Indy Autonomous Challenge Proposal

MIT Driverless: Autonomous Racing Team*
Massachusetts Institute of Technology
Cambridge, MA 02139
`driverless.mit.edu`

1 Introduction

High speed motorsports such as IndyCar, Nascar, and Formula 1 have been critical in pushing the limits of automotive technology to develop higher-performing, more efficient, and - most importantly - safer products for the public. The technologies developed by these competitions have a heavy influence on the design of hardware and software used by consumer vehicles, and in the end not only save the public money, but also save lives. It is competition series like these which can be credited for things as simple as the seat-belt [9] to more complex technologies like the turbocharger [7].

At MIT Driverless, we see the Indy Autonomous Challenge as the next step in bringing the products of high speed motorsports to the world. This competition will bring together the brightest minds from industry and academia to develop quick, agile, and safe autonomous vehicles. Just as it is with the driver-based competitions, this cutting edge technology will push the industry forward by influencing the design of autonomous vehicles on public roads, making them more affordable and safer.

There are many areas for which MIT Driverless sees benefits for the public coming from the competition series, the three most important of which will be:

- Teams will be required to develop high-speed multi-agent planning algorithms to navigate around their opponents. This will be critical for avoiding collisions and maneuvering to safety on highways and which will result in countless saved lives.
- Deploying, testing, and tuning low-compute perception systems will allow the use of lower cost processing units on vehicles, thereby saving consumers' money. This will also allow more complex perception algorithms to be deployed on vehicles sold to the public.
- The competition presents a unique opportunity to develop controls and planning algorithms to exploit the unconstrained nature of the track. Most deployed autonomous vehicles currently rely on proper lane segmentation and thus perform poorly when put on dirt roads or poorly marked highways. Developing algorithms which take advantage of this will allow a wider range of consumers to purchase and use autonomous vehicles in their daily lives.

It is for these reasons among many others that MIT Driverless is motivated to compete in the Indy Autonomous Challenge. We see this as a unique opportunity to push the limits of current autonomous vehicle technology and are excited to be part of this one-of-a-kind competition here in North America.

2 Team History

MIT Driverless was founded out of a small office at MIT with the goal of providing engineering students from all around Massachusetts the skills they needed to succeed as roboticists. In 2018, we partnered with TU-Delft, a strong Mechanical Engineering university, to compete in the Formula Student Driverless Series in Europe as we believed the challenges brought by the competition would provide countless opportunities for our team members. In twelve months, we developed a full-stack autonomous racing platform from scratch and were awarded podium finishes at all competitions we attended against numerous multi-year teams from around the world. With this success, MIT Driverless is now the hub of practical autonomy at MIT and attracts the brightest minds from Boston to come work with, contribute to, and learn from the group.

*MIT Driverless autonomous racing team is affiliated with the MIT Edgerton Center.

Figure 1: Our team finishing on the podium in the biggest European competition *FSG 2019*

MIT's history in the world of autonomy is not limited to the success of our team, as it is also home to some of the most influential robotics researchers of the 20th century and many of the most successful groups to compete in the Darpa Challenges. Examples include Rus Tedrake - the MIT lead professor for the 2015 Darpa Challenge team, John Leonard - a key contributor to the development of SLAM, Sertac Karaman - the co-founder of optimus ride, Luca Carlone - the lead professor of the subterranean challenge, and Daniela Rus - an accomplished researcher in field of multiagent planning. If there is any place on earth to develop algorithms for autonomous vehicles, it is here at MIT.

3 Team Structure

Because our group was set up to provide engineers of varying backgrounds the opportunity to learn the practical skills they need to succeed as roboticists, we have developed an incredibly diverse team. Having engineers specializing in many different fields has been extremely valuable for bringing new ideas to the autonomous vehicle space. Our team of accomplished faculty advisors, engineers and business members include:

- Prof. Song Han (Assistant Professor, EECS) - Song received his PhD from Stanford and his thesis focused on efficient algorithms and hardware for deep learning. He proposed "Deep Compression" technique that widely impacted the industry. At MIT, Song directs the HAN Lab focusing on developing high-performance, accurate, and efficient neural network architectures under low computational resources.
- Jorge Castillo (Industrial Engineering + MBA Candidate) - Prior to heading up the business and operations team at MIT Driverless, Jorge led a business and development team for a multi-billion dollar latin american company, Vitapro. Jorge led Vitapro's expansion into Mexico and subsequently managed the team which took over those operations.
- Shikhar Kumar (PhD Candidate, Nuclear Engineering) - Shikhar's current research involves working with the nation's largest supercomputer to model nuclear reactors. Prior to leading MIT Driverless' computer vision group where his team develops perception algorithms, Shikhar studied financial planning at Columbia University.
- Aaron Ray (PhD Candidate, CSAIL) - Aaron's background in high-speed motorsports includes experience with both MIT Driverless and Brown Formula Racing. Before joining the team as a planning software engineer, Aaron built simulation infrastructure with Kitty Hawk for the recently released eVTOL vehicle.
- Dan Reilly (Mechanical Engineering + LGO Candidate) - With MIT Driverless Dan is responsible for developing industry partners and managing all sponsor relationships. Prior to the role, Dan led the General Electric quality and production teams which manufactured gas turbines at factories in 3 different states with an annual budget of $100M.
- Nick Stathas (Computer Science Engineer) - At NVIDIA Nick developed infrastructure for the Driveworks team, a group focused on enabling the autonomous vehicle engineers of

the future. With MIT Driverless he leads a group of engineers developing high accuracy objection detection algorithms using LiDAR.

- Charlie Vorbach (Computer Science Engineer) - as the controls system architect for MIT Driverless Charlie leads the groups developing both the high level and low level controls. Charlie is a racing fanatic, and prior to his role developing autonomous vehicle software he worked on the MIT Motorsports embedded software team.
- Sibo Zhu (Artificial Intelligence Research Assistant, EECS) - As the director of artificial intelligence at MIT Driverless, Sibo leads a group of engineers developing deep neural networks for both perception and planning. Outside of his work with the team, Sibo's research focuses on bridging the gap between academia and industry by bringing efficient, high-performance, low-latency learning-based algorithms to real life.

4 Software Architecture

We intend to split our pipeline up into 4 distinct sections, each with their own dedicated group. A high level overview of each of these sections is as follows and shown in Figure 2:

- Perception: The perception system will take in data from the cameras, LiDARs, radars, and all other sensors available and output the driveable space in the immediate vicinity of the vehicle. It will also be responsible for localizing all perceivable agents nearby.
- Mapping and Localization: This system will fuse data from the odometry sensors in consort with the perception data to simultaneously produce a map of the environment based on the known track geometry. It will also output the vehicle state and the state of other agents.
- Trajectory Planning: The trajectory planner will ingest the map of the track, the vehicle's current state, and the state of any other agents around the vehicle. The planner will simultaneously predict the future actions of the other agents as well as plan a raceline optimized trajectory for the vehicle based on an estimated friction coefficient.
- Controls System: Based on the trajectory planner's raceline, the controls system will determine the optimal current outputs to the actuators. This will be split into high-level controls which will determine the optimal acceleration and heading profiles, and low-level controls which will convert these profiles into actuator currents.

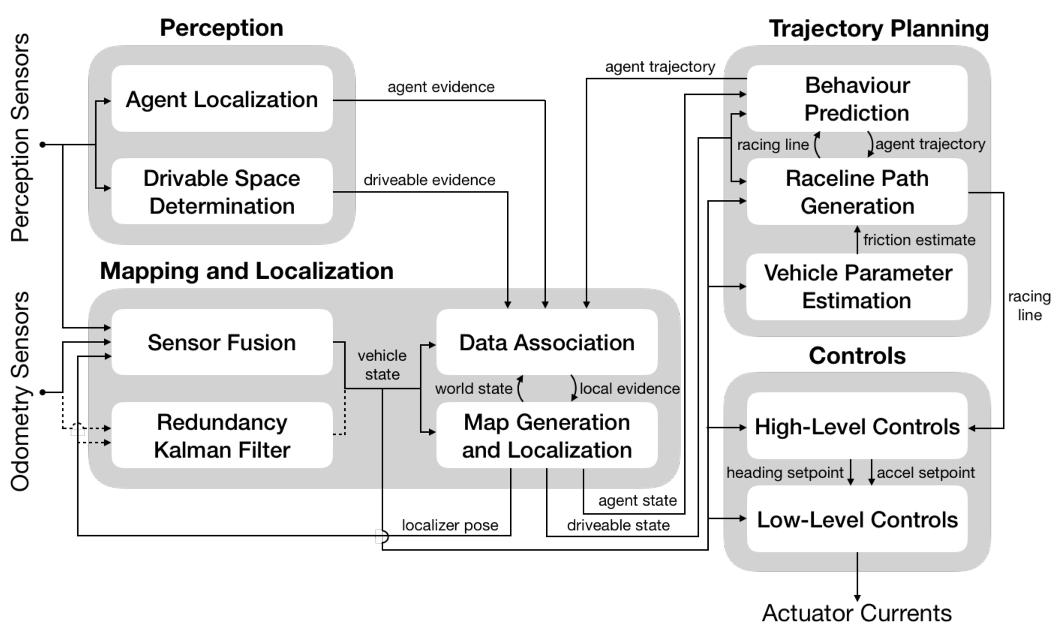

Figure 2: Autonomous Pipeline Concept

5 Perception

Our group has a wide range of experience using camera and LiDAR based vision. Because of our past successes with these sensors we intend to call on this experience for developing a robust and highly accurate perception system to determine the drivable region and localize all external agents. The general strategy will be to use the camera based vision for schematic understanding of the environment (i.e. 2D space localization) and then, combined with the relative pose of the LiDARs and cameras, use LiDAR based vision for ranging.

For semantic understanding we intend to use a machine learning based approach with a separate network for each of the two tasks. Agent localization will be accomplished using a custom YoloV3 [8] based network developed internally at MIT Driverless. Over the last 2 years of development, our work has reduced the latency of this neural network by over 20x and reduced its misclassification rate by 50% relative to the vanilla implementation [12]. This makes it possible to use such a deep network for 2D space agent localization. A common pitfall of most 2D space localizers is the noisiness in the bounding boxes which can cause issues for downstream networks as the output of Yolo is constantly jittering. To smooth this out, we intend to deploy the recent work from the MIT HAN lab, the Temporal Shift Module [5] which exchanges information from the feature maps of neighbouring frames to inform the current frame of what happened in the past. This greatly reduces jitter without any computational burden. For driveable region determination we are currently still exploring machine learning based approaches however we believe that an image segmentation approach will be most appropriate.

We also intend to explore a machine learning based approach for agent ranging using LiDAR. After the point cloud is filtered using the detection from the camera, this information needs to be turned into a single range estimate for the agent's location. For this, we intend to draw on our group's experience developing PVCNN [6] which is capable of segmenting point clouds to separate the agent from the background. With the agent being segmented, taking the centroid of the remaining point cloud would then be possible. PVCNN takes advantage of the low memory footprint of point cloud based models, and the computational efficiency of voxel based models.

6 Localization

As the sensor fusion output feeds almost all downstream nodes it becomes a high risk node for failure. To compensate for this, a dual redundant vehicle state estimator will be used. The main estimator will be based off of a custom built extended kalman filter with output prediction capable of accurately localizing the vehicle at 250Hz with 10cm accuracy. In its current use case with an ADIS16497 IMU and dual antenna Trimble BD992 GPS polled using a pixhawk flight controller, the system outputs state estimates with latencies of approximately 5ms. For simplicity purposes and risk mitigation, the redundancy kalman filter will be based solely on odometry without input from the perception system. Information about the current built up map and the anticipated map (the Indy Oval) will also be leveraged to assist in localization.

The exact formulation of how the map generation process will be combined with the sensor fusion process is still yet to be defined. On our current platform, we are developing a SLAM system based on iSam2 which will be considered for the Indy Challenge. Prior to map generation, evidence of the world will come in from the perception system and be filtered and matched to the existing map in the data association node. In the past we have accomplished this using a Multiple Hypothesis Tracker algorithm [2]. This filtered local data will then be fused with the current map in the mapper node.

7 Trajectory Planning and Behaviour Prediction

Our trajectory planning subsystem will generate high-level plans for the controls system to follow. These plans must take into account both static track layout (e.g. the optimal racing line around each corner) and dynamic responses of other cars (e.g. overtaking and collision avoidance). Traditional autonomy systems have often considered these pieces separately. A behavior prediction system predicts how other agents will behave over a short horizon, and then a path planning algorithm finds the fastest collision-free path. These systems often attempt to plan paths that are guaranteed to be collision-free under any reaction of other agents which we believe falls short in competitive situations such as racing. If some cars always yield to others in order to prevent collisions, then an aggressive car can take advantage of these safer cars by driving more dangerously knowing that the safer cars will yield. As a result, a car planning a guaranteed collision-free path must be overly conservative. We believe the solution is to leverage recent work in simultaneous path planning and behavior prediction in competitive game situations. Recent work from MIT has shown that directly considering opponents responses during the planning process can dramatically improve performance in drone racing [11], and online estimation of cars' aggressiveness can greatly improve trajectory estimation [10].

In addition to these multiagent planning considerations, our trajectory generation pipeline will consider optimal racing lines. Leveraging existing work on racing line optimization [1], our car will be able to follow the globally optimal racing line. We will extend existing offline optimization tools to account for observed friction on the track, which may deviate from the predicted friction map due to modeling inaccuracies, weather differences, or change during the race (tire and track heat changes). The ability to loop the online friction estimation directly into the planning and control algorithms gives us hope of exceeding the limits of how well a human driver can perform.

8 Controls

Our controls strategy will use a high-level predictive controller in combination with a low-latency low-level feedback controller. The high-level controller is intended to be formulated as a nonlinear model predictive controller. Assuming the vehicles have individually actuated brakes, the low-level controller will perform traction control and torque vectoring at a higher sample rate than the computationally intensive high-level controller. This fast feedback allows us to control dynamical responses which would otherwise be too fast for our high-level controller. In addition to this, our low-level controller can use feedback cancellation to simplify the vehicle dynamics the high level controller must account for and thereby reduce solve times.

To aid the trajectory planner's friction estimates we can perform online parameter estimation for our vehicle dynamics model from driving data with a fitted value iteration scheme. This allows us to refine a greybox vehicle model to real-life conditions, including track friction and tire temperature. This will greatly improve the accuracy of the high-level predictive controller.

9 Training Infrastructure

To aid in designing and training the perception system networks and any others downstream we will deploy the Once For All [3] network, developed by the MIT HAN lab. The network builds a specialized neural network for specific hardware by selecting custom architectural settings (depth, width, kernel size and resolution), without any additional training. The result of this is a low-latency network customized for the hardware it is being deployed on. To further reduce network latency model compression and quantization will be used to allow complex models to run with limited compute resources. Conventional model compression/quantization techniques rely on hand-crafted heuristics and rule-based policies that require domain experts to explore the large design space. To accelerate this process, we use MIT HAN lab's recent researches: *AutoML for Model Compression* [4] and *Hardware-aware Automated Quantization* [13], which both leverage reinforcement learning to provide model compression and quantization policies which consistently result in networks with higher accuracy, low latency, and lower memory footprint than their conventionally tuned counterparts. By utilizing AutoML and Once For All neural network architecture search, we have a whole pipeline that automatically design and optimize deep learning model, improving performance and efficiency significantly. It is worth to note that our AutoML techniques can outperform human designed model's performance and received 1st place in Low Power Computer Vision Challenge, both classification and detection track.

10 Testing

The majority of our testing efforts will be focused on physical testing. Although we have benefited significantly from the CI/CD and automated simulation infrastructure we've set up, a key contributor to our team's success during the 2018-2019 Formula Student Racing season was the development of our testbed. Shown in Figure 3, our testbed is a quarter scale version of our race vehicle which enabled the team with a low-cost and low-activation-energy avenue to test out features they had been developing on a weekly, and sometimes daily basis. Although scaled down, the testbed has a full set of the autonomous hardware which greatly simplifies integration onto the large vehicle as many of the systems-level issues are quickly rooted out. An additional benefit is the lack of constraints on testing space this provides. Since the vehicle is much smaller, it can easily be tested in parking lots rather than needing to find designated testing areas, yet it is still capable of going 80kph. This top speed puts the vehicle into a regime where the models used by the controls team must be dynamic rather than kinematic which is similar to those that will be used on the full scale Indy-Light. For full scale testing, the team will use our connections at Palmer Motorsports Park to test out the full vehicle at its design speed in a representative environment.

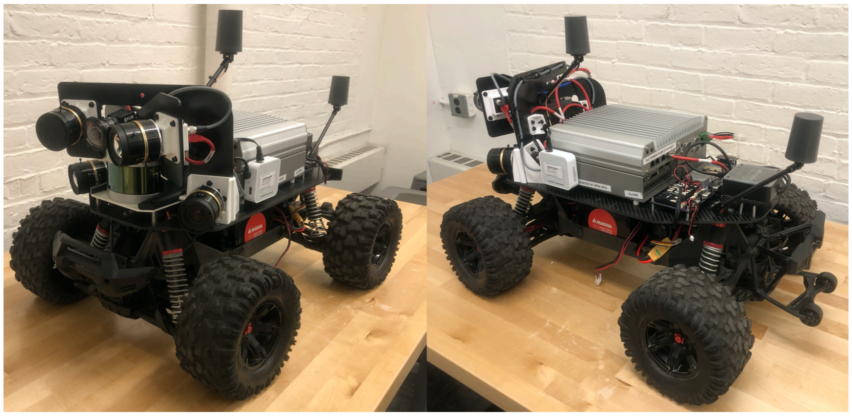

Figure 3: 25% scale testbed vehicle of MIT Driverless

11 Fundraising

MIT Driverless has developed strong relationships with important industry players (Magna, Waymo, Arrow, etc.). Currently, 100% of the team is funded by corporate sponsors who not only value the branding opportunities that the team provides but also the chance to recruit talented MIT engineers who have experience in applied autonomy. We are already having discussions with our current sponsors to extend their support and funding for the Indy Autonomous Challenge. Additionally, the team is engaging in promising discussions with new potential sponsors, as well as some internal departments within MIT which could also provide funding.

12 Project Management

As shown in Figure 4, the team's organizational structure will be set up as follows:

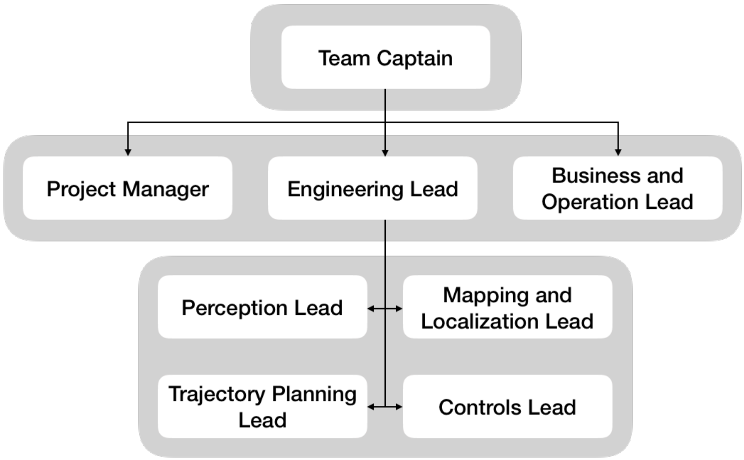

Figure 4: MIT Driverless Organizational Structure

The captain position, engineering lead, and two sub-team leads will be filled with PhD students and postdocs from the following labs:

1. HAN Lab, led by professor Song Han, will provide PhD students to lead the perception pipeline.
2. We are currently in conversation with CSAIL to provide postdocs and PhD students to fill the Team Captain, Engineering Lead, and Planning Lead positions.

The localization and controls leadership positions will be covered by two of the current sub-team leads in the MIT Driverless team. Additionally, the 20 engineers currently working in the team will continue to be part of the 4 engineering departments. The current business operations team, composed of MBA students from MIT Sloan, will continue leading the fundraising, partnership management, procurement, marketing, and HR efforts. Finally, a technical product manager (TPM) will be recruited from the Product Management Group at MIT Sloan. The TPM oversees the project timeline, sprints, leads status/planning meetings, etc. The team uses Asana as its main planning tool.

References

[1] Global racetrajectory optimization. `https://github.com/TUMFTM/global_racetrajectory_optimization`. Accessed: 2020-02-26.

[2] Samuel S Blackman. Multiple hypothesis tracking for multiple target tracking. *IEEE Aerospace and Electronic Systems Magazine*, 19(1):5–18, 2004.

[3] Han Cai, Chuang Gan, and Song Han. Once for all: Train one network and specialize it for efficient deployment. *arXiv preprint arXiv:1908.09791*, 2019.

[4] Yihui He, Ji Lin, Zhijian Liu, Hanrui Wang, Li-Jia Li, and Song Han. Amc: Automl for model compression and acceleration on mobile devices. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 784–800, 2018.

[5] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7083–7093, 2019.

[6] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. In *Advances in Neural Information Processing Systems*, pages 963–973, 2019.

[7] Hugh MacInnes and Betty MacInnes. *Turbochargers*. Penguin, 1987.

[8] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[9] Stephen W Rouhana, Paul G Bedewi, Sundeep V Kankanala, Priya Prasad, Joseph J Zwolinski, Alex G Meduvsky, Jonathan D Rupp, Thomas A Jeffreys, and Lawrence W Schneider. Biomechanics of 4-point seat belt systems in frontal impacts. Technical report, SAE Technical Paper, 2003.

[10] Wilko Schwarting, Alyssa Pierson, Javier Alonso-Mora, Sertac Karaman, and Daniela Rus. Social behavior for autonomous vehicles. *Proceedings of the National Academy of Sciences*, 116(50):24972–24978, 2019.

[11] Wilko Schwarting, Alyssa Pierson, Sertac Karaman, and Daniela Rus. Stochastic dynamic games in belief space. *arXiv preprint arXiv:1909.06963*, 2019.

[12] Kieran Strobel, Sibo Zhu, Raphael Chang, and Skanda Koppula. Accurate, low-latency visual perception for autonomous racing: Challenges, mechanisms, and practical solutions.

[13] Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. Haq: Hardware-aware automated quantization with mixed precision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8612–8620, 2019.

White Paper for the Indy Autonomous Challenge

Nayana Suvarna, Joshua Spisak, Andrew Saba, and Prem Bharatia
University of Pittsburgh

Introduction:

With the rise of autonomous vehicles in research and industry, robust autonomy algorithms and systems are in high demand. Companies such as Argo AI, Uber ATG, and Waymo are all working towards creating autonomous vehicles that can be deployed in the general public. The Indy Autonomous Challenge strives to have teams tackle this emerging need in the rather unique environment of a racetrack. This competition offers an edge case to rival current advancements in the field by having teams race head to head in a high-speed competition.

This whitepaper will give some background information on our team, and introduce technologies related to robotics on a high-level. After which it will go deeper into the challenges to these areas that are introduced by the high-speed environment. Finally, it will discuss some of our approaches to these challenges and our approach as a whole. It is important to note that since the hardware platform has not yet been determined much of this is based on what sensors and compute we expect we might have. Once the hardware platform is determined our strategies will be developed in greater depth in a way that may or may not line up with the technologies and approaches discussed in this paper.

Our Team:

Our team is comprised of members from the University of Pittsburgh's Robotics and Automation Society (Pitt RAS) club. In addition, a significant number of our team are former Pitt RAS members who are now master students, full-time engineers, or research interns at the Robotics Institute at Carnegie Mellon University. Overall, our team has a unique mix of individuals with varying skills and experience.

From involvement with the Robotics Institute, many of our team members have worked on cutting edge technologies and research areas in robotics, including Unmanned Aerial Vehicles (UAVs), Simultaneous Localization and Mapping (SLAM), localization, multi-agent planning, mapping, motion planning, object detection and classification, and more. Other members of our team have engaged in personal projects that have allowed them to gain experience with computer vision, hardware development, and systems integration. Due to the diverse background that our team possesses, we believe that we have the resources and knowledge to do well in this competition.

Current Technologies:

Robotics is an interdisciplinary field; for a robotic system to operate efficiently, many mechanical, electrical, and software components need to be designed and integrated successfully. If any of these areas are lacking, the system as a whole has suboptimal results. Since this competition is software focused, we will be focusing primarily on software-related solutions.

Perception:

A robot's perception system's purpose is to take in multiple sensor sources in order to gain information about their environment and surroundings. Often the sensor data comes from a

LIDAR, which provides highly accurate long-distance information; or a high-speed camera, which provides a much higher resolution and rapid refresh rate. The actual task of identifying objects is accomplished using machine learning and other algorithms, which process incoming data in real-time to identify and track features in the environment.

The multi-modal aspect of perception is particularly important to robust sensing. Another vehicle may be identified from inter-car communication, LIDAR point-cloud processing, and camera images. To combine and make sense of information from these three modalities, the state estimates can be fused in something like an Extended Kalman Filter (EKF), which provides a statistical and non-linear filter for state estimation. By fusing these inputs intelligently the robot will be able to maintain an accurate awareness of the environment regardless of sensor dropout or inaccuracies.

Localization:

Localization is the generalized field of maintaining an accurate belief of where the robot is in the world. Through sensor sources as well as visual features, a robot can infer it's position in the world. With an accurate pose estimate, it can both navigate and interact with the environment around it. Localization is often difficult to accomplish due to noisy data from sensors as well as the prevalence of featureless environments.

In order to tackle this concern, a variety of state estimation solutions are applied. But, the most common application is an Extended Kalman Filter (EKF). Although there are many solutions meant to span a range of applications, the base implementation relies on multiple sensor estimates that are globally "correct" but jump, such as GPS and other estimates that drift over time but are continuous, such as wheel encoders, and Inertial Measurement Units (IMUs). Our approach to localization is discussed in greater depth later.

Mapping/SLAM:

In order for a robot to accurately navigate in an environment over time, it needs to understand the location of the objects it perceives in the world around it. It can then use this map to make inferences about the world as well as craft a global plan or trajectory to follow.

A common solution for mapping is Simultaneous Localization and Mapping. This process combines both localization and mapping into one function to give a robot the ability to both create a map of the world around it as well as place itself and localize the said map. Most SLAM algorithms take in data from LIDARS, cameras, IMUs, and laser range finders in order to accurately reflect limitations and obstacles in the environment.

These sources are used to create a 3d point cloud map of the world. A point cloud is a set of points within a space that are usually extracted from depth measurements. Depending on the quality of measurements a point cloud the quality of the cloud can range from being sparse to being dense. In order to make this map a useful source for planning, a 2D occupancy grid is created. An occupancy grid is a grid representation of a space where zero indicates free space and numbers above zero indicate free space. It is created from the projection of the 3D point cloud onto a 2D plane. Motion planning algorithms utilize this occupancy grid to navigate in the environment.

Motion Planning:

Motion planning is the culmination of mapping, perception, and localization. The robot uses its current map and its position, obtained via localization, to plot its next course of action (i.e. lane changes, swerving in a lane, etc.). The planner takes into account both the current goal and obstacles present on the map to chart an ideal path. This path is constantly being updated to account for dynamic changes to the environment.

It is important that motion planning occurs rapidly, as the vehicle is moving very fast and the map is also being updated at a high rate. Conventional search and optimization-based planners and trajectory generators suffer from potentially being unable to generate a valid plan, which leaves the vehicle in a position where it does not have an updated path to follow. To get around this, state of the art systems use an ensemble of planners, which reduces the chance that the whole system cannot produce a viable plan.

Controls:

Most low-level control structures are implemented using Proportional Integral and Derivative (PID) control loops. A PID controller generates a control signal to reach and maintain the desired state that is a function of the target state and the current state, which is obtained via sensor data. The target state and current state are compared to produce an error, which is fed to the PID control algorithm. The PID controller then takes a weighted summation of the proportional error and the integral and derivative of the error over time to generate a response. The weights used during the summation are parameters that must be tuned in order to achieve the optimal performance of the control loop.

Challenges:

The main challenge of this competition is the issue of speed and the effects that it has on the autonomous performance of the vehicle in addition to the presence of other vehicles on the track. Since the competition is an extreme edge case in regards to current robotics work being done in the field, it tests the limits of current algorithms widely used and implemented in existing applications.

Perception:

With computer vision, we will have to detect the other vehicles in both a fast and accurate manner. A traditional object detection algorithm distorts the image and runs it through a neural network or other feature extraction algorithms. This process is often resource-intensive and slow. The fast nature of the competition precludes slow algorithms. Most of our efforts on this front will be placed towards optimizing our algorithms so they can run at the rate required for the competition.

Localization:

The big challenge with localization is the significant drift that will occur from traditional filter sources. Traditionally, odometry data is highly susceptible to drift which is further exacerbated by the high speeds and can cause poor mapping conditions.

Mapping/SLAM:

Most of the challenges occur when trying to achieve loop closure while creating the map. Loop closure is the stitching of two different maps based on the recognition that a location has been visited before. There will also be noisy point cloud data from other cars surrounding our own. This will create challenges since the creation of a map depends highly on having many distinguishing features in an environment.

There are several challenges as it pertains to SLAM at high speeds. Many popular SLAM algorithms rely on point cloud data from a LIDAR, but time synchronization issues can occur between different sensors when a vehicle is moving at high speeds. This time synchronization is necessary in order to achieve both loop closure and to create a map that is reflective of the environment. In addition, the imperfect nature of circling the track may cause distortions with our map.

Motion Planning:

The main challenge with motion planning is having a planner that can both predict the actions of other vehicles as well as navigate around them. The probabilistic dynamics and constraints for the car may prove to be computationally expensive. This is further complicated by the algorithm having to be performed in real-time.

Controls:

When discussing PID control for use in autonomous vehicles there are two main areas of concern. The first issue is that due to ever-changing road conditions the desired state, or “setpoint”, maybe constantly shifting or even oscillating. This means that we will have to tune our controller to be more aggressive such that the output will match the setpoint faster. However aggressive tuning has its own issues such as higher overshoot and lower margin of stability, which can translate to oversteering and even a complete spin out for the vehicle. So then our challenge becomes to tune the loop so that it is sufficiently aggressive while also minimizing overshoot and maximizing stability.

Another key issue is that the dynamics of our system may change over time. Since classical PID control assumes that the input/output relationship of the system is static, as the dynamics drift from the initial system our control loop will become slower and less stable as a result. This means that we may need to figure out how to recompute the dynamics of the system and return the PID coefficients on the fly in order to maintain optimal movement. If we find that PID control is insufficient for the resolution and control necessary to operate at high-speeds with

Ackerman steering, we are prepared to implement an LQR control scheme instead. While this increases the complexity of control it should achieve robust control for this application.

Our Approach:

In order to perform despite these active challenges in vehicular autonomy, we will utilize several approaches. As discussed in the introduction there is some uncertainty as to what will be on the competing vehicle so we are providing an approach that accounts for varying degrees of specifications of the competition. We anticipate that the car will have a sensor suite including lidars, cameras, encoders, GPS, IMUs, and radar and will strategize around that.

Overall Software Stack:

All the members of our team have extensive experience with the Robot Operating System (ROS). ROS is a comprehensive framework for inter-process communication, data visualization and various other software capabilities that allows for the rapid integration of software for robotics systems development. If ROS is unable to run the algorithms at the high rate required for the competition, we have several members who are experienced with multi-threaded applications and inter-process communication who would be able to craft a custom software framework that is able to meet our needs.

SLAM:

While there is some uncertainty as to what sensors will be on the vehicle, our overall SLAM strategy will involve feature extraction from a few data-rich sensors while using wheel odometry and IMU data or basic interpolation of velocity and acceleration data to achieve the update rate required for planning algorithms.

A few examples of what data-rich sensors might be used as inputs are global-shutter cameras, LIDAR sensors (like the VLP-16) or perhaps a radar. This data must be filtered to remove distortion due to the high speed (with either image processing if necessary or individual point stamping). Then another filter will be used to differentiate dynamic features from static features. The static features can be used to localize unto the track and build a map thereof. The dynamic features (cars) can be tracked over time for use in the planning algorithms.

The odometry from data-rich sensors and wheel odometry and IMU data will continually feed into each other to provide a reliable position The integrated position from wheel odometry and IMU data will give a high rate pose estimate that can be used as a baseline for the data-rich sensors to localize unto the map from. Once these data-rich sensors determine a pose they can feed that back into the high-rate odometry.

It is important to note that these sensors must be positioned somewhere they will detect features around the track and not solely dynamic features or features common to the whole track (eg: part of a railing or a mark on the track itself that is repeated the length of the track). While we find this case unlikely, if no such data is available we will adapt to a different SLAM approach that is driven by internal sensors and perhaps a GPS if that is available.

Motion Planning:

Although the amount of information shared between the vehicles has not been finalized yet, we presume that this information can be used to influence our decision. We plan to use the information swapped between cars in order to predict the future movements of race cars around us. Depending on what information is available, we will attempt to fill in the gaps in the data with statistical filtering and prediction to estimate future vehicle movement and avoid collisions.

Testing Approach:

Simulation:

The Gazebo simulator is a robust simulator that allows for close to real-life testing of robotics systems. Simulation is a great testing tool for observing how software and physical systems translate to real life, but one of the drawbacks is that it occurs under ideal conditions. When dealing with physical systems there are many uncontrollable factors including lighting, weather, as well as imperfect controls, and actuation that cause unintended results. Although many details of the simulation are unknown, we will be taking these factors into consideration when receiving and working with the simulation software as well as developing the software for our vehicle.

Physical System:

In order to test our physical system, we plan to coordinate a test track with the Formula SAE team at the University of Pittsburgh to safely test our vehicle. Every year in Pittsburgh, PA the Pittsburgh Shootout event is held for Formula SAE teams. The event is held at the Pittsburgh International Race Complex. We plan on coordinating with managers in this space to gain access in order to test and validate the performance of our vehicle.

References:

Choudhury, S., Dugar, V., Maeta, S., MacAllister, B., Arora, S., Althoff, D. and Scherer, S. High performance and safe flight of full-scale helicopters from takeoff to landing with an ensemble of planners. *Journal of Field Robotics*, vol. 36, no. 8, pp. 1275–1332, Dec. 2019.

Li, Tuan & Zhang, Hongping & Gao, Zhouzheng & Chen, Qijin & Niu, Xiaoji. (2018). High-accuracy positioning in urban environments using single-frequency multi-GNSS RTK/MEMSIMU integration. Remote Sensing. 10. 205. 10.3390/rs10020205.

Zhang, J, Singh, S. Laser–visual–inertial odometry and mapping with high robustness and low drift. *J Field Robotics*. 2018; 35: 1242– 1264. https://doi.org/10.1002/rob.21809

Zhao, Shibo & Fang, Zheng & Li, Haolai & Scherer, Sebastian. (2019). A Robust Laser-Inertial Odometry and Mapping Method for Large-Scale Highway Environments. 10.13140/RG.2.2.32095.82089.

RIT Autonomous Racing

Andrew Keats
Computer Engineering
Rochester Institute of Technology
Rochester, USA
axk7655@rit.edu

Abhishek Vashist
PhD Engineering
Rochester Institute of Technology
Rochester, USA
av8911@rit.edu

Karan Manghi
Computer Science
Rochester Institute of Technology
Rochester, USA
kxm9436@rit.edu

***Abstract*—With recent advancement in hardware and software design we have seen rise in the field of research for autonomous driving in past few years. There is a lot of research going on in this field and autonomous driving is being expanded to various other fields like space exploration, delivery robots and industrial transport vehicles. With this project, we would like to design, research and implement an autonomous vehicle for car racing. Car racing is a more complex subfield of autonomous driving as the autonomous system is not only responsible for safe driving but also participating in the competition of racing. The competition of racing requires minimizing lap times though increase speed and trying to hit certain way points along with trying to gain position on the track and avoid crashing into other competitors. Further, the environment itself introduces new challenges and opportunities for design of autonomous vehicles. Learning algorithms uses information from variety of on board sensors which are highly dependent on the features they grab on to. So, in this paper we propose the details regarding the process and the procedure on how we plan to going about with the project, how the management of the project is going to be, how we are going to get the required funding and also how we are capable technically to carry out the various required tasks.**

I. Introduction

This paper is designed to outline the Rochester Institute of Technology's (RIT) official entrant into the Indy Autonomous Challenge, RIT Autonomous Racing, and the team's plan to meet the goals of the competition. This paper will outline the history and formation of the team, previous experience with autonomous systems, team management and plans to be successful in the Indy Autonomous Challenge.

A. Team History

The team was formed at the end of November and is recognized by the Kate Gleason College of Engineering as RIT's official entrant into the Indy Autonomous Challenge. The first recruitment meeting was performed at the start of January, 2020. The team was formed specifically to compete in the Indy Autonomous Challenge, but pulls students from other racing competitions like Formula SAE.

B. Team Composition

The team currently consists of a talented and diverse group of 40 students from RIT. This includes students from the Kate Gleason College of Engineering, Saunders College of Business, Golisano College of Computing and Information Sciences, College of Engineering Technology and College of Science. Due to having students from a wide selection of colleges at RIT, this means that the team has a variety of majors being studied to help support the team's efforts. This also includes majors not typically associated with machine learning or autonomous solutions. Members of our team are also receiving degrees in imaging science along with web and mobile development. Currently the team comprises both graduate and undergraduate students, including master students in the discipline of autonomy. Currently members of the team are conducting research to provide autonomous operation for industrial equipment. This includes adding localization to fork trucks using 60GHz millimeter-wave (mmWave) by designing a machine learning based localization system and UWB beacon using time of flight information.

II. Team Background

Two (2) students from the team are also involved with a project where they are working on automating a forklift for a warehouse. The goal of the project would be to use various sensors, similar to this autonomous car racing project, to make the forklift capable of carrying on various complex warehouse tasks on its own without human intervention. Hence the knowledge gained there could be transferred here and hence in this way the team would gain expertise in automated driving systems.

A group of eight (8) students from the team are also participating in a local AWS Deepracer competition. The final race for this competition will be in March, but will provide valuable skills to racing that will carry over into the Indy Autonomous Challenge. This includes training reward based machine learning models, racing line, turn shape and speed control.

III. Competition Development

The section outlines current assumptions about the competition and how the teams plans to tackle the different challenges of an autonomous vehicle.

A. Software and Hardware

All the code that we write would be in either a python based framework like Keras,Tensorflow,PyTorch, etc or C++ or a mixture of both depending on what base libraries are available for the task. Also, we feel that we would use ROS and its libraries like Gazebo and RVIZ for simulation before we get the actual car or its simulator. We have already started to simulate a small Ackermann Steering based car on ROS and Gazebo to see how we could give velocity commands to it and how we could get the odometry from the vehicle.

B. Navigation Localization

Localization and navigation are two of the fundamental requirements of any autonomous system. The localization enables the agent to answer the question of where I am given an environment. Further, navigation uses the localization information to carry out the required task like point-to-point navigation. Various information or features are used to provide the localization information to the autonomous agent. For this different sensors are used like GPS/IMU, LiDAR, RADAR. The idea is to build a map of the environment and then based on that the vehicle can localize. One of the most common and heavily used techniques is called Simultaneous Localization and Mapping (SLAM). Where typically LiDAR is used and a map is generated based on the sensor reading of the environment, then the generated map can be saved and used by the agent to provide the location information. The accuracy of SLAM varies based on the dynamics of the environment and the sensor used, as different sensors are susceptible to different environment changes. To mitigate the shortcomings of different sensors, multiple sensors are used to provide more accurate and robust localization and navigation system. GPS can be used to provide a good startup estimate with good accuracy but considering a 2D positioning, the accuracy of GPS is good in the y-dimension (down the road) compared to the x-dimension. Considering the autonomous movements we require high precision in both the dimensions. For this, sensor fusion approach can be used where different sensors can provide the certainty of the position and can be combined in a probabilistic approach to estimate accurate and precise location. Extended Kalman Filter (EKF) and particle filter are two common approaches towards sensor fusion. In the case of autonomous racing, we would first map the entire circuit and then use localisation in that mapped area. The challenge would be to localise itself given various obstacles like the other race cars and maybe even flying debris. Hence, we would have the car learn various landmarks on the map and for localisation it would refer to those previously marked landmarks. In many cases, the landmarks would be occluded by other cars and hence we would have to use GPS along with the other sensors for localisation.

C. Obstacle Detection/Avoidance

The autonomous agent while in motion should be able to identify many different obstacles and objects. The information from it is used in path planning to estimate the best path for navigation. Further for the safety of the surrounding and the vehicle the object detection is very critical and requires very high classification accuracy. For this, information from vision based sensors can be used to train different classification systems based on machine/deep learning systems. Typically cameras are used to record the images of the surroundings and the images provide the features on which different models can be trained and deployed, then in turn we could use this data from the models to train a model.

D. Path Planning

Path planning can be viewed as the task of moving an autonomous agent from a given location to a final destination or goal in the shortest possible time while avoiding any obstacles in between. Different algorithms like A* (A-Star), D* (D-Star), Dynamic Window Approach (DWA) and greedy best fit search can be used to provide the path planning information for the autonomous agent. Here the time complexity is of the main concern, as we require the shortest possible time for the algorithm to generate the path at runtime. Many different hybrid approaches are also used with the A* to optimize for time. Recently many Reinforcement Learning (RIL) techniques are developed where the initial estimate can be provided by the RIL for the optimizer to solve, resulting in reduction of convergence time. Further, deep learning approaches can also be used to provide the path information but require generation of the dataset and simulation models to train the models. For path planning there would exist two different planners namely global and local. The global planner would be responsible to route the car to the finish line while the local planner would have certain checkpoints and would be responsible to move the car from one checkpoint to the other while also taking into consideration various dynamic obstacles such as other cars. The local planner would modify the global plan based on which route would be the safest to take. The shortest path calculations would happen at both global and local planners but they would not be directly responsible for learning the racecraft. The racecraft would be learned by the deep learning model which will come up with strategies according to the rewards and punishments that we provide it with. For example if we tell the model that hitting another car would be a punishment for it, the model will eventually learn that hitting other cars is bad. Hence in this way if we give it a combination of different rewards and punishments, the model will learn over time what is the best way to get to the finish line. So, in this way the reinforcement learning/deep learning will be integrated into the planners and will finally take over the planners and come up with a better and fast strategy. In this case, we would have to use multiple sensors for learning. For example, the camera data would be responsible to let the car know that it is going out of track and that would be a punishment. In the same way the LiDAR data would be used to gauge the distance between our car and the other cars or maybe a wall and this could be used as a punishment or reward as the model learns over time.

E. Sensor Integration and Communication

As many different sensors are used to enable the autonomous driving capability, that means, the computation requirement and the data bandwidth of these sensors are very high. LiDAR and vision based sensors need very high data bandwidth and processing the features from these sensors are very computationally expensive. The need to process the information online at runtime requires a good computation machine and data processing capability on the vehicle. Further as different information/decision for different tasks require to share information among themselves the data bandwidth reliability needs to be addressed during the development phase.

IV. Team Organization

The team is organized into six groups to facilitate in breaking up work and allowing greater participation from those involved. Each group is dedicated to a specific task of the challenge. The groups are path planning, path validation, vehicle physics, data input, localization and race craft.

A. Path Planning

The path planning group is focused on determining the path that should be taken by the race car. We say the path that the car should take as there are times where the optimal path may not be able to be taken. In the case of trying to perform a pass on another car entering turn 3 of IMS, a path closer to the inside of the turn may need to be taken. This type of path would be not optimal in the terms of a shortest path, but would provide an on track advantage to our team. Furthermore the path planning group has to devise a way to save the path that they want to take so that it can be validated by the path validation group.

B. Path Validation

The path validation group's responsibilities include verifying how close the path taken by the car was to the path that was intended to take. This is intended to allow feedback to the car and team to help control outside unknown forces. Air resistance and direction plays a role at IMS. These winds have the tendency to push a car to the outside of a corner and we need to be able to detect this situation and allow the car to correct for it. This group will also work to validate how closely the car can follow the path that was planned as this may indicate a degradation in tire performance or issues with the handling of the car.

C. Vehicle Physics

Vehicle physics is the group responsible for determining how the car reacts to the physical world. This group will also work to determine the suspension geometry and set up for our car to try and achieve the best performance possible. Vehicle physics also has a lot to play into the other groups as they need to understand how the vehicle will perform under certain conditions. This is important to providing a car that can meet the conditions provided on race day.

D. Data Input

Data input is the group responsible for collecting the data from the sensors on the car and interrupting the information. This includes filtering the data that comes in from the outside world. In the case of a camera, we may not want to use portions of data that include the sky and that introduces noise and processing times. This group is responsible for determining how to filter the data and get it into a usable form for the other groups. Another responsibility of the data input group is understanding the sensors that are available and how to effectively use them.

E. Localization

The localization group handles determining where the car is on the track. This is important as the system needs to know its current location on the circuit. This group will use data from the sensors and determine the best information to use. Ideally we want the system to be as accurate as possible in determining the location of the vehicle on the track. Having a low margin of accuracy could spell disaster as the vehicle could be too close to a wall or another vehicle. We want to be able to be as accurate as possible with a minimal processing time so that the systems depending on the information can receive the current location in a timely manner.

F. Race Craft

The race craft group is responsible for the strategy during the final race along with the acts of racing. We believe that due to the competition and other competitors that getting a vehicle to successfully navigate the IMS oval will not be a challenging task, but how we deal with other competitors on the course will be the biggest challenge to overcome. Due to the construction of the IMS oval only allowing a single racing line and being a short race, being able to time passes correctly will be a crucial part of creating a successful system. The race craft group will be responsible for making decisions related to the art of racing. This will include when to pass another competitor, what position to be in on the closing laps, and what type of driving strategy will be used. Traditionally the Freedom 100, which employs the same chassis that is used for the competition, has had a pass for the lead on the last lap. This makes race strategy very important to being able to win at the Indianapolis Motor Speedway. The race craft group is looking to tackle these challenges so that our system may have the same race planning skills and a driver and their team on race day.

V. Project Management

The team is currently recognized by the Kate Gleason College of Engineering at the Rochester Institute of Technology and is the official team from RIT participating in the Indy Autonomous Challenge. The team is led by a faculty advisor and two project leads. The current faculty advisor to the team is Amlan Ganguly the Interim Department Head for the Computer Engineering Department. The project leads are Andrew Keats and Mark Chang currently both 3rd year undergraduate students in the Computer Engineering department at RIT.

Currently the team meets once a week to provide updates from each group. This includes team announcements and any information that all members must be aware of. We are using GitHub and the Google Suite of products for file and code sharing. These items will help to distribute the code being used and to allow tracking of changes made. We are also using Discord to allow communication for the team. This helps with providing different spaces for people to communicate ideas instead of having one long email or text chain.

VI. Sponsorship/Fundraising

Since the team is a new organization at RIT, there are many challenges that we must face in providing funding for our effort in the competition. One of these challenges is knowledge of the program and what our goals are. RIT host every year an event called Imagine RIT. This event is a college level science fair that takes over the campus to showcase student projects and organizations. One of our goals is to participate in this event to help draw attention to the program and possibly even make sponsorship connections.

Another challenge that we are facing is a lack of sponsorship connections or inability to meet companies sponsorship requirements. Having talked with other organizations at RIT, many of their sponsors pay for a specific part of the project. In the case of RIT's Formula SAE team a company may pay for the engine or the carbon fiber that is used. Since there is a lack of small individual items that have to be covered by the team, we have found it difficult to convince companies to pay for a portion of the Dallara IL-15 chassis as it is also difficult to convince one company to pay for the entire chassis.

Currently RIT Autonomous Racing is working with RIT's Corporate Relations Officer to help in initiating connections with companies for sponsorship. This includes many local companies to Rochester, New York. The team has also looked at running a fundraising campaign to raise money for travel cost and other expenditures. This fundraiser would include perks like t-shirts, sticks, thank you notes and even an invention to the final competition.

VII. Collaboration

At the current time the team is not looking to collaborate with another team. We believe that we have the talent and skill set to meet the competition requirements as a single team. We would also like to avoid the complications of coordinating between two different organizations.

RIT Autonomous Racing does understand that the challenges may change as the competition evolves and that we may need to combine with another team to be successful. We will continue to keep an open mind in the future and will not discourage the opportunity if one comes along.

Acknowledgment

RIT Autonomous Racing would like to thank the Kate Gleason College of Engineering Deans for allowing students to take part in this opportunity. We would also like to thank Dr. Amlan Ganguly in helping to get the engine started.

University of Waterloo's Entry to the Indy Autonomous Challenge

Round 1 White Paper

Rocky Liang, Ross McKenzie, Derek Rayside
{rocky.liang, ross.mckenzie, drayside}@uwaterloo.ca

Introduction

Racing has long been a proving ground for cutting edge automotive technology. Many of the features we take for granted in a modern vehicle, such as rear view mirrors and disc brakes, were innovations that originated from motorsports. It makes sense that autonomous driving, something that is on its way to completely transform not just the automotive industry but also how we live, should be tested in a racing environment as well. The Indy Autonomous Challenge provides an excellent opportunity for students to do exactly that. The University of Waterloo is proud to be announcing its entry to the challenge.

In this paper, we will introduce some of our past and present vehicle projects, vehicle facilities, and relevant faculty/research groups such as Waterloo AI. We will also describe how we plan to tackle the challenge in terms of software design. Lastly, we'll touch on project management and funding.

History in Automated Vehicles

With more than 60 dedicated faculty members and researchers working on connected and autonomous vehicles, the University of Waterloo is heavily invested in automotive research. We are home to Canada's largest academic-industry automotive enterprise, the Waterloo Centre for Automotive Research (WatCAR). Under WatCAR, there are two major autonomous vehicle efforts, Autonomoose (Moose) and Watonomous.

Moose is a Lincoln MKZ that's been equipped with a fully configurable suite of radar, sonar, lidar, inertial, and vision sensors. It's used by research groups across several different domains, including perception, motion planning, and power management.

In November 2016, Moose was granted a license to drive autonomously on public roads in a provincial pilot program, and it was the first car in Canada to do so! Less than two years later, Moose has already covered over 100 km on public roads autonomously. Using data collected over the past few years of driving in the harsh Canadian winter, Moose researchers have created the Canadian Adverse Driving Conditions (CADC) dataset, which was released in February 2020. It aims to promote research to improve self driving in adverse weather conditions.

Figure 1: The Autonomoose vehicle

As it is a research platform, Moose is mostly worked on by our graduate students. However, Waterloo's undergraduate students have their own autonomous car as well. Watonomous is a student led design team that competes in the SAE Autodrive Challenge. With membership count in the hundreds, it's one of the largest student organizations on campus. It has four divisions: software, mechanical, electrical, and business. In early February 2020 they used their autonomous Chevy Bolt to give a ride to the Federal Minister of Science and Innovation, the Honourable Navdeep Bains.

Figure 2: Federal Minister of Science & Innovation Navdeep Bains speaks at the AVRIL opening event after taking a ride in the Watonomous Bolt

The Watonomous Bolt even got a chance to participate in the TV show The Amazing Race Canada. As a Speed Bump challenge, contestants from the show took an autonomous ride around campus in our Bolt.

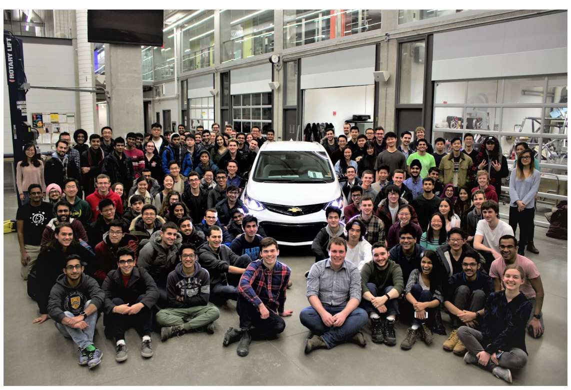

Figure 3: The Watonomous team with their Chevy Bolt

Aside from these two autonomous vehicle teams, Waterloo has several other more conventional vehicle teams. There are two Formula SAE teams, covering combustion and electric. Midnight Sun, our solar car team, has been operating since 1988. They are currently representing us at both the American Solar Challenge and the World Solar Challenge with a custom built solar car. We have also been continuously involved in EcoCAR over the past 22 years, spanning 6 iterations of the competition.

Facilities

The University of Waterloo just opened the Autonomous Vehicle Research and Intelligence Lab (AVRIL), a $4 million dollar research hub for autonomous vehicles. It features 10 truck height bays, level two charging for EVs, and a driving simulator built around a fully functional Chevy Equinox SUV.

Figure 4: The newly opened AVRIL building

The Sedra Student Design Centre (SDC) consists of over 20,000 square feet of space dedicated to design teams and student projects. It includes a sanding bay, paint room, and garages for team projects. It is located right next to one of UW's machine shops, where students can fabricate parts for their projects.

Figure 5: The SDC, with room for multiple vehicle teams to design and build their ideas

Selected Professors Working with AVs

Krzysztof Czarnecki is a Professor of Electrical and Computer Engineering, and he leads the

Moose project. Before joining Waterloo, he was a researcher at DaimlerChrysler Research, focusing on improving software development practices and technologies in enterprise, automotive, and aerospace domains. He was a part of the Moose team that successfully delivered an autonomous vehicle demo at CES 2017, and subsequently drove in autonomous mode on public roads.

Sebastian Fischmeister from Electrical and Computer Engineering performs systems research at the intersection of software and distributed systems with applications in automotive systems, and avionics. His work in reliable and robust embedded systems is crucial to the Moose project. He runs the largest embedded systems lab in Canada.

John McPhee from the Mechanical Engineering Department heads Waterloo's Motion Research Group. He has also served as the Chair for the International Association for Multibody System Dynamics. His research focuses on modeling, simulation model based control, and optimal design of dynamic physical systems, all important things when it comes to making a car go fast.

Dongpu Cao holds the Canada Research Chair in Driver Cognition and Automated Driving, and the Director of Waterloo's Cognitive Driving Lab. He has published over 180 papers and 2 books in the fields of vehicle dynamics/control, driver cognition, driver-automation collaboration, and automated driving; all areas we will need expertise from for the Indy Autonomous Challenge. He has also led a research consortium, CogShift, which collaborated closely with Jaguar Land Rover on level 3 autonomous driving.

Amir Khajepour holds the Canada Research Chair in Mechatronic Vehicle Systems and the General Motors Industrial Research Chair in Holistic Vehicle Control. As the goal of this competition is to get cars driving autonomously at racing speeds, controls will be a huge part of the challenge. Professor Khajepour's insights will undoubtedly be of great help to us.

Derek Rayside is the Director of Software Engineering at Waterloo. He has led Watonomous as its faculty advisor since its inception.

With many knowledgeable professors and a history of fostering successful student design teams, we are confident that Waterloo will be a strong contender in the Indy Autonomous Challenge.

Technical Approach

This section details our proposed autonomous driving pipeline. Considering that the final race will be ran at an average speed of over 120 MPH, it requires the system to have high autonomy ability as well as efficient and real-time computation performance to be safe and competitive. The system consists of four modules: perception, cognition, planning and control.

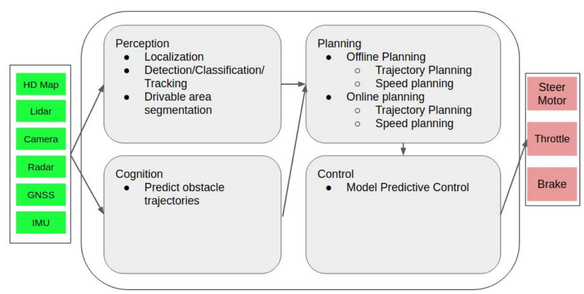

Figure 6: Proposed system architecture

Although not shown in the system architecture diagram, The control module should be able to accept input from not only the onboard planning module, but also signals sent from the race organizers for features such as flagging cars, emergency stop, and calling cars back to the pits.

The perception module enables the racing car to localize itself on the track, and also detect and track drivable areas, other cars, and any other obstacles that may show up on track by using information returned by sensors, together with an HD map. An experienced human racer would memorize all the details of a track, so they can anticipate different parts of the track that's coming up, which makes them more prepared to go through them. Similarly, HD maps also provides the vehicle with such information about the track. It contains information on landmarks, lane lines, road networks, etc, to help the vehicle overcome the limited working range of onboard sensors.

Simultaneous localization and mapping (SLAM) is a useful technology in low speed applications, and when the vehicle has no prior knowledge of the environment it's operating in. Since the Speedway's layout does not change every lap, and we will be driving at high speeds, trying to construct a map of the track while racing is both unnecessary and unsafe. The competition should provide HD maps of the speedway. Then the vehicle will just focus on localization: figuring out where it is on the track/map.

If we are going to drive at 120 MPH, it's critical that our driving pipeline has minimal delays. As shown in the figure above, the car first has to make a best guess on where it currently is before it can decide where to go. When racing at high speeds, a GPS-only system is not sufficient due to its low frequency and errors in the urban areas, so it would make that guess based on a combination of GNSS, Inertial Measurement Unit (IMU), odometry, and perhaps also lidar data. In general, we will deploy a Kalman filter frame to estimate the states of the ego vehicle. GNSS uses satellite signals to obtain position, time, and velocity. IMU directly measures the acceleration of the vehicle for all 6 axes of motion (x, y, z, yaw, pitch, roll). Although IMU and odometry update rates can reach up to 1000Hz, their measurements accumulate error. As time goes on, these errors get larger and the measurements drift away from the true position. By fusing IMU, odometry, and GNSS readings, the errors can be mitigated. However, the update rate of GNSS tend to be much slower than that of IMU, at about 20Hz. Therefore, we use IMU data, steering angle and throttle to make predictions of the states in the short future, and IMU errors are allowed to accumulate between each GNSS reading by an acceptable amount. By comparing the Laser scans from lidar of the current location with the HD-map using the iterative closest point algorithm (ICP), it can serve as the odometry to get the position and heading of the ego vehicle. Combining position and velocity data from GNSS, and position and heading data from lidar, we then feed them into the update step to complete the localization process. Assuming we can reliably obtain our position at 20Hz, at full speed, the vehicle would have traveled about a full car length between readings. Even if this update rate is higher, we will still have to account for such a delay in our planning and control models, which will be discussed later in this paper.

The perception module also incorporates the capability of detecting and recognizing obstacles and signal lights. Given input LiDAR, RADAR, and camera inputs, the obstacle submodule detects, segments, classifies and tracks obstacles in the ROI that is defined by the high-resolution (HD) map. The submodule also predicts obstacle motion and position information (e.g., heading and velocity). The signal submodule detects signals (flags) and recognizes their status in the images.

Obstacle perception includes LiDAR-based and RADAR-based perception, and fusion of both results. The LiDAR-based obstacle perception, based on the Fully Convolutional Deep Neural Network, predicts obstacle properties such as the foreground probability, the offset displacement w.r.t. object center and the object class probability. Then it implements object segmentation based on these attributes. The RADAR-based obstacle perception is designed to process the initial RADAR data. In general, it extends the track ID, removes noise, builds obstacle results and filters the results by ROI. The obstacle results fusion is designed to fuse the LiDAR and RADAR obstacle results. In general, it manages and associates obstacle results from different sensors, and integrates obstacle velocity by Kalman Filter.

The goal of cognition module is to predict other vehicles' intention, which is required by the planning module. First, we segment the forward facing image into drivable and non-drivable areas. Based on the historical trajectory of other vehicles, we can train a network to predict its future trajectory and intentions, like overtaking or making a defense. If V2X data is available, which provides motion and/or control states of other vehicles, we can get even better performance.

For navigating the racecar, an optimal racing line can be generated before arriving at the track by solving an optimization problem, using track geometries and vehicle parameters as constraints. The optimization problem is typically formulated with minimizing lap time as the goal. Considering the vehicle's acceleration limits in both lateral and longitudinal directions, a path and a speed profile along that path is generated.

Of course, we cannot rely solely on this path in a head to head race, where other cars can be in our way and simply reducing speed and following behind means we'll never win the race. There has to be a path planning module running onboard that's constantly replanning to avoid obstacles and find new fastest paths. This path planner will use obstacle locations, predicted future trajectories, and drivable area fed from other modules to decide on a new path. Since this can also be solved as an optimization problem, we can either generate a new path and have a separate control module that modulates steering, gas, and brake to track that path, or we can solve for the optimal driving actions directly with the goal of minimizing time and risk of contact with obstacles with a technique called model predictive control (MPC).

MPC is a type of controller that predicts the system's future trajectory, then adjusts inputs to the system such that the predicted future trajectory lines up with a given desired future trajectory. In our case, the controller will predict the racecar's future path, and solve for a trajectory of future driving inputs (such as steering angle, throttle percentage, and brake pressure) for the car to track a path given by the planning module. When the input trajectory is calculated, only the first time step from that trajectory is actually sent to the vehicle. The entire process happens over and over again as obstacles move and reference trajectory changes. The reasoning behind using only the first time step in the input trajectory is that the prediction model is a simplified version of the physical system and is not perfectly accurate. As the prediction is propagated further in time, we have less and less certainly that it's representative of what the system is actually doing, therefore only the first predicted input is used. As the combustion engine, gearbox, and brake system in these Indy Lights cars have their own complex dynamics, directly predicting throttle and brake pressure would make this an incredibly difficult problem to solve. A more realistic control structure is to use the MPC to predict steering angle and longitudinal acceleration, and have a lower level controller that manages gear shifts, throttle, and braking based on that longitudinal acceleration. We believe MPC is crucial to building a winning autonomous racing system.

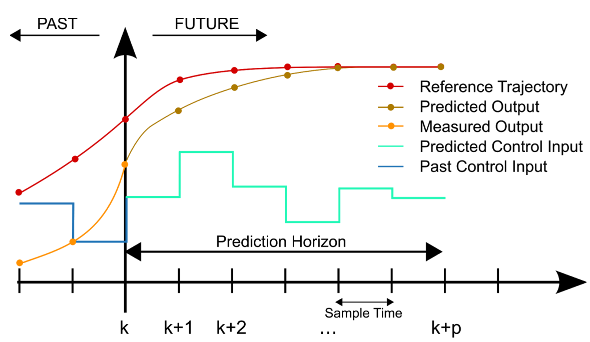

Figure 7: Input and state trajectory plot illustrating operation of a model predictive controller

Project Management

The University of Waterloo has a campus culture that revolves around building innovative technologies. As autonomous driving could be one of the most disruptive technologies of the new decade, Waterloo students will most certainly want to be apart of the team. This has been proven by Watonomous, which receives hundreds of applications at every recruitment round. The new Indy Autonomous Challenge team will be attractive to an even broader range of students, from computer science majors who are interested in the various areas of autonomous driving software, to mechanical engineering majors who just want to build the fastest car, to business students who would love the opportunity to market and attract sponsorship for an incredibly exciting initiative.

Through building the 100+ member Watonomous team, we've learned a great deal about managing a large group of students and we will be applying those lessons in building our Indy Autonomous Challenge team. Watonomous student leadership is among Waterloo's best and our management practices are recommended to all student teams by the university. We also hire 1-3 full time co-op students per term and the experience in managing full time employees will be useful for the Indy team as well.

Similar to Watonomous, there will be 4 divisons: software, mechanical, electrical, and business. A cross divisional drive crew will form the backbone of the team. A lesson learned from Watonomous was that having a multi-level management structure was not conducive to teamwork, so we will trial a flatter hierarchy. We will take full advantage of project management tools such as Jira, Confluence, and GSuite to track and prioritize tasks. We'll use the objectives and key results (OKR) framework to define long term goals and success metrics.

Table 1: Tentative Timeline

Time	Task
January 2020 - May 2020	Submit video of our vehicles operating autonomously
May 2020 - June 2020	Familiarize with the ANSYS simulator, conduct a literature review on how we can solve the simulation round. Define goal for September (that is when summer term students leave for co-op).
June 2020 - September 2020	Work towards September goal. Try to get the simulated car driving reliably.
September 2020 - January 2021	Build with the intention of porting our work to a real car later on. Work towards a performant driving algorithm and have it ready by January.
January 2021 - May 2021	Search for a partnering school with facilities to run the car, get ready to put code onto the car.
May 2021 - July 2021	Work on getting the car running on our code.
July 2021 - October 2021	Get the car to drive safe and fast.

Fundraising

Due to the multidisciplinary nature of the challenge, we can approach sponsorship from many different angles. There are plenty of engineering related companies that could benefit by collaborating with us. Having their logo on our car, apparel, social media, and website is great for brand awareness since Waterloo is one the top engineering universities in Canada. Another benefit they'll receive by sponsoring us is a fantastic recruitment source. If sponsorship is in the form of materials, parts, or expertise, it makes hiring a breeze as they will have already worked with potential hires in a technical capacity. We will offer tiers of sponsorship, with increasing levels of benefits at each tier. These could be more prominent logo placement, social media shoutouts, job posting advertisements to our members, and direct access to resumes. We can also host tours of the team facilities.

Non technical organizations will be approached with the same tier system, but we might have to describe the project from another perspective. For certain organizations, we can present ourselves as students participating in an engineering competition for the purpose of education. We can also choose to highlight the excitement of racing, or the safety and environmental benefits of bringing self driving cars to our roads. According to the National Renewable Energy Laboratory, they have the potential to cut energy consumption by 90% by reducing the number of crashes and driving in a more fuel efficient manner.

Aside from external sponsorships, the University of Waterloo has sources of funding for student projects as well. The Waterloo Engineering Society (EngSoc) invites student design teams to apply for funding 3 times per school year, and more than $10,000 is distributed per round. The student funded Waterloo Engineering Endowment Fund (WEEF) provides funding to any engineering student, staff, and faculty. WEEF focuses on funding projects that benefit undergraduate education, which the Indy Challenge certainly does, as it prepares the next generation of engineers for the rapidly changing automotive industry. There is also the Mathematics Endowment Fund (MEF), which supports projects that improve the undergraduate experience for Math and CS students. We will be eligible for MEF as long as we have CS students onboard.

Conclusion

This white paper outlines The University of Waterloo's rich experience and expertise with autonomous vehicles, including relevant professors and current vehicle programs, and presents our proposed technical approach to automating an Indy Lights car to race around the Indianapolis Motor Speedway. The project timeline and potential sources of funding were also discussed. Waterloo is excited about the coming months and is looking forward to shaping the future of racing, and continuing the tradition of using motorsports innovations to improve conditions of ordinary motorists.

References

1. https://www.novatel.com/an-introduction-to-gnss/chapter-6-gnss-ins/gnss-ins-systems/
2. https://www.hagerty.com/articles-videos/articles/2019/05/15/how-to-sponsor-two-largest-racing-series
3. https://www.nrel.gov/docs/fy13osti/59210.pdf
4. https://arxiv.org/abs/1902.00606
5. https://www.theglobeandmail.com/drive/technology/article-canada-leads-the-way-to-an-autonomous-future/
6. https://news.ontario.ca/mto/en/2016/11/automated-vehicles-coming-to-ontario-roads.html?_ga=1.106594051.1033784844.1479484884
7. https://driving.ca/lincoln/mkz/auto-news/news/u-of-waterloos-self-driving-cars-have-logged-100-km-on-public-roads
8. https://www.cbc.ca/news/canada/kitchener-waterloo/self-driving-car-university-waterloo-winter-driving-1.5459331
9. https://www.autonomoose.net/news
10. https://uwaterloo.ca/stories/waterloo-amazes-amazing-race-canada
11. https://www.theglobeandmail.com/news/national/two-canadian-universities-selected-for-north-american-self-driving-car-competition/article34699641/
12. https://driving.ca/lincoln/features/feature-story/motor-mouth-university-of-waterloo-is-at-the-forefront-of-autonomous-driving