Cavalier Autonomous Racing

University of Virginia
Charlottesville, USA.
Team principal: Prof. Madhur Behl (madhur.behl@virginia.edu)

1 About the team

Located at the University of Virginia, the Cavalier Autonomous Racing team is a mix of faculty and students driven by the mission of building the fastest fully autonomous racing car. We believe that autonomous racing is the next grand challenge for safe self-driving vehicles. Motorsport racing has always been the proving grounds for new automotive technologies; and autonomous 'battle of algorithms' racing will play the same role for self-driving software and hardware. In motorsport racing there is a saying that "If everything seems under control, then you are not going fast enough" - we are building an autonomous racing AI with this as the objective function. The team is led by Prof. Madhur Behl who is the founder, co-organizer, and the race director for the widely popular F1/10 International Autonomous Racing Competitions (running since 2015).

2 Team members and organization

The Cavalier Autonomous Racing Club at UVA is mentored by the PI, Prof.Madhur Behl, who has expertise in developing autonomous vehicles, and conducts research on autonomous systems at the UVA Cyber-Physical Systems Link Lab. Autonomous racing is the ultimate engineering challenge. Students from a variety of engineering, and non-engineering disciplines are members of the club. Autonomous racing is exposing students to the foundations of perception, planning and control in a fun, and challenging environment. We are train the next generation of autonomous systems researchers and experts.

The organization is divided into the following sub-teams. Each team has a team-leader and 5-8 members:

1. Varundev Sureshbabu: Overall Student Lead and Control Team Lead.
2. Sandesh Banskota: Sensor Integration Lead
3. Mert Banskota: Simulation, and Software Integration.
4. Trent Weiss: Deep Learning, Motion Planning, and Simulation.
5. Aron Harder: Head of Perception
6. Kenneth Brown: Chassis Design and Mechatronics
7. Scott Steever: Head of Operations

Dr. Madhur Behl is a professor in the departments of Computer Science, and Systems and Information Engineering, and a member of the Cyber-Physical Systems Link Lab at the University of Virginia. He conducts research at the confluence of Machine Learning, Predictive Control, and Artificial Intelligence with applications in Cyber-Physical Systems, Autonomous Systems, Robotics, and Smart Cities. Examples include: fully autonomous racing at the limits of control (Agile Autonomy), safety of autonomous vehicles (Safe Autonomy), data predictive control for flooding in coastal cities, and AI for building energy optimization. He is the associate editor for the SAE Journal on Connected and Autonomous Vehicles. He is the co-founder, organizer, and the race director of the F1/10 International Autonomous Racing Competitions. He received his Ph.D. (2015) and M.S. (2012), in Electrical and Systems Engineering, both from the University of Pennsylvania. Dr. Behl is the winner of the American Control Conference (ACC 2017) Best Energy Systems Paper Award, TECHCON Best Paper Award (2015), and the best demo award at BuildSys, 2012. He is also the recipient of the Richard K. Dentel Memorial Prize awarded by the University of Pennsylvania for research and excellence in urban transportation.

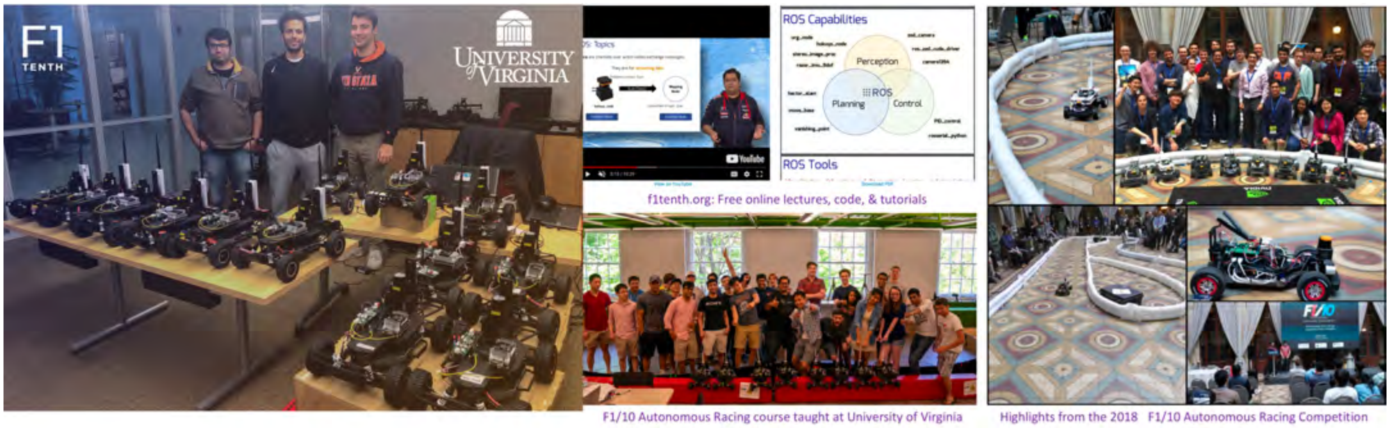

Figure 1: We have access to more than 16 F1/10 autonomous racecars in our lab and a configurable indoor track. The F1/10 testbed instrument is enabling K-12, undergrad, and graduate outreach through our online courses and MOOCs, autonomous racing competitions, summer schools, and hackathaons.

3 Autonomous Racing Experience

The team principal, Prof Madhur Behl, has an excellent track record of running a large scale autonomous racing project. He has pioneered a novel effort called F1/10 Autonomous Racing. He is the lead organizer, founder, and the race director for the F1/10 International Autonomous Racing Competitions being held at premier cyber-physical systems, robotics, machine learning, and embedded systems conference venues worldwide for the past four years. This ‘battle of algorithms’ is rapidly becoming the proving grounds for pitting AI systems for self-driving cars against each other, while steadily advancing the state of the art in perception, planning, and control. F1/10 is one of the fastest growing autonomous competitions in the world. In the last three iterations of the competition, we have grown from 5 to more than 20 teams all over the world who have built the F1/10 cars. Participants at F1/10 span a wide range of backgrounds, from academic, and industrial researchers, to undergraduate, graduate students, and postdocs. Participants are experts in control systems, robotics, embedded systems, machine learning, and software development.

The team has designed educational materials and a 1/10 autonomous vehicle test-bed, and the broader research community is extremely excited about using this infrastructure for educational purposes. What we have done is essentially to design a program analogous to the FIRST LEGO League, but for autonomous vehicles. The course material developed by him for F1/10 is opensource. It has been used by more than a dozen universities around the world to build their own versions of the 1/10 scale autonomous cars and has been adapted for teaching cyber-physical systems and autonomous systems courses at multiple universities. Due to the experience and expertise of running an international level autonomous racing completions for 1/10 scale vehicles, the team is highly confident that they can be very successful in the next step - moving away from 1/10 scale vehicles to a university wide program to develop a full scale autonomous Indy light racecar.

Cavalier Autonomous Racing: Video Highlights

- World’s first high speed fully autonomous racing overtake. [F1/10]: https://youtu.be/Dw_Fg_JLcNg
- 3rd F1/10 Autonomous Racing Competition 2018 - Torino, Italy: https://youtu.be/VlE2Wb_XhoQ
- 2nd F1/10 Autonomous Racing Competition 2018 - Porto, Portugal: https://youtu.be/ZwRGtrXYgmI
- F1/10 Undergraduate Course at UVA [Spring 2019]: https://youtu.be/RpEVCgt18P4
- F1/10 Undergraduate Course at The University of Virginia [Spring 2018]: https://youtu.be/ZQg61UNbr7Q
- UVA LinkLab F1/10 Autonomous Race Car on ESPN! https://youtu.be/Rp8aUOytpno

4 Autonomous Racing Research

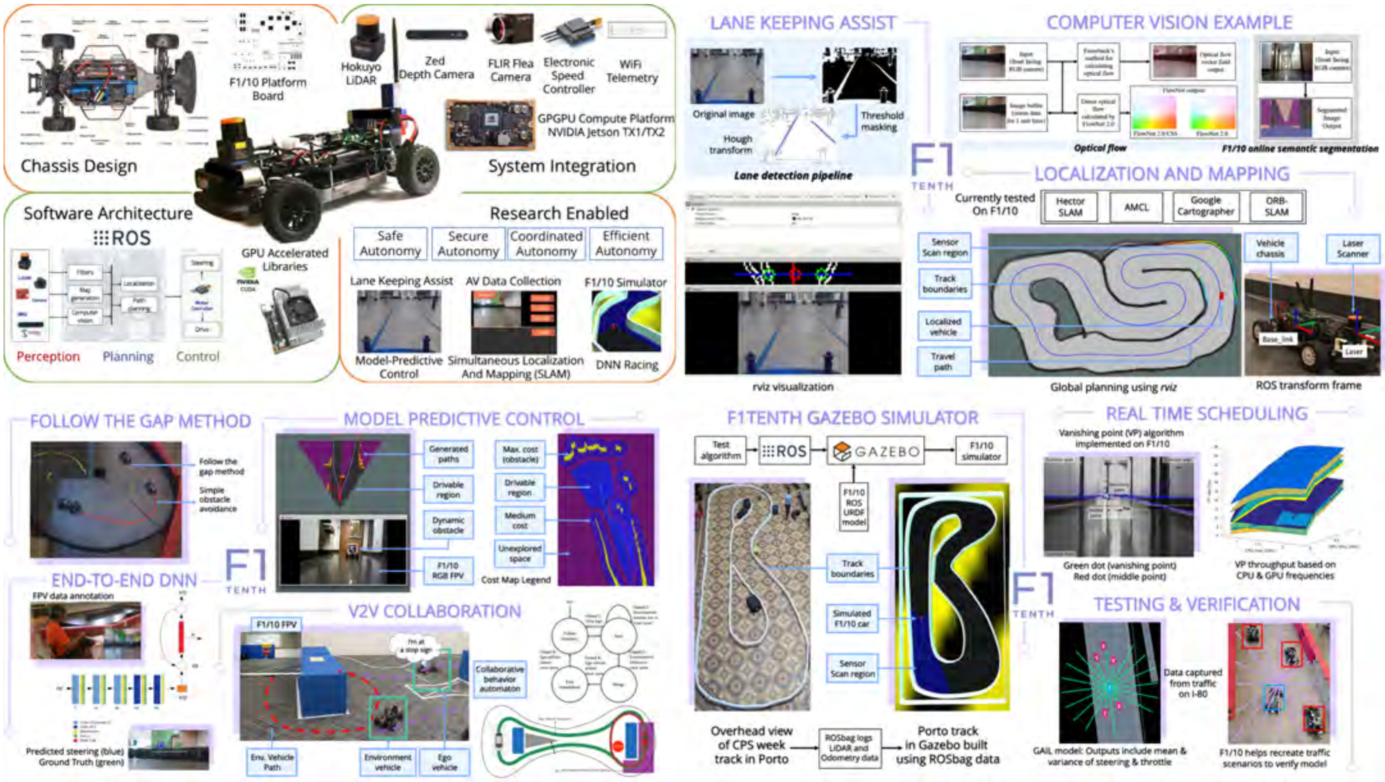

Figure 2: Overview of research approaches for autonomous racing published by the team in recent years.

4.1 Research: Planning and Control

The decision making systems utilized on AVs have progressed significantly in recent years; however they still remain a key challenge in enabling AV deployment [1]. While AVs today can perform well in simple scenarios such as highway driving; they often struggle in scenarios such as merges, pedestrian crossings, roundabouts, and unprotected left-turns. Conducting research in difficult scenarios using full-size vehicles is both expensive and risky. In this section we highlight the research on algorithms for obstacle avoidance, end-to-end driving, model predictive control conducted by our group.

4.2 Obstacle avoidance

Obstacle avoidance and forward collision assist are essential to the operation of an autonomous vehicle. The AV is required to scan the environment for obstacles and safely navigate around them. For this reason, many researchers have developed interesting real-time approaches for avoiding unexpected static and dynamic obstacles [2, 3]. To showcase the capability of the F1/10 testbed, we implement one such algorithm known as *Follow The Gap* (FTG) method [4]. The Follow the Gap method is based on the construction of a gap array around the vehicle and calculation of the best heading angle for moving the robot into the center of the maximum gap in front, while simultaneously considering its goal. These two objectives are considered simultaneously by using a fusing function. The three steps involved in FTG are:
(a) Calculating the gap array using vector field histogram, and finding the maximum gap in the LIDAR point cloud using an efficient sorting algorithm,
(b) Calculating the center of the largest gap, and
(c) Calculating the heading angle to the centre of the largest gap in reference to the orientation of the car, and generating a steering control value for the car.

4.3 End-to-end driving

Some recent research replaces the classic chain of perception, planning, and control with a neural network that directly maps sensor input to control output [5, 6, 7], a methodology known as end-to-end driving. Despite the early interest in end-to-end driving [8], most self-driving cars still use the perception-planning-control paradigm. This slow development can be explained by the challenges of verifying system performance; however, new approaches based on reinforcement learning are being actively developed [9].

The F1/10 testbed is a well suited candidate for experimentation with end-to-end driving pipelines, from data gathering and annotation, to inference, and in some cases even training.

Partly inspired by Pilotnet [5] end-to-end work, we implemented a combination of a LSTM [10] and a Convolutional Neural Network(CNN) [11] cell. These units are then used in the form of a recurrent neural network (RNN). This setup uses the benefits of LSTMs in maintaining temporal information (critical to driving) and utilizes the ability of CNN's to extract high level features from images.

To evaluate the performance of the model we use the normalized root mean square error (NRMSE) metric between the ground truth steering value and the predicted value from the DNN. As can be seen in the point-of-view (PoV) image in Figure **??**[Left], our DNN is able to accurately predict the steering angle with an NRMSE of 0.14.

4.4 Path planning

AVs operate in relatively structured environments. Most scenarios an AV might face feature some static structure. Often this is the road geometry, lane connectivity, locations of traffic signals, buildings, etc. Many AVs exploit the static nature of these elements to increase their robustness to sensing errors or uncertainty. In the context of F1/10, it may be convenient to exploit some information known *a priori* about the environment, such as the track layout and floor friction. These approaches are called *static*, or *global*, and they typically imply building a map of the track, simulating the car in the map, and computing offline a suitable nominal path which the vehicle will attempt to follow. Valuable data related to friction and drift may also be collected to refine the vehicle dynamics model. More refined models can be adopted off-line to compute optimal paths and target vehicle speeds, adopting more precise optimization routines that have a higher computational complexity to minimize the lap time.

Once the desired global path has been defined, the online planner must track it. To do that, there are two main activities must be accomplished on-line, namely *localization* and *vehicle dynamics control*. Once the vehicle has been properly localized within a map, a local planner is adopted to send longitudinal and transversal control signals to follow the precomputed optimal path. As the local planner needs to run in real-time, simpler controllers are adopted to decrease the control latency as much as possible. Convenient online controllers include pure pursuit path geometric tracking [12]. The F1/10 software distribution includes an implementation of pure pursuit, nodes for creating and loading waypoints, and path visualization tools. For the interested reader we recommend this comprehensive survey of classical planning methods employed on AVs [13].

4.5 Model Predictive Control

While data annotation for training end-to-end networks is relatively easy, the performance of such methods is difficult to validate empirically [14] especially relative to approaches which decompose functionality into interpret-able modules. In this section we outline both a local planner which utilizes a model predictive controller (MPC) and a learned approximation of the policy it generates detailing one way planning components can be replaced with efficient learned modules.

Components: The F1/10 platform includes a MPC written in C++ comprised of the vehicle dynamics model, an optimization routine which performs gradient descent on the spline parameters. Peripheral support nodes provide an interface to road center line information, a multi-threaded goal sampler, a 2D occupancy grid, and a trajectory evaluation module. Additionally, we include a CUDA implementation of a learned approximation of the MPC which utilizes the same interface as described above.

Cubic Spline Trajectory Generation: One local planner available on the F1/10 vehicle utilizes the methods outlined in [15] and [16] and first described in [17]. This approach is commonly known as *state-lattice planning with cubic spline trajectory generation*. Each execution of the planner requires the current state of the vehicle and a goal state. Planning occurs in a local coordinate frame. The vehicle state x is defined in the local coordinate system, a subscript indicates a particular kind of state (i.e. a goal) In this implementation we define x as: $\vec{x} = [s_x \ s_y \ v \ \Psi \ \kappa]^T$, where s_x and s_y are the x and y positions of the center of mass, v is the velocity, Ψ is the heading angle, and κ is the curvature.

In this formulation, trajectories are limited to a specific class of parameterized curves known as *cubic splines* which are dense in the robot workspace. We represent a cubic spline as a function of arc length such that the parameters $\vec{p} = [s \ a \ b \ c \ d]^T$ where s is the total curve length and (a, b, c, d) are equispaced knot points representing the curvature at a particular arc length. When these parameters are used to define the expression of $\kappa(s)$ which can be used to steer the vehicle directly. The local planner's objective is then to find a *feasible trajectory* from the initial state defined by the tuple $\vec{x}$ to a goal pose $\vec{x}_g$.

We use a gradient descent algorithm and forward simulation models which limit the ego-vehicle curvature presented in [16]. These methods ensure that the path generated is kinematically and dynamically feasible up to a specified velocity.

4.6 Simultaneous Localization and Mapping (SLAM)

The ability for a robot to create a map of a new environment without knowing its precise location (SLAM) is a primary enabler for the use of the F1/10 platform in a variety of locations and environments. Moreover, although SLAM is a well understood problem it is still challenging to create reliable real-time implementations. In order to allow the vehicle to drive in most indoor environments we provide interface to a state of the art LIDAR-based SLAM package which provides loop-closures, namely Google Cartographer [18]. Included in our base software distribution are local and global settings which we have observed to work well empirically through many trials in the classroom and at outreach events. In addition we include a description of the robots geometry in an appropriate format which enables plug-and-play operation. For researchers interested primarily in new approaches to SLAM the F1/10 platform is of interest due to its non-trivial dynamics, modern sensor payload, and the ability to test performance of the algorithm in motion capture spaces (due to the small size of vehicle).

In addition to SLAM packages we also provide an interface to an efficient, parallel localization package which utilizes a GPU implementation of raymarching to simulate the observations of random particles in a known 2D map [19]. The inclusion of this package enables research on driving at the limits of control even without a motion capture system for state estimation.

4.7 Computer Vision

Our distribution of F1/10 software includes the basic ingredients necessary to explore the use of deep learning for computer vision. It includes CUDA enabled versions of PyTorch [20], Tensorflow [21], and Darknet [22]. We include example networks for semantic segmentation [23], object detection [24], and optical flow [25]; we focus on efficient variants of the state-of-the-art that can run at greater than 10 FPS on the TX2. Recently, it has come to light that many DNNs used on vision tasks are susceptible to so called *adversarial examples*, subtle perturbations of a few pixels which to the human eye are meaningless but when processed by a DNN result in gross errors in classification. Recent work has suggested that such adversarial examples are *not* invariant to viewpoint transformations [26], and hence *not* a concern. The F1/10 platform can help to enable principled investigations into how errors in DNN vision systems affect vehicle level performance.

4.8 Adaptive Pure-Pursuit for Racing

Using pure-pursuit for autonomous racing has a unique set of challenges. Our work focuses on the problem of designing an adaptive lookahead pure-pursuit controller for an autonomous racecar to optimize racing metrics such as lap time, average lap speed, and deviation from a reference trajectory. The challenge is to do so without mathematically modeling the dynamics of the racecar; but instead using a ROS based simulator. We propose a greedy lookahead algorithm to compute and assign optimal lookahead distances for the pure-pursuit controller for each waypoint on a reference trajectory for improving the lap time. We also present a new ROS F1/10 autonomous racing simulator which is open-source and well suited to address challenges in autonomous racing. We use the simulator to design and evaluate the adaptive pure-pursuit algorithm and compare our method with seminal pure-pursuit controllers. Finally, we demonstrate our approach on a real testbed using a F1/10 autonomous racecar. Our method results in a significant improvement ($> 50\%$ in simulation, and $> 20\%$ on the real F1/10 testbed) in the lap times of the autonomous race car compared to the baseline of Ackermann-adjusted pure-pursuit. In our research we have demonstrated that adaptive lookahead pure-pursuit out performs seminal pure-pursuit and Ackermann-steering adjusted pure-pursuit in terms of race related metrics such as lap time, lap distance and average lap speed and is a novel fit for autonomous racing, both in simulation and the F1/10 testbed. The analysis focuses on the single agent setting, where a single race car is tasked with following a reference trajectory (often the race-line), with the minimum lap time. We have also contributed an F1/10 autonomous racing simulator that allows the community to experiment on racing related challenges, including incorporating reinforcement learning, and multi-agent settings. Our future work also involves using the adaptive lookahead pure-pursuit for multiple autonomous racecars & creating a formal framework for autonomous overtaking in high speeds and close-proximty situations.

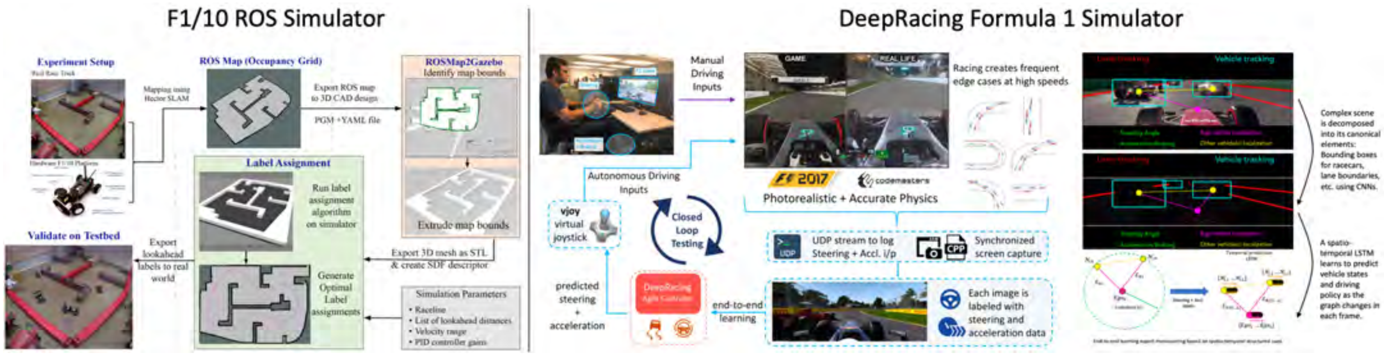

Figure 3: Our group has developed the ROS F1/10 Autonomous Racing simulator, and also the Deep Racing Forumla 1 racing simulator.

5 Racing simulation research

ROS F1/10 Racing Simulator Our ROS and Gazebo based racing simulator is designed to replicate the behaviour of the F1/10 autonomous racecar. The simulator was developed by the authors as a teaching tool for the autonomous racing course offered at the University of Virginia and beyond. The simulator is also used as a verification tool to test the performance of racing algorithms before they are deployed on the physical platform, and is the primary tool used for testing the algorithm proposed by this paper Some features of the simulator include:

- High precision odometry and state estimation (including position, velocity and acceleration)
- Compatible with most major ROS control, SLAM and navigation packages out-of-the-box
- Ability to spawn multiple independent autonomous racecars for dynamic and head-to-head racing

The simulator is freely available onlineand supports several algorithms out of the box (e.g. Wall Following, Follow The Gap [27], Hector SLAM [28], AMCL Particle Filter [29], and TEB Planner [30]). Further details of the simulator design and operation are included in the accompanying supplementary video file.

The race-track start/finish line is located between turns 1 and 9, which is also the location of the *map* and *odom* frames in the simulator. A ROS node is designed for measuring the state of the racecar at all waypoints on the racetrack including instantaneous velocity, and deviation from the reference trajectory.

Autonomous Formula One Racing Simulator We are developing a virtual Formula One (F1) simulator that uses the F1 2019 racing game released by Codemastersc©as the underlying simulation environment. This is the first time the immensely popular and photo-realistic F1 game has been used as a platform for training autonomous race cars. Due to its realism the F1 game was the first game to be used in the Formula One eSports Series, which debuted in 2017. There is also evidence to suggest that real-life F1drivers use this game for virtual practicing.

F1™ 2019 advertises a telemetry stream of information about the driver's current behavior over a UDP socket in a "fire and forget" type broadcast. Each data point in the stream, contains information such as steering angle, brake pressure, throttle pressure, velocity, acceleration, the position; velocity; and acceleration of the other cars, lap-time etc. We developed a custom software wrapper for grabbing screenshots of the driver's perspective in the F1 game and automatically annotating them with ground-truth values of the game's state from the UDP telemetry stream. A single dedicated process spawns two threads: (i) For capturing screenshots of the drivers point of view, which we call the "screen-capture thread", and (ii) For listening for telemetry data from the game on a UDP socket, which we call the "telemetry thread" Each thread has a copy of a shared CPU timer. The screen-capture thread uses a C++11 API, built on top of Microsoft's DirectX, for capturing images of the driver's point of view and tags each image with a timestamp.

Finally, we can also close-the-loop by autonomously driving the F1 car in the game by injecting controls via a virtual joystick API built on top of vJoy. **This is the first time that the F1 Codemasters game is being used as a simulator for autonomous racing.**

Perrone Robotics Test Track – 10 mins from lab   Virginia International Raceway – 1 hour from the lab

Figure 4: The team has access to several full scale test track facilities.

6 Competition Plan

6.1 Round 2: Demonstration

Our plan is to show the capability of the team by demonstrating head to head racing using our F1/10 autonomous racing testbeds. In addition, we are in the process of preparing a fully autonomous electric Go-Kart. We will use this platform to also demonstrate the team's autonomous capabilities.

6.2 Collaboration Plan

A racecar is never a finished product - it is always a prototype. By supporting and collaborating with the Cavalier Autonomous Racing team, you will be supporting students who will master technologies that are going to shape the future of autonomous vehicles. The Cavalier Autonomous Racing team will facilitate a wide range of research, education, and training in autonomy. Approximately 30-50 students per year will be impacted by this program. We welcome support, collaborations, and sponsorship from the industry, foundations, universities, and other organizations in several forms - access to full scale self-driving car platform, hardware (new sensors) and software (simulation, deep learning etc.), testing and research facilities, and student support.

6.3 Round 3: Simulation Race

Our team is very strong in this area. We have developed the widely popular ROS F1/10 Autonomous Racing Simulator being used by dozens of research groups all over the world. In addition, our group is the first in the world to use the widely popular Formula 1 game as a simulation testbed for our DeepRacing project.

6.4 Access to test facilities.

The team has access to two test track facilities. The first is a closed-circuit private track at Perrone Robotics, one of our collaborators. The other is the Virginia International Raceway. These will be useful during the final stages of the competition in case we need to run the Indy light vehicle at full tracks.

7 Funding and Sponsorship

The Cavalier Autonomous Racing Club has already secured \$100K in sponsorship - From the Jefferson Trust Foundation, and industry collaborators.

Cavalier Autonomous Racing program will attract a lot of opportunities for sustained operation and funding. To help with this process we will take advantage of websites, social media, and other methods to promote this program. Team principal Behl has good working relationships with several industry partners including Perrone Robotics, MITRE, Leidos, Amazon, Toyota Research, Mecredes AMG, and Nvidia who could potentially sponsor this program. Furthermore, the program itself is closely related to the PIs research agenda and this will allow opportunities to seek funding from federal sources such as NSF, DARPA, DOT, and ONR in the future. Lastly, it is also possible to receive support in form of hardware donations/gifts; e.g. a LIDAR manufacturer could give us a their prototype for testing, receive cloud computing/powerful workstation hardware from Nvidia, and camera based sensors from Intel's Realsense group.

8 Videos and Websites relevant to this work:

1. DeepRacing AI: https://deepracing.ai/
2. ROS F1/10 Autonomous Racing Simulator: https://f1tenth.dev/
3. UVA F1/10 Autonomous Racing: https://linklab-uva.github.io/autonomousracing/
4. Autonomous Racing Docker Simulator: https://hub.docker.com/r/madhurbehl/f1tenth
5. F1/10 Main Website: http://f1tenth.org/
6. F1/10 Autonomous Racing research is featured on CBS 19 News: https://www.cbs19news.com/content/news/New-UVA-Cyber-Lab-Developing--Self-Thinking--Cars-474884243.html
7. UVA Today Cover Story: https://news.virginia.edu/content/teaching-cars-think-autonomous-future
8. Coverage of our work on Trust in Autonomous vehicles: https://morningconsult.com/2018/04/05/americans-less-trusting-self-driving-safety-following-high-profile-accidents/
9. Research highlight in Mashable: https://mashable.com/2018/04/12/self-driving-cars-safety-autonomous/#_JS9lyOXkOqJ
10. Radio interview regarding agile autonomy and trust in autonomous vehicles: https://wina.com/podcasts/the-future-of-autonomous-vehicles-madur-behl/
11. ROS F1/10 Autonomous Racing Simulator: https://youtu.be/IXxNsMLHdeo
12. 3rd F1/10 Autonomous Racing Competition 2018 - Torino, Italy: https://youtu.be/VlE2Wb_XhoQ
13. F1/10 Undergraduate Course at UVA [Spring 2019]: https://youtu.be/RpEVCgt18P4
14. World's first high speed fully autonomous racing overtake. [F1/10]: https://youtu.be/Dw_Fg_JLcNg
15. UVA LinkLab F1/10 Autonomous Race Car on ESPN: https://youtu.be/Rp8aUOytpno
16. F1/10 Autonomous Race Car Assembly [Timelapse]: https://youtu.be/Pq1WVEsdNXI
17. ROSCon '19 Talk: ROS F1/10 Autonomous Racing Simulator: https://youtu.be/tZeA7ykIYwA
18. F1/10 Undergraduate Course at The University of Virginia [Spring 2018]: https://youtu.be/ZQg61UNbr7Q
19. 2nd F1/10 Autonomous Racing Competition 2018 - Porto, Portugal: https://youtu.be/ZwRGtrXYgmI

References

[1] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. On a formal model of safe and scalable self-driving cars. *arXiv preprint arXiv:1708.06374*, 2017.

[2] Rahul Tallamraju, Sujit Rajappa, Michael Black, Kamalakar Karlapalem, and Aamir Ahmad. Decentralized mpc based obstacle avoidance for multi-robot target tracking scenarios. *arXiv preprint arXiv:1805.09633*, 2018.

[3] Massimiliano Iacono and Antonio Sgorbissa. Path following and obstacle avoidance for an autonomous uav using a depth camera. *Robotics and Autonomous Systems*, 106:38–46, 2018.

[4] Volkan Sezer and Metin Gokasan. A novel obstacle avoidance algorithm: "follow the gap method". *Robotics and Autonomous Systems*, 60(9):1123 – 1134, 2012.

[5] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars. *CoRR*, abs/1604.07316, 2016.

[6] Lu Chi and Yadong Mu. Deep steering: Learning end-to-end driving model from spatial and temporal visual cues. *arXiv preprint arXiv:1708.03798*, 2017.

[7] Hesham M Eraqi, Mohamed N Moustafa, and Jens Honer. End-to-end deep learning for steering autonomous vehicles considering temporal dependencies. *arXiv preprint arXiv:1710.03804*, 2017.

[8] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In *Advances in neural information processing systems*, pages 305–313, 1989.

[9] Alex Kendall, Jeffrey Hawke, David Janz, Przemyslaw Mazur, Daniele Reda, John-Mark Allen, Vinh-Dieu Lam, Alex Bewley, and Amar Shah. Learning to drive in a day. *arXiv preprint arXiv:1807.00412*, 2018.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[11] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term m'emory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[12] R Craig Coulter. Implementation of the pure pursuit path tracking algorithm. Technical report, Carnegie-Mellon UNIV Pittsburgh PA Robotics INST, 1992.

[13] B. Paden, S. Z. Yong, D. Yershov, and E. Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on Intelligent Vehicles*, 1(1):33–55, March 2016.

[14] Shai Shalev-Shwartz and Amnon Shashua. On the sample complexity of end-to-end training vs. semantic abstraction training. *arXiv preprint arXiv:1604.06915*, 2016.

[15] Matthew McNaughton. *Parallel Algorithms for Real-time Motion Planning*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, July 2011.

[16] Thomas Howard. *Adaptive Model-Predictive Motion Planning for Navigation in Complex Environments*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 2009.

[17] Bryan Nagy and Alonzo Kelly. Trajectory generation for car-like robots using cubic curvature polynomials. *Field and Service Robots*, 11, 2001.

[18] Wolfgang Hess, Damon Kohler, Holger Rapp, and Daniel Andor. Real-time loop closure in 2d lidar slam. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 1271–1278. IEEE, 2016.

[19] Corey Walsh and Sertac Karaman. Cddt: Fast approximate 2d ray casting for accelerated localization. abs/1705.01167, 2017.

[20] Adam Paszke, Sam Gross, Soumith Chintala, and Gregory Chanan. Pytorch, 2017.

[21] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[22] Joseph Redmon. Darknet: Open source neural networks in c. *h ttp://pjreddie. com/darknet*, 2016, 2013.

[23] Sachin Mehta, Mohammad Rastegari, Anat Caspi, Linda G. Shapiro, and Hannaneh Hajishirzi. Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. *CoRR*, abs/1803.06815, 2018.

[24] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[25] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE conference on computer vision and pattern recognition (CVPR)*, volume 2, page 6, 2017.

[26] Jiajun Lu, Hussein Sibai, Evan Fabry, and David Forsyth. No need to worry about adversarial examples in object detection in autonomous vehicles. *arXiv preprint arXiv:1707.03501*, 2017.

[27] Volkan Sezer and Metin Gokasan. A novel obstacle avoidance algorithm:"follow the gap method". *Robotics and Autonomous Systems*, 60(9):1123–1134, 2012.

[28] Stefan Kohlbrecher, Johannes Meyer, Thorsten Graber, Karen Petersen, Uwe Klingauf, and Oskar von Stryk. Hector open source modules for autonomous mapping and navigation with rescue robots. In *Robot Soccer World Cup*, pages 624–631. Springer, 2013.

[29] Daniel Hennes, Daniel Claes, Wim Meeussen, and Karl Tuyls. Multi-robot collision avoidance with localization uncertainty. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 147–154. International Foundation for Autonomous Agents and Multiagent Systems, 2012.

[30] C Rösmann. teb local planner ros package, 2015.