F1TENTH: Enhancing Autonomous Systems Education Through Hands-On Learning and Competition

Johannes Betz*, Hongrui Zheng*, Felix Jahncke, Zirui Zang, Florian Sauerbeck, Y. Rosa Zheng, Joydeep Biswas, Venkat Krovi, Rahul Mangharam

***Abstract*—Teaching autonomous and intelligent transportation systems in higher education has traditionally focused on theory, often lacking comprehensive coverage of the practical techniques required for real-world applications. To overcome this, we developed a new university course centered around hands-on learning with a modular autonomous small-scale vehicle platform called F1TENTH. This paper presents a detailed overview of the new course design, its underlying philosophy, the individual teaching modules, and the modular hardware/software of the F1TENTH platform. This new course was then evaluated with a survey conducted at five universities that have adopted the teaching modules for their semester-long undergraduate and graduate courses. The results show that approximately 80% of all involved students strongly agree that the hardware platform and modules significantly increased their motivation to learn. More than 70% of the students agreed that the hardware enhanced their understanding of the material The findings demonstrate that our course setup and the F1TENTH hardware effectively combine theoretical knowledge with practical application, greatly enhancing the educational outcomes and the students' computational thinking skills. Future research is needed to explore the long-term impact of hands-on learning on students' career development in intelligent autonomous systems.**

***Index Terms*—Autonomous vehicles, robotics, computational thinking, machine learning, control, simulation**

I. Introduction

Autonomous vehicles can potentially disrupt our transportation systems as we know them. They are expected to optimize street capacity, resulting in more efficient traffic flow [1]. Furthermore, autonomous vehicles could create $488 billion [2] in annual savings by reducing traffic accidents and additional savings due to reduced fuel costs and, therefore, reduced emissions [3]. To achieve autonomous capabilities, the senses and actions of a human driver are emulated by suitable sensors, actuators, and respective software [4].

Manuscript received XX 2022; revised XX XX, 2022. Corresponding author: J. Betz (email: johannes.betz@tum.de)

*: Equal Contribution

J. Betz, F. Jahncke, F. Sauerbeck are with Professorship Autonomous Vehicle Systems and Munich and Munich Institute of Robotics and Machine Intelligence (MIRMI), Technical University of Munich, 85748 Garching, Germany (e-mail: johannes.betz@tum.de)

H. Zheng, Z. Zang, and R. Mangharam are with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, USA (e-mail: hongruiz, zzang, rahulm@seas.upenn.edu)

R. Zheng is with the College of Engineering and Applied Science, Lehigh University, Bethlehem, PA 18015, USA (e-mail: yrz218@lehigh.edu)

J. Biswas is with the Department of Computer Science, University of Texas Austin, Austin, TX, USA (e-mail: joydeepb@cs.utexas.edu)

V. Krovi is with the College of Engineering, Computing and Applied Sciences, Clemson University, Clemson, SC 29634, USA (e-mail:vkrovi@clemson.edu)

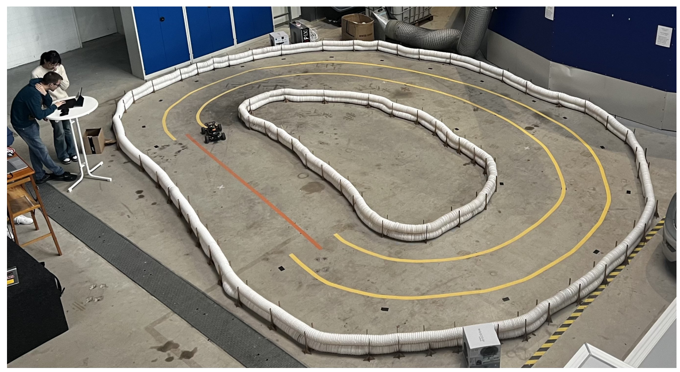


Fig. 1. An F1TENTH vehicle alongside a typical track layout at one of the participating universities. This setup offers students a safe environment to experiment with various algorithms in real-world scenarios.

Unfortunately, developing solutions for autonomous driving invokes complexities since it requires well-trained engineers with broad and expert knowledge in machine learning for embedded systems, control theory, and optimization [5]. There will be an increasing demand for specialized engineers, and teaching autonomous systems topics at higher education institutions can be seen as a global strategic initiative [6] [7]. However, current robotics and autonomous systems course curricula lack hands-on teaching and actual hardware usage, and literature reviews agree that teaching autonomous systems in higher education needs to be enhanced to facilitate learning at an early level [6] [7]. While teaching the foundational theory of autonomous systems remains important, offering deeper insights into applying software on real hardware is equally essential.

To overcome this issue, we created a new course for teaching autonomous systems in a more applied way with new modular autonomous vehicle hardware (Figure 2). We provide three autonomous systems themes: 1) foundations, 2) advanced, and 3) multi-vehicle, which are split into six different course modules (A-F). First, the students learn the theoretical foundations in each of these modules and then have

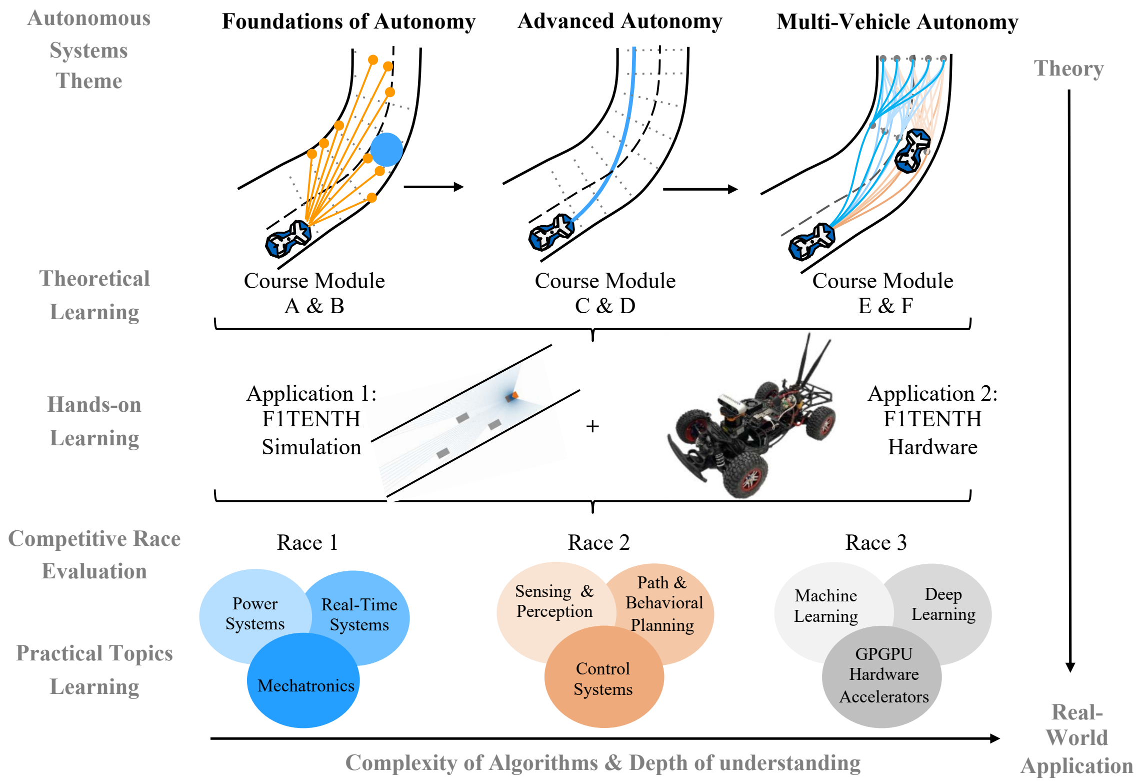


Fig. 2. The F1TENTH course structure: Each autonomy theme provides both the theoretical content and practical learning units with simulations and real hardware. Over the different course modules, the complexity of algorithms is increasing while the depth of understanding also increases.

a hands-on learning part, where the theory of each module is applied in simulation and on the real vehicle hardware. Second, each of the three themes is then completed with a race with the autonomous vehicles. With this setup, this new course ultimately provides learning in various practical topics like mechatronics, control systems, and artificial intelligence.

The hypothesis is that autonomous driving fundamentals must be taught in combination with actual hardware to prepare the students for industry and academic jobs. This combination will enhance the students' *computational thinking* regarding the software and their *systems thinking* regarding the whole autonomous vehicle. This is because the students are allowed for repeated testing and iteration and have the affordance of a physical device to learn as opposed to on-screen simulation only [8]. Furthermore, it is hypothesized that by teaching autonomous driving in a competitive environment called *Autonomous Racing* [9], the motivation and fascination for learning in the field of autonomous vehicles and programming can be kept higher [10]. The idea behind this variation of *competition-based learning* [11] is to have three races in the course that incentivize and motivate the students [12]. In summary, the main contributions of this paper are:

- A detailed overview of a new academic course called F1TENTH that integrates theoretical lectures with hands-on learning modules and competitive racing is given.
- The development of the F1TENTH hardware platform, a modular and realistic 1:10 scale autonomous vehicle for hands-on learning, bridging the gap between toy models and full-scale research vehicles, is presented. Further, we use modular hardware and software to cater to different educational levels, from high school to graduate courses, allowing for scalable and adaptable teaching methods.
- We present survey results on this new course, indicating high student satisfaction and an improved understanding of autonomous driving concepts.
- We make all course materials, hardware setup instructions, and software stacks available as open-source resources, facilitating wider adoption and collaboration among educational institutions.

II. Related Work

Educational robotics: When teaching the fundamentals of autonomous systems and robotics, the goal is to increase the *computational thinking* (CT) of a student, which describes skills developed by students to solve problems in the field of computer science. In 2014, Bers et al. [13] proposed that CT could be increased by leveraging real robot hardware in the classroom because real robots motivate students through gamification. Based on this, the number of courses that leverage the usage of real robots for teaching different robotic subjects at distinct education levels has grown [14]. In a quasi-experimental study with 24 third-grade students, Diago et al. [14] revealed that in contrast to traditional education approaches, using real-world educational robotics created statistically significant gains in computational thinking and computational knowledge. Furthermore, in courses with real-world robots, students' negative attitudes toward heavy mathematical subjects can be improved [15]. Students develop a greater interest in mathematics when they can move beyond theory and apply algorithms to real-world problems. For example, working with a robot demonstrating how optimization algorithms function in practice makes learning more engaging and tangible [15]. Various papers present their ideas on how to teach robotics at undergraduate or graduate levels at various universities [16], [17]. The authors derived that the educational usage of robotics creates high popularity among the students, high collaboration within the teams, and high competition in developing individual solutions for the provided robots. To further increase motivation and engagement within these real-world robot courses, Frank et al. [18] provide the outline of a research project that needs to consist of various elements: (1) The students must build their team that will work with one robot; (2) The team consists of individual roles to manage all tasks; (3) the teams get a budget that can be spent on electronic hardware for the robot; the team must (5) design, (6) build and (7) program the robot on their own; (8) the team must write documents on their developments.

Real-world robots for education: Bakala et al. [19] conducted a systematic review of empirical studies that apply real-world robots for preschoolers. Based on a review of 15 empirical studies, the authors found that commercial robotic kits were mainly used. Unfortunately, in all these kits, only a limited number of input and output interfaces were given, limiting developmental appropriateness to children's cognitive levels. Many studies and evaluations were conducted to get an inside into using robotic hardware kits at the elementary school level [20]. In most of these studies, the bee-bot [21] or Lego Mindstorms kits [22] are used.

Small-scale autonomous and intelligent vehicles: Many universities have real-world research vehicles [23], with which experimental studies and research are conducted. Unfortunately, those vehicles are expensive and difficult to maintain. This issue led to developing small-scale testbeds for connected and automated vehicles and robot swarms [24]. *Duckietown* [25] teaches students to program a small-scale robot in an urban environment. In [26], a scaled RC-Car platform is used to run in a scaled indoor environment, but only a fixed set of hardware and software is provided. The Amazon *DeepRacer* [27] is a small-scale autonomous car that educates students on simulation and reinforcement learning. The most commonly used vehicle is a modified 1:10 scale RC car, and institutions released documentation for hardware and setup on transforming this conventional car into an autonomous racecar. These vehicles are then used either for research or educational purposes and the most prominent ones are the *MIT Racecar* [28], the *MuSHR racecar* [29], the *RoSCAR* [30] and the *F1TENTH* [31] vehicle. The work presented in this paper heavily extends the F1TENTH vehicle in various ways, now deeply exploring its capabilities as an educational platform by providing modular hardware and software and ultimately surveying the students about the course and the vehicle's usefulness.

III. The F1TENTH Course

The state of the art displayed that using real robots provides the chance to create a better understanding of abstract robotics problems [32] and allows exploring autonomous driving solutions closer to real-world application. Unfortunately, a course setup that provides applied knowledge for the whole autonomous driving pipeline, which consists of hardware and software selection, simulation testing, and real-world application, is not available so far. In the following, we display a solution by providing a detailed overview of the new F1TENTH course setup.

A. Prerequisites

The course material is aimed at graduate students but can be reduced to the undergraduate or even high school level. Students enrolling in the F1TENTH course should have basic programming skills in languages such as Python, as much of the coursework involves coding and software development. Proficiency in linear algebra and calculus is crucial for grasping the mathematical aspects of path planning and control theories. A basic understanding of machine learning concepts is not necessary, as the course covers and explains neural network-based perception techniques that cover these topics.

B. Course Philosophy & Learning outcomes

The general course philosophy of the proposed course is "Define the Problem. Implement. Understand" and "Competitions (Races) replace Exams" [33]. The goal is to focus on teaching autonomous systems as hands-on as possible with the provided F1TENTH vehicle, allowing students to enhance their computational and systems thinking.

By grouping 2— 3 students into teams, a diverse set of teams can be created: A mix of majors (only one per team), a mix of programming expertise (Python, C++), a mix of the countries of origin, a mix of genders or ethnic groups. Solving the tasks in teams improves teamwork and collaboration while enhancing social and emotional learning.

The learning outcomes focus on providing in-depth knowledge in the field of autonomous vehicles. The students learn about the theoretical software fundamentals of different autonomy algorithms (perception, planning, and control) and apply them to the autonomous vehicle hardware afterward. The following ten learning outcomes are set up; after the F1TENTH course, the students should be able to

1) understand the current challenges in state of the art for autonomous driving,
2) understand the role of middleware with ROS2 (Robot Operating System 2),
3) understand common sensors for detection and localization,
4) explain vehicle dynamics by visualizing vehicle states,
5) explain the different concepts of path planning,
6) understand the necessity of stabilizing control actions and the responsibilities of the control algorithm,
7) design and tune a path tracking controller,
8) apply software for perception, planning, and control to a 2D and 3D simulation environment,

This article has been accepted for publication in IEEE Transactions on Intelligent Vehicles. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIV.2024.3495227

9) apply software for perception, planning, and control to the F1TENTH hardware, and
10) develop their own software for perception, planning, and control and apply it to the F1TENTH hardware.

C. Content and Syllabus

The F1TENTH course is structured to cover a broad range of topics in autonomous driving, starting from foundational concepts and advancing to complex, real-world applications. The F1TENTH course is split into six modules (Module A-F), which consist of 25 lectures that are displayed in the following list:

Module A: Introduction to F1TENTH, the Simulator & ROS2	
1	Introduction to Autonomous Driving
2	Automatic Emergency Braking
3	Rigid Body Transform
Module B: Reactive Methods	
4	Vehicle States, Vehicle Dynamics and Maps
5	Follow the Wall: First Autonomous Drive
6	Follow the Gap: Obstacle Avoidance
7	Race 1: Preparation
8	Race 1: Single-Vehicle: Obstacle Avoidance
Module C: Mapping & Localization	
9	Scan matching
10	Particle Filter
11	Introduction to Graph-based SLAM
Module D: Planning & Control	
12	Local Planning: RRT, Spline Based Planner
13	Path Tracking Simple: Pure Pursuit
14	Path Tracking Advanced: Model Predictive Control
15	Behavioral Planning: Trustworthy Autonomous Vehicles
Module E: Vision	
16	Classical Perception: Lane Detection
17	Machine Learning Perception: Object Detection
18	Final Project Selection
19	Race 2: Preparation
20	Race 2: Single-Vehicle: High-Speed
Module F: Special Topics and Invited Talks	
21	Ethics for Autonomous Systems
22	Raceline Optimization
23	Special Topic 1: e.g. invited speaker session
24	Special Topic 2: e.g. invited speaker session
25	Special Topic 3: e.g. invited speaker session
Module G: Race 3 And Project Demonstrations	
26	Race 3: Preparation
27	Race 3: Multi-Vehicle Head-to-Head
28	Project Demonstrations

As the hypothesis defines that students need to learn autonomous driving in a hardware-applied and hands-on way, the course is taught in a *clab* style (classroom + lab). There are two modules a week, consisting of a 45-minute lecture and a 2-hour practice session. A seventh module (G) is for the final race and the project demonstration. The general idea of the F1TENTH course is to incrementally increase the depth of knowledge, the difficulty of the algorithms, and the complexity of combining multiple software modules. As depicted in Figure 2, the course starts with teaching single-vehicle behavior only and then moves to more complex vehicle behavior like high-speed driving and multi-vehicle scenarios. Having lab sessions with the real hardware is then providing a concrete learning experience inside the module.

In *Modules A & B*, the students learn the theoretical foundations of autonomous driving. Here, the car is driving at slow speeds. With the primary sensor (LiDAR), the students can perceive the environment and avoid obstacles. In the first race, the goal is to drive a single car around a given track while avoiding obstacles.

In *Modules C & D*, the theoretical foundations of localization (e.g., graph-SLAM), planning (e.g., sampling-based planning), and control (e.g., PiD controller) are explained, and a variety of algorithms are presented. This part is listed as *high-speed autonomy* and involves heavy tuning since both the localization's accuracy and the controller's quality lead to different vehicle behavior. In the second race, the goal is to drive a single car at high speed around a given track.

In *Modules E & F,* the theory of classical and machine learning-based perception techniques are introduced, focusing on lane detection and object detection. The special topic section discusses advanced and interdisciplinary topics in autonomous systems, including ethics for autonomous systems, raceline optimization, and special lectures from industry and research experts. In the final module (G), the students must apply everything they learned throughout the semester in a multi-vehicle race (2 vehicles against each other) and tune the car to drive fast and reliably. Additionally, the results of the projects are presented. The following core components for the course are established:

1) **Theoretical Lectures:** The theoretical fundamentals of the various algorithms in perception, planning, and control are explained in a lecture.
2) **Labs:** Here, the students need to apply the autonomous driving concepts from the lecture to the 2D simulation environment. The labs are explained and discussed in the class, the lab assignments need to be completed outside of class time. The code is evaluated in simulation only.
3) **Races:** The students participate in three autonomous races with their F1TENTH vehicle. The students have to write quality software for the vehicle to be successful in the races. The races help the students improve their risk analysis because they must decide how fast they go with their car to achieve good race results. While the competitive scenario of the races builds up mental toughness for the students, it also creates a way to develop a social community around learning [8].
4) **Special Topics:** A series of special topics with guest lectures that present their applied autonomous driving work from a research or industry perspective are provided. This gives the students some inspiration about state-of-the-art research and industry work.
5) **Final Project:** A final project (or cornerstone project) is set up as an ill-structured software design project with the explicit goal of giving the students the experience of struggle and challenge, which can result in failures and setbacks [34]. These failures are intended to teach the student fundamentals of fault diagnosis [35] and data visualization. Being guided by the teaching assistants ensures the project has a reasonable scope. By demonstrating their project at the end of the semester, the students still achieve a positive result and learning outcome [34].

D. Grading

Since no final exam is held, the final course passing mark will be based on a cumulative score and is composed of the following components and their weighting:

- **40% Labs:** Results of the code submitted in the labs.
- **30% Competition performance:** Results of the races weighted by the race difficulty. 95% of this grade is based on participation, and 5% is based on ranking in the races.
- **20% Final Project:** Quality of the project demonstration and documentation.
- **5% Competition document:** An 5-8 page document summarizing the students' approach to the competition (software architecture, algorithms, hardware, tests, etc); examples of performance results, etc.
- **5% Peer review:** An anonymous evaluation of the student's work performed by their teammates.

E. Feedback & Improvement Phase

After the final grades are given, the students go into a feedback and evaluation phase with the tutors and the professor. Here, the tutors provide a summary of the student's achievements along the course and explain possibilities for further improvement. Feedback is given regarding (1) the code, (2) the algorithms used, (3) the quality of the algorithm in the application, (4) the qualities of a team, and (5) the quality of the documentation. This summary and feedback enable a rethinking of the achievement and further enable the students to learn from their own and others' failed trials.

IV. F1TENTH Vehicle: Modular Hardware

Deploying algorithms on real-world autonomous vehicles is expensive, time-consuming, and dangerous. Especially not many higher education institutions have a real-world autonomous vehicle that can be used for teaching. From an educational point of view, special hardware with car-like characteristics is needed, provides the possibility to integrate different sensor components, and is easy to program and handle. Therefore, we chose a small-scale autonomous vehicle called *F1TENTH* (Figure 3) for this course.

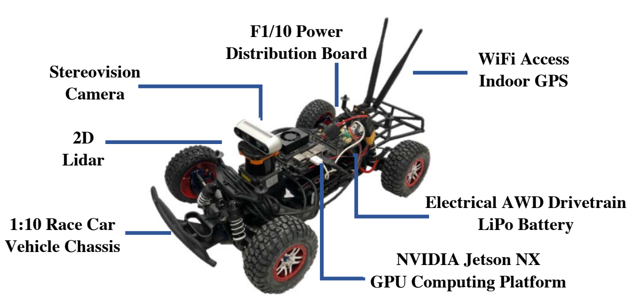


Fig. 3. F1TENTH hardware setup used in the F1TENTH course.

The F1TENTH vehicle is a low-cost, low-effort, low-entry bar 1:10 scale vehicle that enables safe and rapid experimentation and easy student handling. The vehicle is based on a remote-controlled chassis, which includes various components to transform it into an autonomous vehicle. In comparison to other educational robots listed in section II, this small-scale hardware is very close to a real-world vehicle: Ackermann steering; real chassis system with damper and springs; changeable vehicle hardware, e.g., tires; different drivetrain setups, e.g., AWD and RWD; high-speed (max. 60 km/h) and high acceleration (9 m/s^2). All components displayed here closely resemble those used in industrial applications, making work on the car aligned with industry standards, and teaching students these systems explicitly prepares them for their first job after university. The car has an electrical all-wheel drivetrain (AWD) powered by a $5000\,\mathrm{mA\,h}$ lithium polymer battery. A specially developed power distribution board powers all electrical components. The F1TENTH vehicle has a 2D LiDAR and a stereovision camera mounted on the front to perceive its environment. The main computation unit is an NVIDIA embedded GPU computer called *Jetson Xavier NX* with an Ubuntu-based operating system (OS). Besides this, the F1TENTH car does not have only one fixed hardware setup that is used for teaching. A wide variety of hardware components can be integrated. Figure 4 shows a combination of three LiDARs, two mono cameras, three stereo cameras, and three different computation units that can be used on the F1TENTH vehicle.

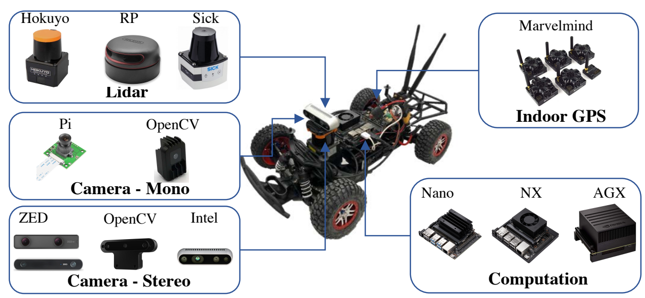


Fig. 4. F1TENTH modular hardware: The vehicle offers the possibility of combining and integrating various hardware components like cameras, LiDAR, or computation systems.

This modular hardware setup of the F1TENTH vehicle provides many advantages:

1) The possibility of switching to a different sensor component is given. For example, the 2D LiDAR sensors offer different sampling rates, field-of-views, and ranges. Changing them on the vehicle impacts the autonomy software, e.g., obtaining distance data of obstacles. Another example is the use of different computing hardware. By running the same software modules on other computation hardware, the students experience slower/faster algorithm calculation times, leading to a slower/faster control frequency and ultimately to worse/better car control. The NVIDIA Jetson computer differs in the overall performance (TOPS, TFLOPS), the number of GPU and CPU cores, RAM memory, and SSD storage [36].
2) Based on the set of heuristics for developing educational robots defined by [37], all 14 defined heuristics are fulfilled. These include a high level of *adaptability*,

TABLE I
OVERVIEW OF F1TENTH HARDWARE MODULES AND THEIR COMBINATION FOR DIFFERENT EDUCATIONAL LEVELS

Educational Level	**Loca -lization**	**Mono camera**		**Stereo camera**			**2D LiDAR**			**Main Computation Unit**		
	Indoor GPS	Raspberry PI	OpenCV Oak-1	Intel Realsense D345i	Zed Mini Zed ZED2	OpenCV Oak-D	Hokuyo 10LX 30LX	Sick	RP A3M1	Nvidia Jetson Nano	Nvidia Jetson NX	Nvidia Jetson AGX
High School		X								X		
University: Undergraduate			X						X		X	
University: Graduate			X	X		X		X			X	
Research and Industry Training	X				X		X					X

the possibility for *collaboration and communication*, the *relevance* of the autonomous driving task, and the list of *challenges* provided for the students throughout the course.

3) The modular hardware design of the F1TENTH vehicle allows for its use across various educational levels, thus broadening its impact. This adaptability means that the F1TENTH can be customized to suit different educational needs: a simplified hardware setup can be used for high school students, while a more complex configuration can be implemented for PhD researchers or industry training (Table I). This versatility ensures that the F1TENTH vehicle is an effective teaching tool for a wide range of learners, from beginners to advanced professionals.

V. F1TENTH VEHICLE: MODULAR SOFTWARE STACK

The hardware establishes the foundational capabilities and limitations of an autonomous vehicle, defining the operational boundaries of the system. However, the core functionalities are developed within the software, enabling the complex decision-making and control processes that replicate and replace human actions in conventional driving scenarios. The complexity increases with each level of automation - more profound knowledge and even more software is required. In an autonomous vehicle, many software components need to be combined; this is usually called a *software stack*. For autonomous vehicles, this software stack consists of the three big modules *perception, planning, and control*, which will enable safe and robust autonomous operation in real-world situations [38].

A. F1TENTH Stack

For the F1TENTH vehicle and the course, a completely new software stack was developed by the authors consisting of several software modules. The modules themselves are based on well-known algorithms but were implemented in Python and ROS2 to work on the F1TENTH vehicle. The software stack consisting of the following software modules is displayed in Figure 5.

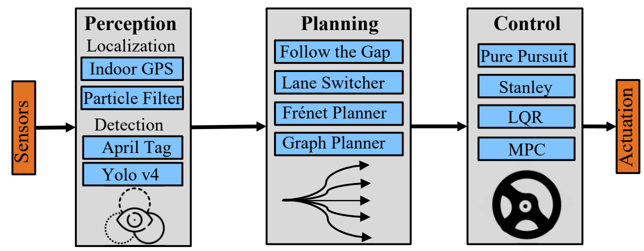


Fig. 5. Software modules for the F1TENTH vehicle

Although many examples of software stacks are given in research papers, the center of attention in this project was to create a *modular* software stack. The goal was to include as many different software modules as possible to offer the students a wide variety of algorithms. First, this allows teaching simpler algorithms at the course's beginning and moving on to more difficult algorithms later. Second, this enables comparing the quality of all algorithms. For example, all algorithms in the control module can track a predefined path, but some algorithms achieve better tracking quality than others. Third, all algorithms have a different need for computation power and need to apply resources on either the GPU or CPU. Fourth, since not all algorithms fit well together, this modularity enables the demonstration of coupled effects between the individual software modules. Figure 6 shows the combination of three algorithms from the stack. The vehicle tracks a reference trajectory (control) based on its current pose (localization) while continuously generating new feasible trajectories (planning) to avoid obstacles.

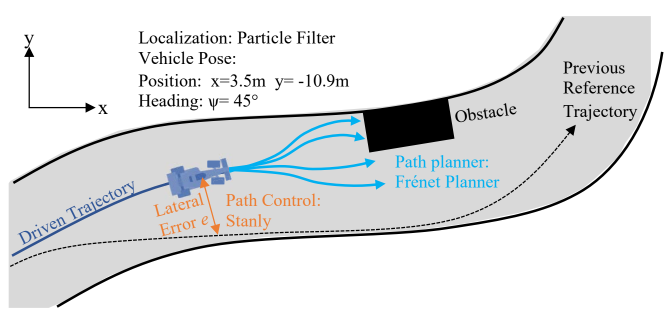


Fig. 6. Example of a combination of different software modules from the F1TENTH software stack. Localization: Particle Filter; Local Path Planning: Frenet Planner; Control: Stanley

The perception modules consist of algorithms from the field of localization and detection. With these algorithms, the autonomous vehicle can find its position and heading (pose) and detect obstacles (e.g. other vehicles) in front of it. The perception modules in the F1TENTH stack consist of the following algorithms.

- **Localization - Indoor GPS:** The car can get its absolute pose states x, y, ψ from and indoor GPS hardware. This hardware is localizing via triangulation and sends an absolute position in a pre-defined area with an accuracy of about 2 cm. In the simulation, the GPS position is provided with absolute ground truth.
- **Localization - Particle Filter:** A particle filer is a localization algorithm that uses a set of random positional samples that update every detection to approximate the car's pose states x, y, ψ . When equipped with a LiDAR, the vehicle can run a particle filter to localize based on LiDAR point cloud detection and a beforehand created map of the environment.
- **Detection - AprilTag:** An AprilTag is a set of 2D barcodes designed to be detected quickly and accurately. With camera calibration, AprilTag detection will provide encoded information and the relative translation and rotation between the camera and the opponent car [39].
- **Detection - YOLO v4 Object Detection:** A simple neural network structure called YOLO [40] is used, which takes in camera images and outputs bounding box detection. Positional information can be calculated based on camera calibration. Students can use a pre-trained neural network or explore their own designs of neural networks to perform object detection

The planning module consists of algorithms that plan a trajectory in front of the vehicle. A trajectory consists of a path (x- and y-position) and a velocity profile. The trajectories need to be collision-free and enable a feasible vehicle behavior. The planning modules in the F1TENTH stack consist of the following algorithms.

- **Simple Planner - Gap Follower:** This algorithm finds gaps in the LiDAR scan by finding the broadest range of scan angles with the highest depth value displayed by [41]. The vehicle plans its motion and steers in the direction to follow the most significant gap to avoid obstacles.
- **Simple Planner - Lane Switcher**. This algorithm creates equispaced lanes that span the entire track and utilize an optimal trajectory [42]. The algorithm switches to a specific lane or back to the optimal trajectory when trying to overtake an opponent.
- **Advanced Planner - Sampling-based:**. This algorithm runs in a frenetic coordinate system and is based on a semi-reactive method by [43]. This planner can select goal coordinates in the Frenet-Frame of the track and generate multiple trajectories to follow an optimal trajectory and avoid obstacles.
- **Advanced Planner - Graph-based:** This algorithm developed by [44] generates a graph covering the track. The nodes in the graph are vehicle poses in the world frame, and the edges of the graph are generated trajectories similar to those in the Frenet Planner. The algorithm then selects appropriate actions for the vehicle from the action set for overtaking and following.

Finally, the control module includes all algorithms that track the desired path and velocity of the planned path. The control modules consist of the following algorithms.

- **Geometric Control - Pure Pursuit:** This algorithm developed by [45] uses a fixed distance look-ahead point on the planned path (reference), a steering angle can be calculated, making the vehicle steer correctly on the path.
- **Geometric Control - Stanley Controller:** This algorithm was displayed in [46] and leverages a PD-controller. Here, the goal is to minimize the heading and cross-track errors (deviation from the reference trajectory). A correction steering angle can be calculated based on both errors.
- **Optimization-based Control - Linear Quadratic Regulator (LQR):** This algorithm was displayed in [47]. The LQR reduces the lateral error from the reference path and optimizes a given cost function. The output is an optimal vehicle speed and steering.
- **Optimization-based Control - Model Predictive Control (MPC):**. The MPC looks at a given receding horizon into the future, predicts the vehicle behavior (vehicle states) for these time steps, and then solves an optimization problem based on constraints [48]. The output is optimal vehicle acceleration and steering.

The modular software of the F1TENTH vehicle enhances its versatility, allowing it to be used for a wider variety of teaching purposes across different educational levels. Based on discussions and interviews with former students and other F1TENTH instructors, four distinct software setups have been identified for different educational levels, as outlined in Table II. This adaptability ensures that the F1TENTH can meet the diverse needs of learners, from high school students to PhD researchers and industry professionals.

VI. Simulation Environments

While simulations are used in R&D to ensure the safety and maturity of the algorithm, in education, it is used to teach the proposed software components in a safe and reliable environment. One source of failures is dismissed by excluding the hardware, focusing on teaching the algorithm fundamentals, and educating the students using software-in-the-loop (SiL) environments. While a variety of simulation environments and platforms for autonomous vehicles exists [49], this course offers two simulation environments.

A. 2D-Simulator

For fast evaluation and testing of the code developed by the students, a 2D simulation environment is provided [31]. Figure 7 shows the process and workflow of the F1TENTH 2D simulator and the related components.

This simulator's advantage is that it is lightweight and runs on all OS (Mac, Linux, Windows). The students can run their developed code directly without any significant changes. The

This article has been accepted for publication in IEEE Transactions on Intelligent Vehicles. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIV.2024.3495227

TABLE II
OVERVIEW OF F1TENTH SOFTWARE MODULES AND THEIR COMBINATION FOR DIFFERENT EDUCATIONAL LEVELS

Educational Level	**Perception**		**Planning**				**Control**			
	Localization	Detection	Follow the Gap	Lane Switcher	Frenet Planner	Graph Planner	Pure Pursuit	Stanley	LQR	MPC
High School	-		X				X			
University: Undergraduate	-		X				X	X		
University: Graduate	GPS Particle Filter	Yolo v4	X	X	X		X	X	X	
Research and Industry Training	GPS Particle Filter	Yolo v4			X	X			X	X

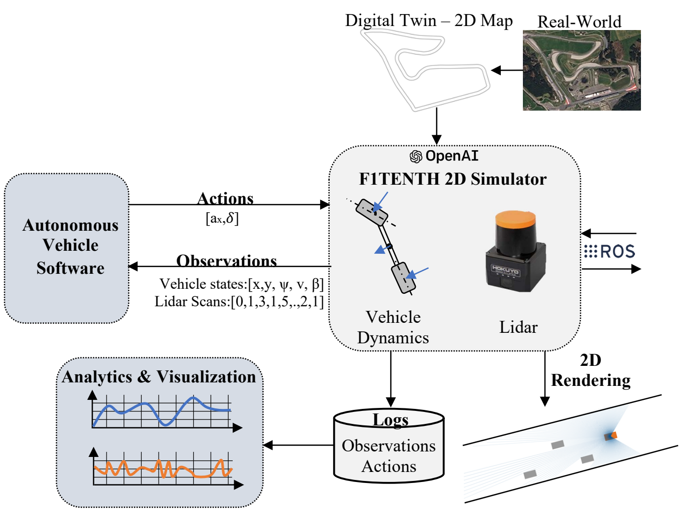


Fig. 7. Process and workflow of the F1TENTH 2D Simulator

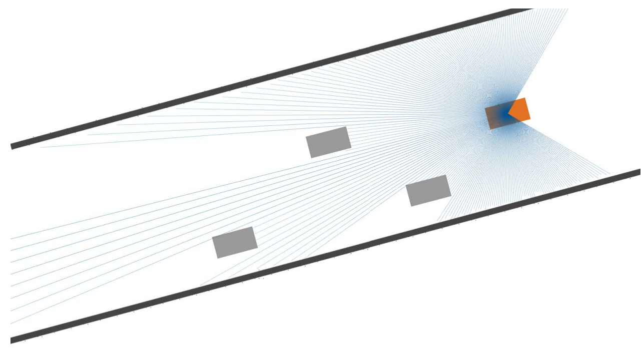


Fig. 8. Exemplary 2D rendering of the F1TENTH Simulator. The ego vehicle is depicted as the orange box, driving in a multi-vehicle (grey boxes) environment. The ego vehicle visualizes its LiDAR stream (blue lines), which detects the obstacles and the walls along the track.

simulation environment is set up in Python code and enables the exchange with ROS2 via an additional bridge.

The 2D simulator allows for different racetracks to be integrated. In the current course setup, over 20 real-world racetracks are provided as a digital twin 2D map for the simulation. The 2D environment is deterministic with realistic vehicle dynamics based on a single-track vehicle dynamics model [50], [51]. This means that the vehicle maneuvers are closer to the physical limits and significant effects like understeering and oversteering are simulated with a linear tire forces approximation. The vehicle dynamics model needs the longitudinal acceleration a_x and the steering angle δ as an action input. The 2D simulator then provides observations on certain vehicle states like position x, y, heading ψ, vehicle velocity v, and the side slip angle β. In addition, collision with the racetrack boundaries and other vehicles is detected automatically, giving the students feedback that their code failed. Additionally, a 2D LiDAR sensor simulation is integrated. This enables simple perception-based algorithms such as object detection (clustering) and localization methods. Figure 8 shows an exemplary 2D rendering of the simulator in a multi-vehicle environment.

The physics engine used in this simulator is faster than real-time simulation and state serialization (loading and saving), making this simulation environment interesting for running experimental evaluations simultaneously. We provide an additional tool that allows the students to visualize the collected data in the simulator. This is necessary to gain insights into vehicle behavior to debug the developed autonomous driving software. Furthermore, this simulator has an Open AI Gym [52] interface, enabling further education in the field of reinforcement learning.

B. 3D-Simulator

While the current 2D F1TENTH simulation environment serves its intended purpose well, it limits the development and testing of algorithms such as camera-based object detection, 3D LiDAR-SLAM, or vision-based end-to-end neural networks. A multitude of 3D-capable simulation environments exist in the robotics and automotive world to enable such capabilities, allowing for the integration of new sensors and a virtual representation of the test environment. One example of such a simulator is CARLA [53] for the entire development pipeline of full-scale autonomous vehicle algorithms. The Donkey Simulator [54] is a simulation environment specifically developed for a similarly sized robotics platform as F1TENTH called Donkey, while Flightmare [55], developed by Song et al., offers a realistic and good-performing simulation environment for quadrotors. Currently, AutoDrive [56] is the 3D simulator (Figure 9) that is heavily used for the F1TENTH courses. In contrast to the previous environments,

This article has been accepted for publication in IEEE Transactions on Intelligent Vehicles. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIV.2024.3495227

already has the F1TENTH vehicle included in its framework and does not require custom vehicle modeling, adjustments of the rendering environment, or the implementation of custom dynamic models.

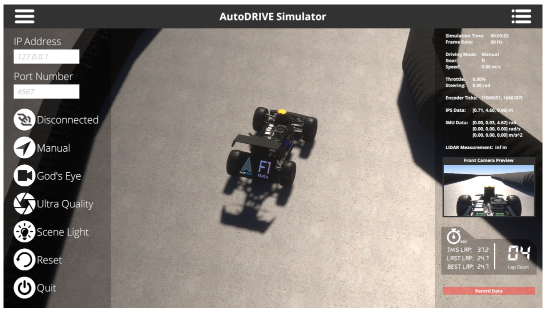


Fig. 9. Example 3D rendering in the AutoDrive [56] simulator.

VII. Course Survey

For this paper, a formal assessment was conducted in terms of a student survey at the end of the semester from spring 2022 to spring 2024. The survey was handed out to the students after finishing all mandatory work but before the final course grade was given. The course survey was done at five different universities, reaching 94 students in total. The survey was anonymous so no demographic data (gender, ethnicity) was collected this time. To exclude the bias created by various instructors, teaching styles, and university setups, the focus of the survey questions primarily assesses the usage of the vehicle and the theme of ”racing” in the course [57]. To condense and structure the survey results, four research questions are defined that are answered with the help of the survey outcomes.

A. Q1: Does the course cover the necessary content to teach autonomous driving?

First, the students are surveyed regarding the course content and if, from the student's perspective, the topic of autonomous driving is covered holistically in the course. Survey questions and results related to this research question are displayed in Figure 10.

The results show that more than half of the students strongly agree that the learning outcomes were communicated clearly and that the complicated topics were displayed understandably. Although it is evident that this course is teaching the complete pipeline of autonomous driving, we see here potential to make the material more apparent to the students. Exactly 73.4% of the surveyed students strongly believe they now have a more fundamental understanding of autonomous driving technology. The answer supports the feedback that 63.8% of the students can reproduce the most important concepts (perception, planning, control) from the field of autonomous driving. This indicates that the course content contains all essential aspects of teaching autonomous driving.

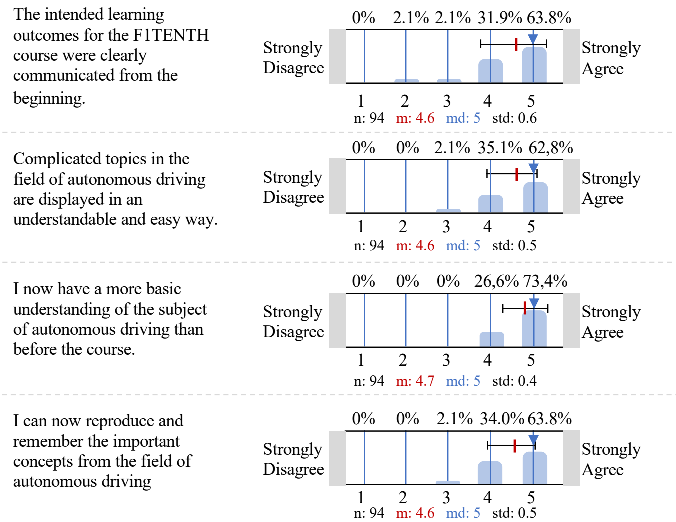


Fig. 10. Survey Results: These questions are related to the content of the F1TENTH lecture. The students give feedback on the overall quality of the lecture content. (n: number of answers, m: mean value, md: median value, std: standard deviation)

B. Q2: Is the F1TENTH hardware the right tool to teach autonomous driving hands-on?

Second, the students are surveyed regarding the usage of the F1TENTH hardware. The goal is to get feedback on whether the small-scale vehicle is a good support for teaching autonomous driving-related topics and if it helps the students learn content in this area. Survey questions and results related to this research question are displayed in Figure 11 With a mean value of 4.7, the students strongly agree that combining theory and real hardware leads to a better learning outcome. Additionally, 79.3% of the students strongly agree that the F1TENTH vehicle is a good educational artifact. This reveals that the proposed F1TENTH hardware has a high educational value for learning about autonomous driving. Additionally, as general feedback, the students answered the question, ”What did you most like about the course?” with the following written answers: Applying the code to real hardware; working with the car; Cars and Hardware; Working with Hardware; the hands-on work and the competitive spirit of the course.

C. Q3: Is the aspect of “racing” a good theme and concept for an educational course?

Third, the students are surveyed regarding the course's racing theme and their thoughts on competing in three different races with the F1TENTH vehicle. Survey questions and results related to this research question are displayed in Figure 12.

The goal was to identify if the students felt that both the topic of racing and the three races helped them stay motivated and active in the course. With an average answer of 4.7 and 4.5 to these questions, the students indicated that they strongly agree that the racing setup helps them to stay motivated. We conclude that the students acknowledged the general course philosophy ”Competitions replace Exams” in a way that they liked to come to the lecture and stayed motivated

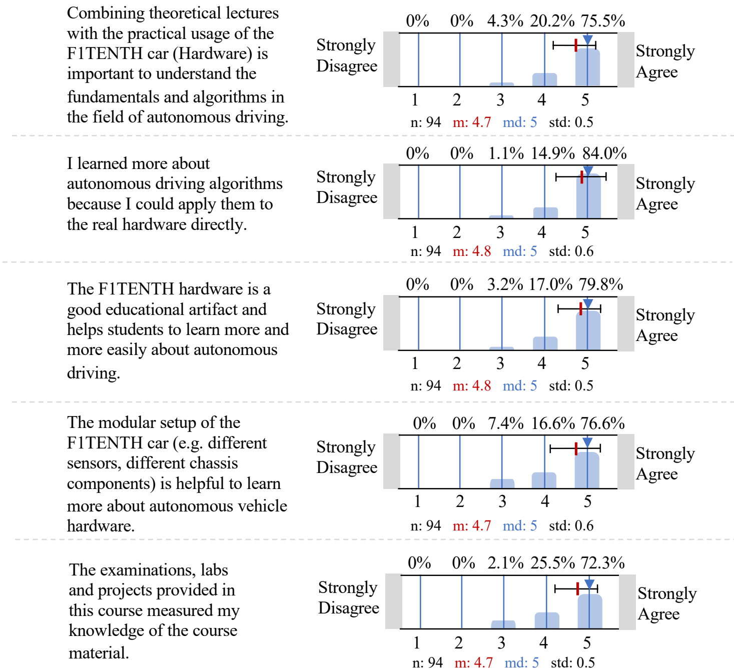


Fig. 11. Survey Results: These questions are related to the usage of the F1TENTH Hardware in the course. The students give feedback on the usage of the hardware and if they think this vehicle helped them to learn more about the topic. (n: number of answers, m: mean value, md: median value, std: standard deviation)

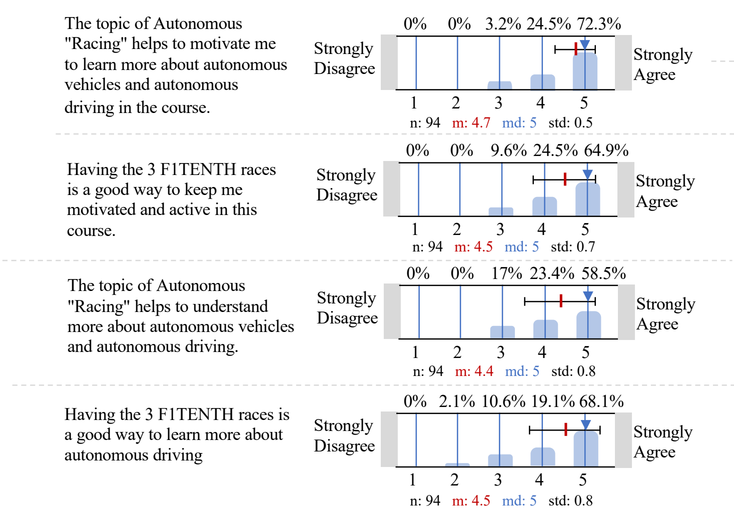


Fig. 12. Survey Results: These questions are related to the racing theme of the F1TENTH course and the three competitions throughout the semester. (n: number of answers, m: mean value, md: median value, std: standard deviation)

throughout the whole semester. In addition, we wanted to know if the students feel that both the topic of racing and the three races help them understand and learn more about autonomous driving. These questions were received with an average of 4.4 and 4.5 — the lowest score in the survey. Only 58.5% of the students strongly agree that the topic of racing, albeit fun and motivating, has added value in helping them learn the subject matter. The general observation was that the racing tracks and rules often lead to more complex vehicle behaviors that the lectures may not clearly explain. Also, high speeds and high accelerations, which are often needed for winning the races, usually favor simpler algorithms in the reactive paradigm rather than more advanced planning algorithms. Winning a race also calls for extensive trial-and-error testing, which is a niche application/aspect often ignored by most college courses and can be regarded as repetitive without additional educational value.

D. Q4: Is the F1TENTH course helpful for the students' career paths?

In the final questionnaires, the students are asked whether the course was relevant to their future career plans and if this course would help them get more involved in autonomous driving. Survey questions and results related to this research question are displayed in Figure 13.

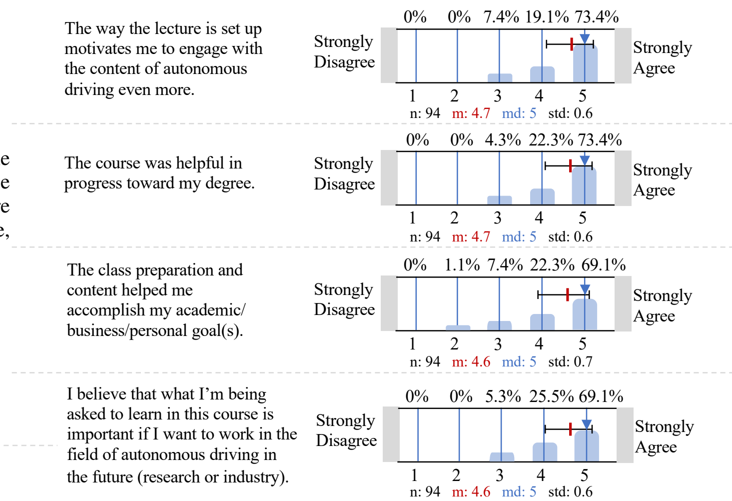


Fig. 13. Survey Results: These questions are related to the career perspectives of the students. (n: number of answers, m: mean value, md: median value, std: standard deviation)

73.4% of the students indicated that this course was helpful in their progress toward their degree. Additionally, 73.4% of the students strongly agree that the F1TENTH lectures motivated them to engage more with the topic of autonomous driving. Additionally, 69.1% of the students strongly agree that the course teaches them important content if they want to work in the field of autonomous driving in the future. It can be concluded that the course setup provides the proper range to teach future students the essential topics in the field of autonomous driving. Furthermore, the course content provides all the necessary know-how to prepare the students for their first job after the university - either in research or academia.

VIII. DISCUSSION & IMPROVEMENTS

A. Course Design - Overall

The goal of the F1TENTH course is to teach autonomous systems more hands-on and to combine the theory of perception, planning, and control with the application of the learned content on an actual autonomous vehicle. The F1TENTH car fulfills these requirements by offering a 1:10 small-scale platform to the students. By keeping the hardware and software close to industry practices, the course creates a sense of authenticity among learners and supports them better to engage with questions regarding software and hardware development for autonomous systems. The survey results confirm the hypothesis that the F1TENTH vehicle is an excellent educational artifact and helps students to learn more and more efficiently about autonomous driving.

The student's feedback in the survey indicated that the course provides the right content to teach autonomous driving. More than 80% of the students strongly agree that the hardware platform and lab modules greatly motivate their learning, and more than 70% of the students strongly agree that the hardware enhanced their understanding of the subjects. Based on this finding, it can be concluded that teaching autonomous driving with hands-on application is helpful in supporting student learning.

B. Course Design - Competition

In contrast, survey results on racing competitions show that, although more than 80% of the students strongly agree that the competitions motivate them, only 50% strongly agree that the competitions enhance their learning outcomes. Therefore the topic of racing is an excellent way to motivate the students throughout the semester, but it has no additional educational value. However, we believe that after learning the fundamentals from the course, the students can properly connect the advanced concepts and optimize the different modules within the races. They can make a vehicle project superior to others, which can be an excellent way to judge students' understanding through some healthy competition. Therefore, we conclude that new theme formats with the vehicle need to be explored for an improvement of the course, for example, a *cargo delivery* theme or a *valet parking* theme.

C. Hardware

Both the displayed hardware and the software stack provide a highly modular setup to teach content in the field of autonomous driving. This modularity and variety of hardware and software were not provided by any other course yet. This setup allows the teacher to teach various autonomous driving content in perception, planning, and control. As a significant advantage, the modular hardware will enable teachers to teach autonomous systems at different educational levels. The F1TENTH course design carefully considers the optimization of resources to balance cost and functionality. While the current estimated cost of the F1TENTH vehicle is around 3500 USD, this pricing reflects a strategic selection of components that provide the necessary performance for educational and research purposes without excessive expenditure.

On the downside, currently, the F1TENTH hardware has high costs, which means that not all universities or schools can afford ten or more cars for their students. In the future, the aim is to reduce the costs of vehicle hardware to create a more economical solution that can be taken up by a broader variety of schools, not just the resourceful ones.

D. IV research and application

Since all course material, hardware setup instructions, and the software stack are made open-source, the adaptation rate of the F1TENTH course by various universities is much easier. The co-authors of this paper are all instructors of a version of the F1TENTH course at their home universities and adapted the usage of the F1TENTH vehicle as part of their teaching. Additionally, regular international competitions are offered so students from different universities can meet each other at conferences and compete with their F1TENTH vehicles.

Furthermore, the F1TENTH hardware is currently used in various research projects to evaluate the next generation of algorithms for autonomous and intelligent vehicles. As a low-cost, low-risk platform, the F1TENTH vehicle is ideal for performing research in challenging settings that would be dangerous with full-scale vehicles, such as high-speed off-road driving [58]. The usage of multiple F1TENTH platforms also makes it particularly conducive to multi-agent controls research, where it is feasible to demonstrate state-of-the-art distributed controller synthesis on multiple F1TENTH cars [59] or doing IV-related research [60]–[62].

IX. Summary

This paper presents a new teaching course and hardware platform for hands-on teaching of autonomous systems. The article describes the course syllabus and teaching modules, which aim to teach autonomous driving fundamentals hands-on. From basic reactive methods to advanced planning algorithms, the teaching labs enhance students' computational and systems thinking through autonomous driving with the F1TENTH vehicle. The hardware setup of the F1TENTH vehicle and the software stack for the vehicle and course are explained in detail. Both modular setups allow teaching autonomous driving on different educational levels: simple hardware and algorithms in the beginning and more complex hardware and algorithms in the end. With this design, teaching the theoretical fundamentals and applying them to real-world hardware is possible.

Furthermore, all work presented in this paper is made available open-source and interested students, teachers and researchers can access the F1TENTH course material on openEdx, F1TENTH hardware build, F1TENTH 2D Simulator and the F1TENTH software stack online.

References

[1] O. Pribyl and M. Lom, "Impact of autonomous vehicles in cities: User perception," in *2019 Smart City Symposium Prague (SCSP)*. IEEE, May 2019.

[2] A. Hayes, "Self-driving cars could change the auto industry (gm, f)," Aug 2020. [Online]. Available: https://www.investopedia.com/articles/personal-finance/031315/selfdriving-cars-could-change-auto-industry.asp

[3] Z. Wadud, D. MacKenzie, and P. Leiby, "Help or hindrance? the travel, energy and carbon impacts of highly automated vehicles," *Transportation Research Part A: Policy and Practice*, vol. 86, pp. 1–18, Apr. 2016.

[4] S. D. Pendleton, H. Andersen, X. Du, X. Shen, M. Meghjani, Y. H. Eng, D. Rus, and M. H. Ang, "Perception, planning, control, and coordination for autonomous vehicles," *Machines*, vol. 5, no. 1, 2017.

[5] M. Chowdhury and K. Dey, "Intelligent transportation systems-a frontier for breaking boundaries of traditional academic engineering disciplines [education]," *IEEE Intelligent Transportation Systems Magazine*, vol. 8, no. 1, pp. 4–8, 2016.

[6] J. Tang, L. Shaoshan, S. Pei, S. Zuckerman, L. Chen, W. Shi, and J.-L. Gaudiot, "Teaching autonomous driving using a modular and integrated approach," in *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*. IEEE, Jul. 2018. [Online]. Available: https://doi.org/10.1109/compsac.2018.00057

[7] M. Shibata, K. Demura, S. Hirai, and A. Matsumoto, "Comparative study of robotics curricula," *IEEE Transactions on Education*, vol. 64, no. 3, pp. 283–291, Aug. 2021.

[8] S. Papert, *Mindstorms children, computers, and powerful ideas*. London, England: Basic Books, 1980.

[9] J. Betz, H. Zheng, A. Liniger, U. Rosolia, P. Karle, M. Behl, V. Krovi, and R. Mangharam, "Autonomous vehicles on the edge: A survey on autonomous vehicle racing," 2022.

[10] R. P. Medeiros, G. L. Ramalho, and T. P. Falcao, "A systematic literature review on teaching and learning introductory programming in higher education," *IEEE Transactions on Education*, vol. 62, no. 2, pp. 77–90, May 2019. [Online]. Available: https://doi.org/10.1109/te.2018.2864133

[11] J. C. Burguillo, "Using game theory and competition-based learning to stimulate student motivation and performance," *Computers &Education*, vol. 55, no. 2, pp. 566–575, Sep. 2010.

[12] J. Shim, D. Kwon, and W. Lee, "The effects of a robot game environment on computer programming education for elementary school students," *IEEE Transactions on Education*, vol. 60, no. 2, pp. 164–172, May 2017. [Online]. Available: https://doi.org/10.1109/te.2016.2622227

[13] M. U. Bers, L. Flannery, E. R. Kazakoff, and A. Sullivan, "Computational thinking and tinkering: Exploration of an early childhood robotics curriculum," *Comput. Educ.*, vol. 72, pp. 145–157, Mar. 2014.

[14] P. D. Diago, J. A. González-Calero, and D. F. Yáñez, "Exploring the development of mental rotation and computational skills in elementary students through educational robotics," *Int. J. Child Comput. Interact.*, no. 100388, p. 100388, Aug. 2021.

[15] S. Kucuk and B. Sisman, "Students' attitudes towards robotics and STEM: Differences based on gender and robotics experience," *Int. J. Child Comput. Interact.*, vol. 23-24, no. 100167, p. 100167, Jun. 2020.

[16] G. Cielniak, N. Bellotto, and T. Duckett, "Integrating mobile robotics and vision with undergraduate computer science," *IEEE trans. educ.*, vol. 56, no. 1, pp. 48–53, Feb. 2013.

[17] J. M. Esposito, "The state of robotics education: Proposed goals for positively transforming robotics education at postsecondary institutions," *IEEE Robot. Autom. Mag.*, vol. 24, no. 3, pp. 157–164, Sep. 2017.

[18] D. Frank, K. Kolotka, A. Phillips, M. Schulz, C. Rigney, A. Drown, R. Stricko, K. Harper, and R. Freuler, "Developing and improving a multi-element first-year engineering cornerstone autonomous robotics design project," in *2017 ASEE Annual Conference & Exposition Proceedings*. ASEE Conferences, 2018.

[19] E. Bakala, A. Gerosa, J. P. Hourcade, and G. Tejera, "Preschool children, robots, and computational thinking: A systematic review," *Int. J. Child Comput. Interact.*, vol. 29, no. 100337, p. 100337, Sep. 2021.

[20] S. E. Jung and E.-s. Won, "Systematic review of research trends in robotics education for young children," *Sustainability*, vol. 10, no. 4, 2018. [Online]. Available: https://www.mdpi.com/2071-1050/10/4/905

[21] M. C. Di Lieto, E. Inguaggiato, E. Castro, F. Cecchi, G. Cioni, M. Dell'Omo, C. Laschi, C. Pecini, G. Santerini, G. Sgandurra, and P. Dario, "Educational robotics intervention on executive functions in preschool children: A pilot study," *Comput. Human Behav.*, vol. 71, pp. 16–23, Jun. 2017.

[22] A. Savard and V. Freiman, "Investigating complexity to assess student learning from a robotics-based task," *Digit. Exp. Math. Educ.*, vol. 2, no. 2, pp. 93–114, Sep. 2016.

[23] G. Mehr, P. Ghorai, C. Zhang, A. Nayak, D. Patel, S. Sivashangaran, and A. Eskandarian, "X-car: An experimental vehicle platform for connected autonomy research," *IEEE Intelligent Transportation Systems Magazine*, vol. 15, no. 2, pp. 41–57, 2023.

[24] A. Mokhtarian, J. Xu, P. Scheffe, M. Kloock, S. Schäfer, H. Bang, V.-A. Le, S. Ulhas, J. Betz, S. Wilson, S. Berman, L. Paull, A. Prorok, and B. Alrifaee, "A survey on small-scale testbeds for connected and automated vehicles and robot swarms," 2024.

[25] J. Tani, L. Paull, M. T. Zuber, D. Rus, J. How, J. Leonard, and A. Censi, "Duckietown: An innovative way to teach autonomy," in *Educational Robotics in the Makers Era*. Springer International Publishing, 2017, pp. 104–121.

[26] W. Feng, Y. Pei, B. Hai, W. Huang, X. Gong, and F. Zhu, "Autonomous RC-car for education purpose in iSTEM projects," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Jun. 2018.

[27] B. Balaji, S. Mallya, S. Genc, S. Gupta, L. Dirac, V. Khare, G. Roy, T. Sun, Y. Tao, B. Townsend, E. Calleja, S. Muralidhara, and D. Karuppasamy, "DeepRacer: Autonomous racing platform for experimentation with sim2real reinforcement learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2020.

[28] S. Karaman, A. Anders, M. Boulet, J. Connor, K. Gregson, W. Guerra, O. Guldner, M. Mohamoud, B. Plancher, R. Shin, and J. Vivilecchia, "Project-based, collaborative, algorithmic robotics for high school students: Programming self-driving race cars at MIT," in *2017 IEEE Integrated STEM Education Conference (ISEC)*. IEEE, 2017.

[29] S. S. Srinivasa, P. Lancaster, J. Michalove, M. Schmittle, C. Summers, M. Rockett, J. R. Smith, S. Choudhury, C. Mavrogiannis, and F. Sadeghi, "Mushr: A low-cost, open-source robotic racecar for education and research," 2019.

[30] K. Hart, C. Montella, G. Petitpas, D. Schweisinger, A. Shariati, B. Sourbeer, T. Trephan, and J. Spletzer, "RoSCAR," in *Proceedings of the 2014 workshop on Mobile augmented reality and robotic technology-based systems - MARS '14*. ACM Press, 2014.

[31] M. O'Kelly, V. Sukhil, H. Abbas, J. Harkins, C. Kao, Y. V. Pant, R. Mangharam, D. Agarwal, M. Behl, P. Burgio, and M. Bertogna, "F1/10: An open-source autonomous cyber-physical platform," 2019.

[32] S. Anwar, N. A. Bascou, M. Menekse, and A. Kardgar, "A systematic review of studies on educational robotics," *Journal of Pre-College Engineering Education Research (J-PEER)*, vol. 9, no. 2, Jul. 2019.

[33] P. Smith, "Bridging the gap in engineering education through student competition," in *2009 ICCAS-SICE*, 2009, pp. 1877–1877.

[34] C. C. W. Hulls and C. Rennick, "Use of a cornerstone project to teach ill-structured software design in first year," *IEEE Transactions on Education*, vol. 63, no. 2, pp. 98–107, May 2020.

[35] J. M. G. de Gabriel, A. Mandow, J. Fernandez-Lozano, and A. Garcia-Cerezo, "Mobile robot lab project to introduce engineering students to fault diagnosis in mechatronic systems," *IEEE Transactions on Education*, vol. 58, no. 3, pp. 187–193, Aug. 2015.

[36] Connect Tech, "Jetson module comparison," Aug 2021. [Online]. Available: https://connecttech.com/jetson/jetson-module-comparison/

[37] C. Giang, A. Piatti, and F. Mondada, "Heuristics for the development and evaluation of educational robotics systems," *IEEE Transactions on Education*, vol. 62, no. 4, pp. 278–287, Nov. 2019.

[38] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun, "Towards fully autonomous driving: Systems and algorithms," in *2011 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Jun. 2011.

[39] L. Yu, M. Li, and G. Pan, "Indoor localization based on fusion of apriltag and adaptive monte carlo," in *2021 IEEE 5th Information Technology,Networking,Electronic and Automation Control Conference (ITNEC)*, vol. 5, 2021, pp. 464–468.

[40] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[41] V. Sezer and M. Gokasan, "A novel obstacle avoidance algorithm:"follow the gap method"," *Robotics and Autonomous Systems*, vol. 60, no. 9, pp. 1123–1134, 2012.

[42] A. Heilmeier, A. Wischnewski, L. Hermansdorfer, J. Betz, M. Lienkamp, and B. Lohmann, "Minimum curvature trajectory planning and control for an autonomous race car," *Vehicle System Dynamics*, vol. 58, no. 10, pp. 1497–1527, Jun. 2019.

[43] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a frenet frame," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, May 2010. [Online]. Available: https://doi.org/10.1109/robot.2010.5509799

[44] T. Stahl, A. Wischnewski, J. Betz, and M. Lienkamp, "Multilayer graph-based trajectory planning for race vehicles in dynamic scenarios," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, Oct. 2019.

[45] R. C. Coulter, "Implementation of the pure pursuit path tracking algorithm," Carnegie-Mellon UNIV Pittsburgh PA Robotics INST, Tech. Rep., 1992.

[46] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann *et al.*, "Stanley: The robot that won the darpa grand challenge," *Journal of field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[47] J. Chen, W. Zhan, and M. Tomizuka, "Autonomous driving motion planning with constrained iterative lqr," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 2, pp. 244–254, 2019.

[48] A. Katriniok, J. P. Maschuw, F. Christen, L. Eckstein, and D. Abel, "Optimal vehicle dynamics control for combined longitudinal and lateral autonomous vehicle guidance," in *2013 European Control Conference (ECC)*. IEEE, Jul. 2013.

[49] G. Yang, Y. Xue, L. Meng, P. Wang, Y. Shi, Q. Yang, and Q. Dong, "Survey on autonomous vehicle simulation platforms," in *2021 8th International Conference on Dependable Systems and Their Applications (DSA)*. IEEE, Aug. 2021.

[50] M. Althoff, M. Koschi, and S. Manzinger, "CommonRoad: Composable benchmarks for motion planning on roads," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Jun. 2017.

[51] T. Zhang, Y. Sun, Y. Wang, B. Li, Y. Tian, and F.-Y. Wang, "A survey of vehicle dynamics modeling methods for autonomous racing: Theoretical models, physical/virtual platforms, and perspectives," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 3, pp. 4312–4334, 2024.

[52] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[53] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

[54] "Donkeycar: a python self driving library," https://github.com/autorope/donkeycar, 2017, [Online; accessed 01-August-2024].

[55] Y. Song, S. Naji, E. Kaufmann, A. Loquercio, and D. Scaramuzza, "Flightmare: A flexible quadrotor simulator," in *Conference on Robot Learning*, 2020.

[56] T. V. Samak, C. V. Samak, and M. Xie, "Autodrive simulator: A simulator for scaled autonomous vehicle research and education," in *2021 2nd International Conference on Control, Robotics and Intelligent System*, ser. CCRIS'21. New York, NY, USA: Association for Computing Machinery, 2021, p. 1–5.

[57] B. Li, T. Xu, X. Li, Y. Cui, X. Bian, S. Teng, S. Ma, L. Fan, Y. Tian, and F.-Y. Wang, "Integrating large language models and metaverse in autonomous racing: An education-oriented perspective," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 59–64, 2024.

[58] H. Karnan, K. S. Sikand, P. Atreya, S. Rabiee, X. Xiao, G. Warnell, P. Stone, and J. Biswas, " VI-IKD: High-Speed Accurate Off-Road Navigation using Learned Visual-Inertial Inverse Kinodynamics ," arXiv Preprint arXiv:2203.15983, 2022.

[59] J. Wei, T. Li, S. Chaudhuri, I. Dillig, and J. Biswas, "Onevision: Centralized to distributed controller synthesis with delay compensation," in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, 2021, pp. 398–405.

[60] X. Sun, M. Zhou, Z. Zhuang, S. Yang, J. Betz, and R. Mangharam, "A benchmark comparison of imitation learning-based control policies for autonomous racing," in *2023 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2023, pp. 1–5.

[61] B. Li, L. Fan, Y. Ouyang, S. Tang, X. Wang, D. Cao, and F.-Y. Wang, "Online competition of trajectory planning for automated parking: Benchmarks, achievements, learned lessons, and future perspectives," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 16–21, 2023.

[62] B. D. Evans, R. Trumpp, M. Caccamo, F. Jahncke, J. Betz, H. W. Jordaan, and H. A. Engelbrecht, "Unifying f1tenth autonomous racing: Survey, methods and benchmarks," 2024.

Johannes Betz is an assistant professor in the Department of Mobility Systems Engineering at the Technical University of Munich (TUM), where he is leading the Autonomous Vehicle Systems (AVS) lab. He is one of the founders of the TUM Autonomous Motorsport team. His research focuses on developing adaptive dynamic path planning and control algorithms, decision-making algorithms that work under high uncertainty in multi-agent environments, and validating the algorithms on real-world robotic systems. Johannes earned a B.Eng. (2011) from the University of Applied Science Coburg, a M.Sc. (2012) from the University of Bayreuth, an M.A. (2021) in philosophy from TUM, and a Ph.D. (2019) from TUM.

Hongrui Zheng received his B.S. degrees in Mechanical Engineering and Computer Science from Georgia Institute of Technology, and his M.S. degree in Robotics from University of Pennsylvania. He is currently a Ph.D. candidate at the University of Pennsylvania where he is working at the xLab for Safe Autonomous Systems. His research focuses on building the tools and theoretical foundations necessary to scale design, testing, and optimization of safe-autonomous systems.

Felix Jahncke received his bachelor's degree in Mechanical Engineering and master's degree in Automotive Engineering from the Technical University of Munich. During his master's studies, he completed research visits at the Technical University of Delft (Netherlands) and the University of Pennsylvania (USA). Since 2023, he has been pursuing his Ph.D. at TUM, focusing on the transition from traditional software architectures towards end-to-end learning methods, using F1TENTH vehicles.

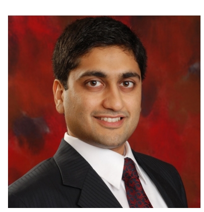
Zirui Zang received his B.S. degree in Optical Engineering from University of Rochester and his M.S. degree in Robotics from University of Pennsylvania. He is currently a Ph.D. candidate at the University of Pennsylvania where he is working at the xLab for Safe Autonomous Systems. His research focuses on multi-agent interactions between autonomous vehicles.

Florian Sauerbeck is a PhD student at the Institute of Automotive Technology at the Technical University of Munich (TUM). He received both his B.Sc. degree in 2017 and his M.Sc. degree in 2019 from (TUM). His main research interests include localization, object detection, and sensor fusion-related applications for autonomous driving with a focus on unstructured environments.

Yahong Rosa Zheng received the B.S. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 1987, and the M.S. degree from Tsinghua University, Beijing, China, in 1989, both in electrical engineering. She received the Ph.D. degree from the Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada, in 2002. Currently she is a full professor a Lehigh University, her research interests include array signal processing, wireless communications, and wireless sensor network.

Joydeep Biswas' earned his PhD in Robotics from Carnegie Mellon University in 2014, and his B.Tech in Engineering Physics from the Indian Institute of Technology Bombay in 2008. He was an Assistant Professor in the College of Information and Computer Sciences at University of Massachusetts Amherst. Currently Joydeep is an Assistant Professor at UT Austin.

Venkat Krovi is the Michelin Endowed Chair of Vehicle Automation in the Departments of Automotive Engineering and Mechanical Engineering at Clemson University. He received his Ph.D. degree in Mechanical Engineering and Applied Mechanics from the University of Pennsylvania in 1998.



Rahul Mangharam received his Ph.D. in Electrical and Computer Engineering from Carnegie Mellon University, where he also received his MS and BS. He is a Full Professor in the Department of Electrical and Systems Engineering at the University of Pennsylvania. He is a founding member of the PRECISE Center and directs the xLab for Safe Autonomous Systems Lab at Penn. His research is at the intersection of formal methods, machine learning and control for medical devices, energy-efficient buildings, and autonomous systems.