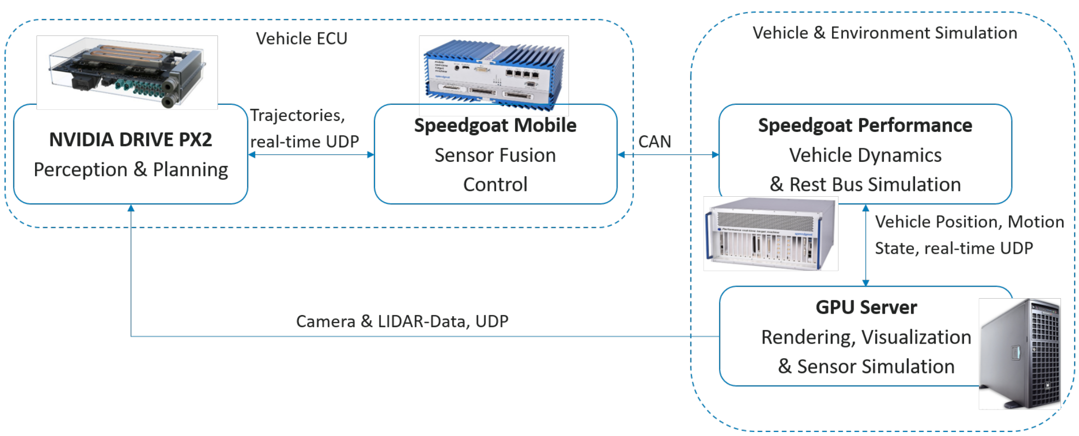


Figure 5: The TUM Hardware-in-the-Loop Simulator

4.4 Technical Insights and Specification

In this section, we outline the trajectory planning algorithms, running both off- and online, the algorithms which control the vehicle during racing and tools for optimizing the vehicle setup and the race strategy when competing against multiple opponents. All algorithms are leveraged by our institute's extensive background in vehicle dynamics simulation and testing [16]. Currently, there are several projects which focus on the given racing application. This includes global and local trajectory planning, adaptive vehicle dynamics control and race strategy and lap time simulation.

4.4.1 Trajectory Planning

Our trajectory planning is split into a global race trajectory planner [10],[5] and a local trajectory planner [17]. Global trajectory planning uses spatial knowledge about the racetrack, represented as a static map, to generate a time optimal trajectory. The algorithm models vehicle dynamics using a double-track vehicle model which is able to consider varying friction conditions on the racetrack [5]. The result is a global trajectory which represents the desired trajectory on a static track without opponent vehicles.

The general goal of local trajectory planning is to follow the pre-calculated global trajectory whenever possible [17]. However, the local trajectory planner must react to obstacles which are located on the racetrack, e.g. static obstacles and opponent vehicles. To always provide a feasible path in presence of moving obstacles, the local trajectory is re-planned frequently (about 10 Hz) and has to consider the vehicle dynamics limits at every time. Finally, the local trajectory should always come back to the global trajectory after it deviated, e.g. in order to pass an object.

The local trajectory planner consists of an offline and an online part [17]. Within the offline part, a state lattice covering the complete track is generated. Afterwards, splines connecting every state lattice node and the corresponding costs are calculated. High costs mean a low desirability of the vehicle to follow a specific path. During online execution, the offline calculated graph is used to create feasible spline combinations which consider static and dynamic obstacles. Since these spline combinations are associated with costs, a selection of the best action sets is possible. Finally, each chosen action set is post-processed by calculating a $C2$ continuous spline and generating a velocity profile. These trajectories are then handed to the behavior planer which makes the final decision on what trajectory to follow based on the current race situation.

4.4.2 Adaptive Vehicle Dynamics Control

In general, the control module is split into two parts, the trajectory-tracking and the low-level control part [2]. The trajectory-tracking part is a PD-like-Controller with gain scheduling by measuring lateral deviation and velocity heading error, the low-level velocity controller is a P-controller combined with feedback of a disturbance estimator [3]. Furthermore, we extended the low-level control part by integrating a learning feedforward controller which constantly updates the relation between curvature request, velocity and the resulting measured curvature [2]. As a result, a corrective steering angle is obtained enabling the controller to compensate for steering calibration errors and steady-state under- and oversteering.

To further enhance vehicle performance and to adapt to changing conditions, both of the track surface and the vehicle itself, we currently develop algorithms which enable online adaption. These algorithms allow the software to adapt to changing tire behavior and tire-road friction [11] and to position-depended control errors [2].

4.4.3 Race Strategy and Lap Time Simulation

To prepare for race events, a lap time simulation and a race strategy evaluation tool are used. The lap time simulation allows us to evaluate the effects of vehicle setup changes on lap time and energy consumption [8], [12]. It provides a precise reference for lap time on a given race track and has proofed as a important measure for what lap times we can target.

Whereas the lap time simulation considers only one single vehicle on the track, the race strategy simulation tool takes multiple opponent vehicles into account [9]. Its main goal is to provide an optimal race strategy under certain assumptions and for a set of boundary conditions. Next to pre-event optimization, the race strategy can be adjusted in real-time when providing information about the actual progress of the race.

5 Project Management

5.1 Overall

With the acknowledgement that we are part of the Indy Autonomous Challenge our intensified development begins. With the knowledge of the hardware architecture of the autonomous Indy Car we start to enhance our current software architecture to the new computing platform and interfaces of the vehicle. In addition we begin to model the vehicle dynamics of the autonomous Indy car as well as modeling the Indy racetrack in our simulator. Our special focus for the Indy Challenge is an enhancement of our trajectory planner where we put As a Postdoc, Johannes Betz will be mainly responsible for the organization (calls, discussion etc.) with the Indy organizers and other teams. The team members itself will be responsible for individual software parts where they have the most knowledge in (section 2). The software development will be splitted in SCRUM sprints of about 4 weeks with a final test in our HiL. Each team member will be supported by several graduate students we are searching right now. In addition we plan to hire 2-3 more PhD students which help to enhance the trajectory planer and vehicle prediction part.

5.2 Fundraising

The fundraising is splitted. Three of the team members are founded by basic funds of the university, two teammates are funded by third party funding's and two are funded by public funds for a research project. The funding's are enough to cover the monthly payment, publication costs, material costs as well as travel costs. Currently we are talking to different companies (e.g. Audi, BMW, AID,..) for additional support for our team in the Indy Autonomous Challenge. This support is focusing more on covering the travel expanses than covering the costs of a PhD student.

5.3 Collaboration with other Universities

In general our team is open for collaboration with other Universities. In particular we are searching for a University that is helping us with the localization part. As described we have knowledge in the field of SLAM but these algorithms are not as robust as we would wish. In return we would offer our special trajectory, adaptive vehicle dynamics control and lap-time simulation knowledge (describe in subsection 4.4) as well as access to our HiL Simulator. In addition we would like to partner with an US University because it would safe time and costs for traveling to the US for us.

References

[1] Betz, Heilmeier, Wischnewski, Stahl, and Lienkamp. Autonomous driving—a crash explained in detail. *Applied Sciences*, 9(23):5126, November 2019.

[2] Johannes Betz, Alexander Wischnewski, Alexander Heilmeier, Felix Nobis, Thomas Herrmann, Tim Stahl, Leonhard Hermansdorfer, and Markus Lienkamp. A software architecture for the dynamic path planning of an autonomous racecar at the limits of handling. In *8th IEEE International Conference on Connected Vehicles and Expo (ICCVE 2019)*. IEEE, November 2019.

[3] Johannes Betz, Alexander Wischnewski, Alexander Heilmeier, Felix Nobis, Tim Stahl, Leonhard Hermansdorfer, and Markus Lienkamp. A software architecture for an autonomous racecar. In *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*. IEEE, April 2019.

[4] Johannes Betz, Alexander Wischnewski, Alexander Heilmeier, Felix Nobis, Tim Stahl, Leonhard Hermansdorfer, Boris Lohmann, and Markus Lienkamp. What can we learn from autonomous level-5 motorsport? In *Proceedings*, pages 123–146. Springer Fachmedien Wiesbaden, September 2018.

[5] Fabian Christ, Alexander Wischnewski, Alexander Heilmeier, and Boris Lohmann. Time-optimal trajectory planning for a race car considering variable tyre-road friction coefficients. *Vehicle System Dynamics*, pages 1–25, December 2019.

[6] Frederic Chucholowski, Stefan Buechner, Johannes Reicheneder, and Markus Lienkamp. Prediction Methods for Teleoperated Road Vehicles. In Markus Lienkamp, editor, *Conference on Future Automotive Technology*, München, 2013.

[7] Frederic Chucholowski, Michael Sauer, and Markus Lienkamp. Evaluation of Display Methods for the Teleoperation of Road Vehicles. In *9th International Conference on Intelligent Unmanned Systems*, Jaipur, 2013.

[8] Alexander Heilmeier, Maximilian Geisslinger, and Johannes Betz. A quasi-steady-state lap time simulation for electrified race cars. In *2019 Fourteenth International Conference on Ecological Vehicles and Renewable Energies (EVER)*. IEEE, May 2019.

[9] Alexander Heilmeier, Michael Graf, and Markus Lienkamp. A race simulation for strategy decisions in circuit motorsports. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, November 2018.

[10] Alexander Heilmeier, Alexander Wischnewski, Leonhard Hermansdorfer, Johannes Betz, Markus Lienkamp, and Boris Lohmann. Minimum curvature trajectory planning and control for an autonomous race car. *Vehicle System Dynamics*, pages 1–31, June 2019.

[11] Leonhard Hermansdorfer, Johannes Betz, and Markus Lienkamp. A concept for estimation and prediction of the tire-road friction potential for an autonomous racecar. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, October 2019.

[12] Thomas Herrmann, Fabian Christ, Johannes Betz, and Markus Lienkamp. Energy management strategy for an autonomous electric racecar using optimal control. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, October 2019.

[13] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[14] Felix Nobis, Johannes Betz, Leonhard Hermansdorfer, and Markus Lienkamp. Autonomous racing:a comparison of slam algorithms for large scale outdoor environments. In *Proceedings of the 2019 3rd International Conference on Virtual and Augmented Reality Simulations - ICVARS*. ACM Press, 2019.

[15] Felix Nobis, Maximilian Geisslinger, Markus Weber, Johannes Betz, and Markus Lienkamp. A deep learning-based radar and camera sensor fusion architecture for object detection. In *2019 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*. IEEE, October 2019.

[16] Institute of Automotive Technology. Vehicle dynamics and control systems. https://www.ftm.mw.tum.de/en/main-research/vehicle-dynamics-and-control-systems/, 2019.

[17] Tim Stahl, Alexander Wischnewski, Johannes Betz, and Markus Lienkamp. Multilayer graph-based trajectory planning for race vehicles in dynamic scenarios. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, October 2019.

[18] Tim Stahl, Alexander Wischnewski, Johannes Betz, and Markus Lienkamp. ROS-based localization of a race vehicle at high-speed using LIDAR. *E3S Web of Conferences*, 95:04002, 2019.

[19] Alexander Wischnewski, Johannes Betz, and Boris Lohmann. A model-free algorithm to safely approach the handling limit of an autonomous racecar. In *8th IEEE International Conference on Connected Vehicles and Expo (ICCVE 2019)*. IEEE, November 2019.

[20] Alexander Wischnewski, Tim Stahl, Johannes Betz, and Boris Lohmann. Vehicle dynamics state estimation and localization for high performance race cars. *IFAC-PapersOnLine*, 52(8):154–161, 2019.