

Design of an Autonomous Race Car for the Formula Student Driverless (FSD)

Marcel Zeilinger¹, Raphael Hauk¹, Markus Bader² and Alexander Hofmann³

Abstract—Formula Student Germany is a race car competition for student teams competing with self-designed, self-developed and self-built vehicles. These cars have been competing to win in various disciplines every year since 2006 at the Hockenheimring in Germany. In 2016, a new discipline, Formula Student Driverless, was announced for the following year, targeting autonomous race cars that complete tracks of 5 km over 10 laps as fast as possible without the help of human racer pilots or remote control systems. This paper will cover the framework, the sensor setup and the approaches used by the Viennese racing team TUV Racing.

I. INTRODUCTION

Formula Student Germany (FSG) is an international construction competition for students with the aim of promoting research in multiple disciplines on a concrete application.

The competition is divided into three classes: electric, combustion and driverless vehicle. The last classification, which has existed since 2017, is called Formula Student Driverless (FSD), where either combustion or electric cars are permitted. Contrary to the other two classes, the driverless class rules permit the use of a team's *existing race car* which has already attended previous FSG race(s). Therefore, formerly manually-driven race cars equipped with appropriate sensors, actors and computing hardware can be adapted to driverless cars. Nevertheless, vehicles have to comply with the restrictions of their appropriate class; in particular, manual operation must still be possible.

Regardless of the class in which a team participates, the event itself is divided into static and dynamic disciplines, and points can be scored in each discipline. In the static discipline, a cost report and a business plan must be presented by each team. Before a car is allowed to participate in a dynamic event, a technical inspection must be passed to ensure the car is mechanically and electrically safe, in accordance with the rules [3]. Since a car's total score is comprised of both disciplines, the fastest car need not necessarily win.

In the following sections TUV Racing will describe its concept for participation in the FSD 2017. First, an overview is provided of the solutions and academic approaches available in the area of autonomous driving, with regard to similar



Fig. 1. Racecar Edge8 from the previous season (2016) which has been redesigned to drive autonomously

solutions in general mobile robotics. Then an overview of the vehicle and the sensors, actuators, as well as the software components used is provided. Furthermore, we will describe our approach to the problems posed by the competition and reflect upon its effectiveness up to this point.

II. RELATED WORK

In order to describe related work, time was invested in studying competitions with a similar focus. In addition, studies were conducted on commercially available solutions to autonomous driving such as the NVIDIA DRIVE¹ products.

The Carolo Cup² is a competition where student teams build autonomous 1:10 sized cars, which have to cope with reality-inspired driving situations faced by passenger cars [12]. While there is a static competition in the form of a concept presentation as well, the dynamic events consist of parking and driving a free drive with and without obstacles. During the free drive, the model car needs to master challenges like intersections, speed limits or dynamic obstacles such as other vehicles or passengers crossing the track, which is marked with white lines on a dark ground similar to real road surface markings. In the paper [12], *Technical Evaluation of the Carolo-Cup 2014*, the authors state under *Lessons Learned* the importance of high quality hardware products and the need for a robust computer vision algorithm in detecting road markings. Similar to this competition are the NXP Cup student competition, also known as the Freescale

¹Marcel Zeilinger and Raphael Hauk are with TUV-Racing, Adolf-Blamauergasse 1-3, 1030 Wien, Austria firstname.lastname@racing.tuwien.ac.at

²Markus Bader is with the Automation Systems Group at Institute of Computer Aided Automation, TU Wien, Karlsplatz 13, 1040 Vienna, Austria firstname.lastname@tuwien.ac.at

³Alexander Hofmann is with the Institute of Computer Science at the University of Applied Sciences FH Technikum Wien, Höchstädtplatz 6, 1200 Vienna, Austria firstname.lastname@technikum-wien.at

¹NVIDIA DRIVE <http://www.nvidia.com/object/drive-px.html>

²Carolo Cup: <https://wiki.ifr-ing.tu-bs.de/carolocup/>

Cup³, and the Crazy Car⁴ race, which has been hosted by FH Joanneum in Austria since 2008, where students from schools and universities are eligible for participation.

Other relevant competitions comparable to the FSD are the DARPA Challenges of 2004 to 2007. The car Stanley [9] won this competition in 2005, but since that time many products in the field of autonomous driving have entered the market.

Autonomous control is an important new subfield in the automotive sector as commercial autopilots and driver assistance systems become more and more popular.

To classify the different levels of system functionality and compare actual driverless cars, the Society of Automotive Engineers has introduced a classification scheme⁵.

Many commercially available software products fall into classes 0-2, thus only observing driver environment and taking on only limited driving tasks. The rare cases of actual automated driving systems in classes 3-5 can be found in Google Autonomous Cars or Tesla Autopilots. Google makes heavy use of a multi-beam 3D-laser scanner to understand the entire environment. It also solves problems related to the traffic behaviour of human drivers and it is integrated into external maps and weather services. Tesla uses radar and sonar, in some cases mounted behind the vehicle's outer structure, to detect individual classes of obstacles, e.g. with radar: moving obstacles; or, with ultrasonic sensors: other cars beside the vehicle. [1] [5]

Nvidia presented an autonomous control solution where the system was taught to drive exclusively by RGB camera input [2]. Machine Learning was used to match steering input with camera images, so the vehicle steers based on scenes recognised in the camera view. Nevertheless, the driver has to control the throttle.

An example of a framework for autonomous driving is provided by Nvidia DriveWorks. It provides interface layers that allow easy incorporation of new sensors, execute machine vision algorithms and even perform trajectory planning. Advantages are its range of out-of-the-box detection algorithms and the heavy use of graphics-accelerating hardware with perception computation times of within a few milliseconds. An apparent disadvantage is that only Nvidia hardware can be used. Another drawback is that the architecture lacks an actuator control component.

The approaches and frameworks in Google's, Tesla's and Nvidia's applications solve a variety of problems associated with driverless cars in traffic that are not relevant to the FSD competition. Accidents, recognising people, lane markings, sidewalks or traffic signs do not have to be handled at all in the FSD. There are no intersections where maneuvers have to be negotiated with other cars; for that fact, there no other cars on the track at all. This year, TUW Racing's main goal is to design a solid base system as an ideal start for improvements and qualitative increments in upcoming years.

Therefore our solution is based on open-source components. Due to the fact that TUW Racing initiated the project with experts in mobile robotics, the commonly used Robot Operating System ROS[7] is used for modularisation and communication.

ROS enables TUW Racing to interface the required components of the system, from camera to motor control. Therefore, for speedy development, the team selected hardware devices with drivers already available.

In the next section, the hardware and the software frameworks used will be described.

III. THE RACE CAR

In this section, the competition, the race car's hardware, as well as the software implemented are described in detail.

A. Competition

The dynamic component of the race has three challenges: an acceleration race, a skid pad and a track drive. The acceleration race is a 75m-long straight track followed by a 100m straight exit lane. The skid pad track consists of two congruent circles, touching externally, with a diameter of 18.25m. The vehicle enters on the tangent of the circles at their contact point, drives the right circle twice followed by the left twice before leaving via the tangent again. The track drive is a closed loop circuit with an unknown layout consisting of up to 80 m straights, up to 50 m diameter constant turns, and hairpin turns with a 9 m diameter, among other miscellaneous features such as chicanes, or multiple turns. The vehicle must recognise and drive exactly ten laps with a maximum distance of 500 m each. The track is marked by blue and white striped traffic cones on the car's left edge and yellow and black striped cones on its right edge. The cones are connected via a high-contrast colored line sprayed onto the road; however, the line may extend out to the side. Additionally, the stop zones on the acceleration and skid pads are marked using orange and white striped cones. The distance from one cone to another along either edge of the track is up to 5 m, and the minimum track width is 5 m, except on the skid pad, where the width is 3 m. The traffic cone layout is predefined in the rules.

B. Hardware

The total power allowed for formula student cars with any kind of powertrain is 80kW, and for electric powertrains a maximum voltage of 600V at any time is allowed. The other major limitation is the wheelbase of at least 1525mm. Aside from those stipulations, students are free to design their cars' characteristics as they wish.

Both rear wheel hub motors of the TUW Racing car have a maximum torque of 30 Nm at a weight of 3.6kg and a gear transmission ratio of 12:1. The total weight of the car is 160.5kg with a top speed of 30.5m/s, accelerating in 3.1 seconds to 27.8m/s. Fig. 2 shows a rough sketch of the vehicle with sensors.

³Freescale Cup: <https://community.nxp.com/docs/DOC-1011>

⁴Crazy Car: <https://fh-joanneum.at/projekt/crazycar/>

⁵SAEReport: https://www.sae.org/misc/pdfs/automated_driving.pdf

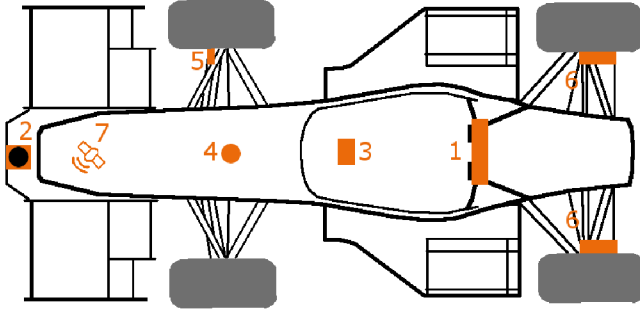


Fig. 2. Sensor placement on the vehicle: 1 stereo camera, 2 laser scanner, 3 IMU, 4 steering angle encoder, 5 wheel speed encoder, 6 rotor position encoder, 7 GPS

1) *Actuators*: TUV Racing had to adapt its vehicle with actuators that still allow for human operation of the vehicle. The brake pedal may not be blocked and the steering must be easily steerable by hand despite the gearboxes and motors that are to be added.

- **Brake System**: Due to the already-optimised brake balance, the brake system is mechanically operated via the pedal. Details in the mounting allow the driver to still press the pedal. An alternative way to decelerate is through reverse operation of the motors, although brake response is faster and conserves energy. The brake system must always be able to stop the car within a maximum of 10 m, even in the face of a single failure in the system, including power loss or any mechanical failure.
- **Steering**: The additional steering motor is mounted to the existing steering strut at the top of the monocoque. With an average-size driver in the vehicle, the car weighs about 240 kg and requires 25 Nm to steer while the car is not moving. This is the force TUV Racing designed for, since it would enable testing with a driver.

The competition requires students to design emergency systems in a detailed Failure Mode and Effects Analysis (FMEA). For instance, power or mechanical failure to the brakes or steering must be accounted for with fallback systems. When emergency braking is initiated by remote or failure detection in another subsystem, the vehicle must enter a safe state that simultaneously relies on the actuator's operation. For instance, a vehicle steering 60 degrees to the left while a full brake is initiated should first steer to the center position to optimise friction on the wheels.

2) *Sensors*: Sensor placement on the vehicle is shown in Fig. 2.

- **Camera**: TUV Racing selected a ZED⁶ stereo camera for its visual sensors. It features an opening angle of 90deg and a base line of 120mm, connected via USB3, which enables generation of depth images at a range of 20m. The camera is mounted on the topmost point of the roll bar (1). In order to automate the calibration process and to estimate the extrinsic camera matrix, the team used visual markers attached to the vehicle's chassis on

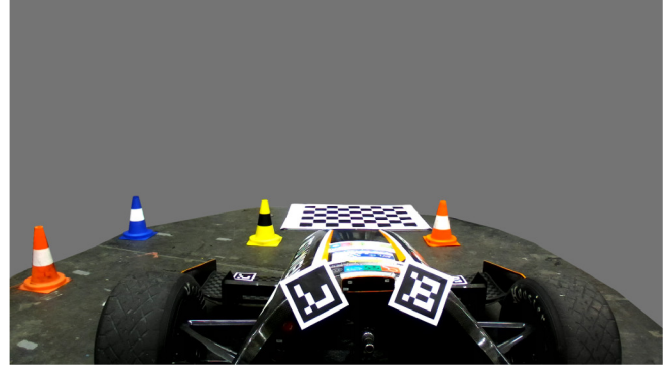


Fig. 3. Image of left camera with visual markers attached to vehicle to estimate extrinsic camera matrix

specific locations, as shown in Fig. 3

- **Laser Scanner**: A laser scanner was placed inside the front wing, at the lowest point, as it is a planar scanner and the cones are relatively short. It is tilted downwards, so that it points at the bottom of a cone from the maximum distance. This counteracts damping influence, which could lead to the rising of laser orientation during accelerations, and the lowering thereof during braking. With the use of (2) a Hokuyo 20LX planar laser scanner, the autonomous system is able to detect cones within the field of view using circle detection and heuristics to filter non-cone circular obstacles, e.g. wall detection.
- **GPS**: For accurate absolute position measurement, a dGPS, provided by a Piksi Multi GNSS module (7), is used along with two beacons placed outside the race-track. The beacons allow for more precise positioning than a common GPS system does.
- **IMU**: The relative movement of the vehicle is measured by a motorsport-grade IMU (3), which measures rotation in the yaw axis as well as acceleration in the x and y directions.
- **Odometry**: To accurately determine the front wheels' speed and distance travelled, TUV Racing uses an inductive wheel spin sensor (5) at each front wheel, and for the rear wheels a rotor position encoder is used for the car's TUV-Racing-developed motors(6). The steering angle is measured by a rotary position encoder (4) connected to the steering shaft. All measurements acquired from these devices are used within the software framework which is described in the following section.

C. Software

In order to integrate and process sensor measurements, ROS is used as the base framework and nodes were implemented for the following tasks.

- **SLAM (Simulations Localization and Mapping)**: The vehicle's pose must be estimated and the race track must be reconstructed while driving[8].
- **Machine vision**: The traffic cones, their positions, and colors, must be detected using multiple cameras and a laser range sensor.

⁶ZED Stereo Camera: <https://www.stereolabs.com/>

- **Path extraction:** A path must be computed which guides the vehicle between the traffic cones to the destination and marks the drivable area.
- **Motion control:** The motion controller uses the drivable area detected between the cones, computes an optimal trajectory, and dispatches the actual motion commands for steering and accelerating the car.

In Fig. 4 one can see an overview of the messaging system among the ROS components. It shows the connection of the actuators to the sensors, as well as the processing steps needed to achieve desired control sequences. Some of them are implemented as nodelets to minimise messaging delays.

In the following section techniques used are described in more detail.

IV. APPROACH

The EKF-SLAM implementation used is based on Macsek's [6] Master's thesis. The filter uses perception of cone measurements with uncertainties and generates a map of the cones in 2D. It also determines the car's own position on the map, computes its position relative to previous laps, and performs motion control. It is necessary to predict the vehicle's pose ahead of time in order to send proper motion commands, since actuators have a considerable reaction delay.

For sensing, it is vital to detect traffic cones in the environment and estimate their location relative to the sensors. The known location of the sensors on the vehicle enables one to create a global map of landmarks for the trajectory-planning algorithm.

In the first development phase, TUV Racing used Pioneer 3AT in a simulated and real world scenario in order to generate measurements, try out sensors and plan algorithms, as shown in Fig. 5.

A. Traffic cone detection

Because of the variety of sensors used (lasers and cameras) various techniques have to be implemented accordingly. Two approaches are used to perform cone detection and position estimation on the stereo camera images. In the first approach, a depth image based on a block-matching algorithm is used. This allows for extraction and removal of planes in a 3D data set. The track floor that is geometrically known can be pre-segmented, and cone candidates appear as isolated objects. These candidates are then further evaluated using classical image processing steps on the estimated location in the image plane, similar to the approach of Yong and Jianru [11]. However, a block matching algorithm cannot be easily adapted to use the known structure of the environment for improved perception of target objects. Moreover, the computation effort and thus hardware requirements would be unnecessarily high, since one would have to operate on an entire image.

In the second approach, cones can be detected in both images separately. The algorithm computes the disparity, the z-depth and consecutively the 3D position of the cone, based

only on the UV center points of the same object determined in the left and right images.

In both approaches, the program determines the colors heuristically with RGB color thresholds. If the estimated center point is within a white or a black stripe, the color sample point is moved downwards until another color match is determined.

In addition, the planar laser scan is searched for point clusters of distances that match the radius of the cone at the height scanned. If the total distance spanned by coherent scan points is above this radius but below the total radius of the cone base, the algorithm considers the center point between the cluster to be the center point of a cone. In order to avoid detection of obstacles other than cones, the detector filters long connected components such as walls.

The detection results of both algorithms are then fused with a feature-based EKF-SLAM [6]. Thus the team benefits from the advantages of both of the detection approaches, because the fusion algorithm respects the reliability of each detection result and takes the more reliable detection result into greater account. For example, distance estimation at high distances is less accurate at image-based detection, due to the fact that pixel disparity is smaller. On the other hand, laser-based detection cannot reliably detect cones that are behind other cones.

B. Mapping

For TUV Racing, it is essential that the mapping components be able handle failures in the detection component. The camera or the laser could have short periods of blindness due to sunlight or uneven road conditions. In these cases a cone might be missed. Since cones are guaranteed to be at most 5 m apart, missing ones can be filled in. Because all cones on the left side have a different color than those on the right, heuristics can fill in cones missing from one side based on the cones observed on the other side.

Since the traffic cones themselves are the best reference for position and orientation on the track, the team uses an EKF-SLAM that maps them and corrects the vehicle's position based on their perceived change of position. An additional challenge here is that the same track sections are driven ten times throughout the competition. Thus, the mapper must recognize previously mapped cones by ID, even though they are of the same color and shape. An index counter from the starting line is insufficient due to potential perception failure. Nor does the vehicle have a full 360deg line of sight. In a narrow left turn, due to the vehicle's pose in an optimal trajectory, it is possible that only the outer cones might be seen. If the mapping algorithm performs corrections to positions on the map on those right lane cones, the left lane cones must also be updated automatically.

C. Trajectory Planning

From the mapping one can derive a path which lies in the center of the travel area; path-planning is therefore obsolete. To start, a normal vector is projected from the center point between the first left and right cones forward. The first cones

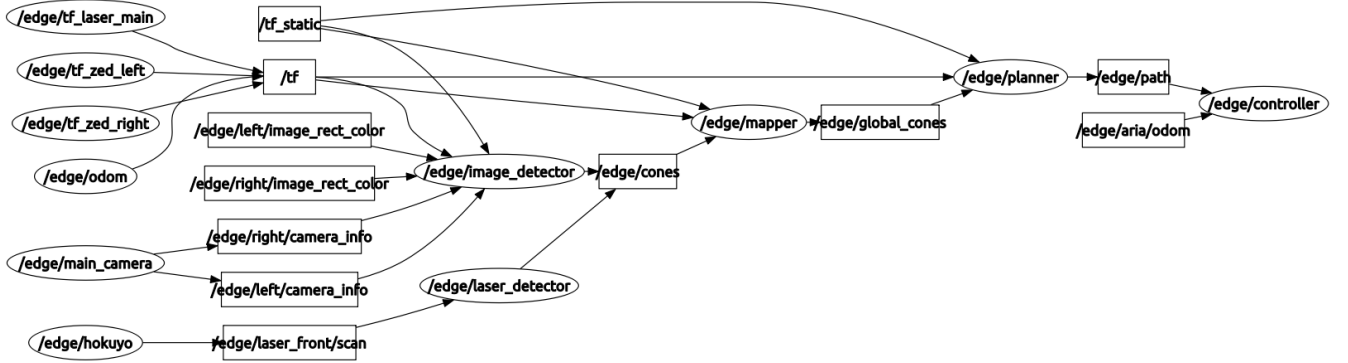


Fig. 4. ROS nodes and message subscriptions in rqt_graph

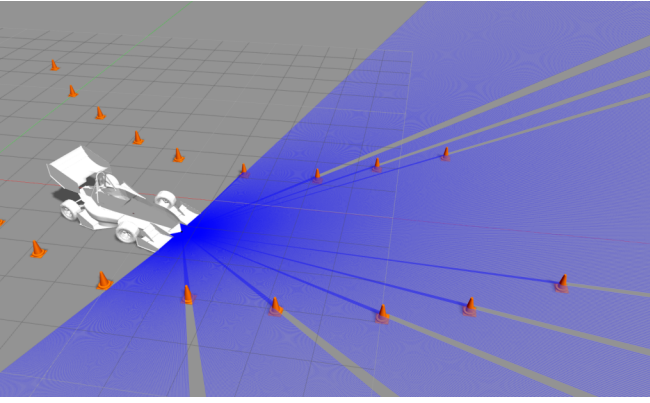


Fig. 5. Edge8 in a simulated environment using GazeboSim.

to the left and right of this vector are identified, then the center point identified, and a vector projected forward again. Since the left and right sides can be distinguished, a situation where the cut to the left of the vector which contains a right-side cone, or vice-versa, is manageable. The vector is then rotated accordingly so that the sides are split correctly.

But following the center path would not be very efficient, so the model predictive [4] motion controller computes its own optimal local trajectory. The MPC [10] used optimises the trajectory using the latest perception state and incorporates the mechanical constraints of the vehicle's motion model, such as maximum linear acceleration and deceleration, actuator delays and disturbances resulting from vehicle dynamics, e. g. sideslip angle.

V. RESULTS

In order to build a foundation for simulation of the race car in software that integrates well into ROS, TUV Racing extended its approach to the Pioneer 3AT and used gazebo simulations to test marker mapping, odometry corrections based on simulated IMU data, trajectory planning, as well as a basic self-developed motion controller. Fig. 5 shows the robot following a narrow turn in the simulation, with obstacles of width and height comparable to those of cones on the track. It successfully navigates a 180deg turn, detects shapes based on simulated laser scans and determines the center path accurately.

In Fig. 6 results with a cone detection algorithm on a still from a video of last year's autocross race at the FSG



Fig. 6. Cone detection on still of last year's race recorded using a GoPro camera

are shown. The cones detected in the image are enclosed by yellow rectangles. The algorithm is based on a trained cascade classifier using a histogram by an oriented gradients feature descriptor. Photos of the specific cones seen in the image, from angles of 0deg horizontal to about 40deg turned downwards, served as training data. The algorithm turned out to be computationally expensive and susceptible to variations in lighting and weather conditions. With the trained algorithm on last year's video we achieved an unstable detection rate of 30% to 90% depending on the track section. The algorithms explained above, evaluated on recordings with the stereo camera, detected 75% of the traffic cones in the image regardless of lighting. Most of the missed cones were affected by very strong shadows cast by large nearby obstacles, wear of the cone itself or cones that partially covered each other. The location, color and shape of the cones, the point of view, background, the vehicle itself and lighting are very similar to the scene in the image.

Evaluation of detection on a recording of last year's race has shown that different light conditions can be a problem for the car's visual sensor. Extreme light exposure to the camera or even driving from shade into sunlight can be disturbing, as the camera needs some time to adjust to the brightness. Similar issues emerged in evaluation of the stereo camera mounted on the car driving a short sample track. In Fig. 7, the camera adjusts to increasing brightness using integrated exposure control when the car enters a sunlight-drenched part of the track. As the more distant cones become visible by the aperture's adjustment, the cone marked with the orange



Fig. 7. Images of the ZED stereo camera showing how light conditions affect the visual cone appearance.

circle gets darker and less visible.

The Lidar suffers from similar issues, as can be seen in Fig. 8. The cone detection algorithm was tested in an outdoor environment during rain. Since dealing with environmental conditions is part of the challenge of Formula Student, the vehicle will have to run regardless of the current weather. The small axis markers show laser scan points that hit a cone in the vicinity of the robot. The small red dots represent points relative to the robot where an obstacle was detected. The white visual markers on the field show the cones the robot itself is currently detecting; the numbered axis markers show the cones mapped. Usually the field would be empty at the sensor's maximum range, but during rain, reflections are measured at some point, resulting in a noisy outer cloud structure. Near the robot some of the rays are instantly reflected and reduce the amount of rays that hit a cone, thus making accurate detection more difficult. Fig. 8 shows that the algorithm used is able to map cones with decreasing accuracy but at higher distances. Since scan data has to be compared over time until a cone can be classified correctly, the speed of the detection algorithm is also adversely affected.

One can expect the effect of small rocks on the track to have similar effects on detection, as at high speeds they are usually sprayed across the track by the vehicle.

VI. CONCLUSION

In this paper the TUV Racing team's approach to the FSD 2017 competition was presented. Details were provided on its hardware and the approach to its control software.

The team will begin testing within the next month on a test site with road conditions that resemble the event location and a setup with obstacles corresponding to those detailed in the rules. Initially the team will validate the accuracy of the motion model and perception system in separate road sections. Then it will complete entire tracks at low speeds before gradually increasing speeds while improving controller parameters.

While TUV Racing's main goal this year was successful participation in the first Formula Student Driverless event, its goal for further seasons will extend to improving the efficiency of the vision algorithms and accuracy of the underlying motion model. The choice of the sensors, actuators

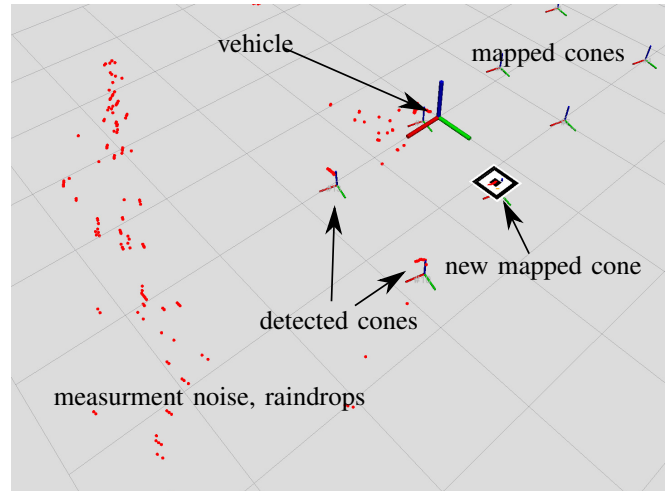


Fig. 8. ROS Rviz view of an outdoor test with the pioneer during rain. The laser scan (red) is disturbed by the raindrops.

and computers as well as the software framework will be influenced by performance in this year's competition.

ACKNOWLEDGMENT

The project is supported by TTTech Automotive GmbH.

REFERENCES

- [1] "How Google's Self-Driving car works," Accessed: 14-March-2017. [Online]. Available: <http://spectrum.ieee.org/automaton/robotics/artificial-intelligence/how-google-self-driving-car-works>
- [2] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to end learning for self-driving cars," *CoRR*, vol. abs/1604.07316, 2016. [Online]. Available: <http://arxiv.org/abs/1604.07316>
- [3] Formula Student Germany e.V. Formula Student Rules. Accessed: 14-March-2017. [Online]. Available: <https://www.formulastudent.de/fsg/rules/>
- [4] T. Howard, M. Pivtoraiko, R. Knepper, and A. Kelly, "Model-Predictive Motion Planning: Several Key Developments for Autonomous Mobile Robots," *Robotics Automation Magazine, IEEE*, vol. 21, no. 1, pp. 64–73, March 2014.
- [5] S. Ingle and M. Phute, "Tesla Autopilot : Semi Autonomous Driving, an Uptick for Future Autonomy," vol. 3, no. 9, Sep 2016.
- [6] M. Macsek, "Mobile Robotics: EKF-SLAM using Visual Markers for Vehicle Pose Estimation," Master's thesis, Vienna University of Technology, 2016.
- [7] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA Workshop on Open Source Software*, 2009.
- [8] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.
- [9] S. Thrun and et al., "Stanley: The robot that won the DARPA Grand Challenge," *Journal of Field Robotics*, vol. 23, no. 9, pp. 661–692, 2006. [Online]. Available: <http://dx.doi.org/10.1002/rob.20147>
- [10] G. Todoran and M. Bader, "Expressive navigation and Local Path-Planning of Independent Steering Autonomous Systems," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 4742–4749.
- [11] H. Yong and X. Jianru, "Real-time traffic cone detection for autonomous vehicle," in *2015 34th Chinese Control Conference (CCC)*, July 2015, pp. 3718–3722.
- [12] S. Zug, C. Steup, J. B. Scholle, C. Berger, O. Landsiedel, F. Schuldt, J. Rieken, R. Matthaei, and T. Form, "Technical evaluation of the Carolo-Cup 2014 - A competition for self-driving miniature cars," in *2014 IEEE International Symposium on Robotic and Sensors Environments (ROSE) Proceedings*, Oct 2014, pp. 100–105.