Research on the Development of Autonomous Race Cars And Impact on Self-driving Cars

Alexander Jiang
Woodberry Forest School, 898 Woodberry Forest Rd, VA 22989

Author's Email: zander.jiang@gmail.com

Abstract. The rise of artificial intelligence has drawn attention to self-driving cars. Although a commercial autonomous vehicle still have not appeared on market, autonomous race cars are already fighting for the podium. One of the leading series is Roborace, a special series under FIA specifically to autonomous racing. Season alpha of the series was held in 2019 with two teams competing, Technical University of Munich and Arrival. Autonomous driving functionality contains a holistic software structure containing multiple individual driving functions. This paper analyzes the software employed by team TUM, a leader in autonomous racing software development, on their Devbot 2.0 racecar. The paper concludes the impact of autonomous racing on commercial self-driving cars along with a suggestion of what else can be done to further improve upon the concept.

1. Introduction
Since the last century, people have always attempted to create a fully functional self-driving vehicle that can relieve humans from driving. The development in computer science sheds light on the topic, allowing people to reach a closer step towards autonomy. The topic of autonomous vehicles is an extremely popular one in the current world. Car companies along with research labs and technology companies all try to develop their own self-driving vehicle. In fact, autonomy in automobiles are already present in our daily lives with ADAS (advanced driving assisting systems). However, functions such as cruise control and lane keeping assist are only considered level 0-2 autonomy by SAE. These functions require human participation in driving. The ultimate goal is to reach level-5 autonomy when the vehicle can drive itself in all conditions[1].

In the midst of self-driving vehicles, a division known as Roborace emerged pioneering in AI technology evolved in motor racing. Roborace utilizes the same tracks used by Formula E, with the racing teams only developing the software for autonomous racing and the organization providing autonomous platforms. In the earlier seasons the teams would “driving” Devbots, a bespoke platform developed by Roborace based on a le mans-prototype vehicle. The Devbots can be piloted by both human and software, allowing teams to explore the connection between men and machine for autonomous technology. Future races will be based on fully commit autonomous racing vehicles.[2] Level 5 autonomous racing is pushing the
boundaries of both hardware and software in a self-driving vehicle and is extremely crucial to its development.

Technical University of Munich is one of the leading organizations in autonomous racing, with its team being one of two to race in the first ever Roborace series. Therefore, this paper is going to focus on the software employed by the team in its 2019 Devbot racecar along with some proposed concepts that could be employed in the future.

2. Autonomous driving in general

Autonomous driving can be divided into 6 levels rating from 0-5 according to SAE[3]. Most ADAS (advanced driving assistance systems) seen in cars sold to the public fall in between the category of 0-3. Mercedes Benz recently launched their new W223 S-class which contains the most advanced driving assistance systems on the market. The system is based on Nvidia’s Drive AGX Orin platform, enabling the car to achieve level 3 automation that autonomously performing most of the driving tasks, but still requiring human monitor and override when necessary, and achieve level 4 during parking that full automation is under specific circumstances. Although the collaboration between Nvidia and Mercedes-Benz brings ADAS to a new level, it is still far from level 5 automation. Vehicle can self-perform in all conditions, not requiring any human interference. [4]

3. Level 5 autonomous software

A self-driving software consists multiple autonomous driving functions and an integrated architecture that connects the individual subtasks in order to achieve autonomy. Figure 1 shows the three basic steps that have to be conducted by the software: perception, planning, and control. This is same for all autonomous vehicles, no matter racing or urban use.

![Figure 1. Autonomous driving software structure](image-url)

3.1. Perception

In the perception stage, the “car” processes information provided by sensors that monitor the surroundings. For TUM, their Roborace Devbot contains 5 LiDAR sensors, 6 cameras, 2 radar sensors, and 17 ultrasonic sensors. Similar principal to sonar, LiDAR sensor is an instrument that calculates distance by shooting lasers and recording the time it takes for them to bounce back[5]. The information acquired from the
sensors are processed by the Nvidia Drive PX 2 ECU running the Linux operation system. The perception software is written in C++ along with open source ROS functions implemented into the language. The team uses gmapping SLAM algorithm (Simultaneous Localization and Mapping) to generate an occupancy grid map. SLAM is a really useful algorithm because it allows the car to generate a map of an unknown environment while also keeping track of the car’s location. The TUM team also uses an AMCL algorithm (Adaptive Monte Carlo Localization)[6] also named adaptive particle filter to locate the vehicle using LiDAR. This algorithm takes a similar concept of the MCL algorithm (Monte Carlo Localization), but improves its efficiency using KLD-sampling (Kullback-Leiber distance).[7,8] When a human drives the race-car around the track, the team incorporates LiDAR, GPS, and odometry data into a ROS data file that can later be used to create a 2D gridmap of the racetrack. The gridmap presents the car with 3 situations about the race track, occupied, unknown, and free of obstacles.

3.2. Planning
The information gathered from the perception stage is passed on to the planning stage through ZeroMQ. The planning stage’s goal is to calculate the optimal trajectory of the car. A layered approach is commonly used among autonomous driving software engineers. Team TUM uses Python to write the planning software, with global minimum curvature path planning along with speed profile planning and local trajectory planning to plan the racing line of the vehicle. [9, 10] The most efficient racing line is achieved when the variation of its curvature, or squared sum of the curvature values is minimal. [11] The team then uses a forward-backward algorithm to calculate the velocity profile. Apart from path planning, an additional behavior planner is also employed to make decisions while on track: safety breaking, overtaking, or lane changing.

3.3. Control
The control software runs independently on a real time motion controller, Speedgoat Mobile Target Machine, as Figure 2 shown. The software (written in MATLAB/Simulink) takes information about path and velocity profile from the planning stage and calculates the control output values used to control the electric motors, apply brakes, and steer direction. The real time trajectory controller runs at 250Hz to ensure fast response times. The software architecture employed by team TUM consists both online and offline sections.

![Figure 2. Software structure and interface for the ECUs in the robocar by TUM](image-url)
4. Developing concepts in autonomous driving

The current level 5 autonomous software is still in its initial stage. The software can only carry out functions of basic driving, more development is needed until an autonomous racer can defeat a human driver, and more development is needed until an autonomous vehicle can be mass-produced and serve the public. Nowadays, numerous functions are being developed to further enhance autonomous driving functions.

Apart from autonomous driving, another major shift in the automotive industry is the electrification of vehicles. The need for improvement in battery technology makes the range of electric vehicles vulnerable. Autonomous driving makes an EMS (energy management system) possible because the vehicle has a holistic understanding of the status of each individual component. For Roborace, the robocars are electronically powered. Team TUM is trying to develop an EMS based on the OCP (optimal control problem) of a time minimum global trajectory. [13] OCP is solved through explicit solution, where they calculate one-dimensional velocity profiles for different paths and minimize the energy demand of the car. MILP (mixed integer linear programming) or MIQP (mixed integer quadratic programming), which solves for the optimal path for driving maneuvers like overtaking and lane changing [14]. Graph search method is used to account complex models of vehicle dynamics, nonlinearities in objective functions. Convex optimization is useful for the OCP because it requires minimal time to solve the problem. Nonlinear programming is used to model complex dynamic scenarios (gear shifts, three-dimensional tracks, etc.), however, nonlinear programming suffers from long calculation time.

5. Impact of level-5 autonomous racing on future series production self-driving cars

The race car industry has always been the pinnacle of automotive technology. It is a standard practice now in the car industry where the latest technology is developed and tested in racing, and then utilized on other automobiles. This practice, known as the trickle-down effect, has existed for quite a long time. An example is hybrid technology, hybrid systems essentially combine an electric motor which sources its power from a kinetic energy recovery system (KERS) through braking with an internal combustion engine, increasing car’s efficiency and thereby making the car more economical and environmentally friendly to drive. [15]

Although the concept of a hybrid wasn’t first applied to race cars, it was formula 1 racing that leads to the enhancement in the technology and ultimately results in hybrid vehicles people purchase today. The trickle-down effect can also be applied in the artificial intelligence field of automotive, and racing can play a major role in future self-driving cars aimed to serve the commercial market.

A benefit of a racetrack compared with other testing grounds is that it can provide the car with a safe environment to test the vehicle at its limits. During races, cars have to constantly make maneuvers at speeds that may exceed 200 kilometers per hour, passing other cars, defending position, or driving around efficiently by taking the best racing line. These harsh conditions allow developers to fully test the software’s stability and performance.

The current method of gaining perception for autonomous vehicles is through 2D mapping algorithms that process laser scanned data from LIDAR sensors.[16] Although this is a commonly used method with lots of different ROS (Robot Operating System) mapping packages and tuning methods, current perception algorithms still lack the accuracy to allow mass-produced commercial self-driving cars to safely hit the road.

In the filed of racing, one of the most important aspects is line planning. Race cars have to take the most efficient line in order to achieve the best times. An optimal global trajectory can be difficult to achieve since it can vary due to factors like the tires, car performance, or track design. Passing through the apex, a point on the inner edge of the track, is an important technique in creating an efficient race line. It allows the vehicle to carry more speed through the corner and power up earlier for a faster exit speed. However, there are center apexes, early apexes, and late apexes. The best apex to hit depends on the overall track design, for example, a late apex is preferred when the corner is followed by a long straight. There are lots
of approaches to line optimization, such as optimal control and Euler spiral method, according to Figure 3. [17,18]. But how to plan trajectories that can meet all vehicle dynamic constraints remains an question. As the planning of autonomous racecars is enhanced, the software can be adjusted and applied to commercial self-driving vehicles, making their maneuvers safer and more efficient.

![Figure 3. The center, early and late apex[18]](image)

Steady control of an autonomous vehicle in a corner at high speeds still remains a problem, although there are a few working models, such as “Stanley”, the Stanford robot that won the 2005 Darpa challenge, and the TUM devbot.[9,19] Due to the environment, autonomous vehicles heavily depend on different sensors to give it perception and a network of decentralized algorithms. There are uncertain time delays and higher chance of failure in the complicated system. At high speeds, the delays will be aggrandized and heavily affect the vehicle’s control performance. There are attempts to solve the multi-sensor fusion issue such as the novel multisensory approach. [20] But current methods are too fragile to meet the requirements for commercial autonomous vehicles to use.

6. Conclusion
This paper presented the autonomous software used by the Roborace team of Technological University of Munich. Although they have already achieved level 5 autonomy, more can still be done. Some concepts still in development were presented including an energy management system, tire friction prediction system, as well as a model-free algorithm extension to current planning software. Instead of simply replacing human controls, developers should make use of the integrity between hardware and software. The “vehicle” has access to more information than a driver, and that knowledge could be used to make cars safer and more efficient. The further development of autonomy will push the automotive industry into a new era, allowing cars to achieve tasks unachievable with a driver. Autonomous racing is necessary in this development because it can provide a safe environment for researchers to push their software to the limits. There are still problems that need solving before a commercial self-driving vehicle that can be mass-produced hitting the market. Autonomous racing provides a perfect environment for engineers to test their resolutions.

7. References
[1] SAE International Releases Updated Visual Chart for Its "Levels of Driving Automation" Standard for Self-Driving Vehicles. (2018, November/December). SAE International. https://www.sae.org/news/press-room/2018/12/sae-international-releases-updated-visual-chart-for-its-%E2%80%9Clevels-of-driving-automation%E2%80%9D-standard-for-self-driving-vehicles

[2] Roborace, Robocar. [Online] Available: http://roborace.com/. Accessed on: Nov 28 2020

[3] SAE International, “Automated Driving - Levels of Driving Automation,” [Online] Available: https://www.sae.org/misc/pdfs/automated_driving.pdf.

[4] Shapiro, D. (2020, June 23). Mercedes-Benz, NVIDIA Partner to Build the World's Most Advanced, Software-Defined Vehicles. Nvidia. https://blogs.nvidia.com/blog/2020/06/23/mercedes-benz-nvidia-software-defined-vehicles/

[5] How does LiDAR work? [Fact sheet]. (n.d.). Lidar UK. Retrieved November 28, 2020, from http://www.lidar-uk.com/how-lidar-works/

[6] Stahl, Tim & Wischnewski, Alexander & Betz, Johannes & Lienkamp, Markus. (2019). ROS-based localization of a race vehicle at high-speed using LiDAR. E3S Web of Conferences. 95. 04002. 10.1051/e3sconf/20199504002.

[7] Fox, Dieter. (2001). KLD-Sampling: Adaptive Particle Filters and Mobile Robot Localization.

[8] Stahl, Tim & Wischnewski, Alexander & Betz, Johannes & Lienkamp, Markus. (2019). A Software Architecture for an Autonomous Racecar. 1-6. 10.1109/VTCSpring.2019.8746367.

[9] Betz, Johannes & Wischnewski, Alexander & Heilmeyer, Alexander & Nobis, Felix & Stahl, Tim & Hermansdorfer, Leonhard & Lienkamp, Markus. (2019). A Software Architecture for an Autonomous Racecar. 1-6. 10.1109/VTCSpring.2019.8746367.

[10] C. Liu, W. Zhan and M. Tomizuka, "Speed profile planning in dynamic environments via temporal optimization," 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, 2017, pp. 154-159, doi: 10.1109/IVS.2017.7995713.

[11] T. Berglund, A. Brodnik, H. Jonsson, M. Staffansson and I. Soderkvist, "Planning Smooth and Obstacle-Avoiding B-Spline Paths for Autonomous Mining Vehicles," in IEEE Transactions on Automation Science and Engineering, vol. 7, no. 1, pp. 167-172, Jan. 2010, doi: 10.1109/TASE.2009.2015886.

[12] Betz, Johannes & Wischnewski, Alexander & Heilmeyer, Alexander & Nobis, Felix & Stahl, Tim & Hermansdorfer, Leonhard & Lohmann, Boris & Lienkamp, Markus. (2019). What can we learn from autonomous level-5 motorsport?: chassis.tech plus. 10.1007/978-3-658-22050-1_12.

[13] T. Herrmann, F. Christ, J. Betz and M. Lienkamp, "Energy Management Strategy for an Autonomous Electric Racecar using Optimal Control," 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 2019, pp. 720-725, doi: 10.1109/ITSC.2019.8917154.

[14] X. Qian, F. Altche, P. Bender, C. Stiller, and A. de La Fortelle, Optimal Trajectory Planning for Autonomous Driving Integrating Logical Constraints: An MIQP Perspective, in 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC). Piscataway, NJ: IEEE, 2016.

[15] The F1 Trickle Down Effect. (n.d.). Budget Direct. Retrieved November 28, 2020, from https://www.budgetdirect.com.au/interactives/special-feature/f1-trickle-down-effect.php

[16] A. R. Corregedor, J. Meyer, and F. Du Plessis, “Design principles for 2D local mapping using a laser range finder,” in AFRICON 2011, Victoria Falls, Living- stone, Zambia, pp. 1–6.

[17] Xiong, Ying. (2011). Racing Line Optimization. https://dspace.mit.edu/bitstream/handle/1721.1/64669/706825301MIT.pdf?sequence=2&isAllowed=y

[18] Gustafsson, T. (2008). Computing The Ideal Racing Line Using Optimal Control [D]. Institutionen för systemteknik https://www.diva-portal.org/smash/get/diva2:17976/FULLTEXT01.pdf

[19] Betz, Johannes & Wischnewski, Alexander & Heilmeyer, Alexander & Nobis, Felix & Stahl, Tim & Hermansdorfer, Leonhard & Lienkamp, Markus. (2019). A Software Architecture for an Autonomous Racecar. 1-6. 10.1109/VTCSpring.2019.8746367.
[20] K. Hausman, S. Weiss, R. Brockers, L. Matthies and G. S. Sukhatme, “Self-calibrating multi-sensor fusion with probabilistic measurement validation for seamless sensor switching on a UAV,” 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, 2016, pp. 4289-4296, doi: 10.1109/ICRA.2016.7487626.

Acknowledgments

I would like to thank professor Rakesh Kumar from University of Illinois at Urbana Champaign, Jiatong, and Olivia Sun for helping me construct the paper.