CSCI 4302: Autonomous $\frac{1}{10}$th-Scale Vehicle Competition*

Sumeet Batra, Cameron Casby, Ryan Ibarra, Nelson Mitchell, and Marissa Tracy

University of Colorado Boulder, Boulder CO 80309, USA

Abstract. The team developed a fully autonomous, one-tenth scale vehicle capable of navigating a racetrack and overcoming several challenges, such as object detection, tracking, and avoidance. The team integrated IR sensors, a camera, an IMU, etc. to create sensor fusion algorithms that helped locally map and navigate the environment. The team used techniques from computer vision and control theory to help steer through the environment at relatively high speeds without crashing into walls, obstacles, or other such hazards. The team created custom parts to mount the hardware onto our car's frame. We used Robotics Operating System (ROS) to integrate all of our software for various hardware into a single framework.

Keywords: Proportional – Integral – Derivative · Computer Vision · Autonomous · Robot · Vehicle.

https://github.com/nelsonnn/odroideka.git

* Supported by University of Colorado Boulder - Computer Science Department

1 Introduction

1.1 The Goal

The goal of our autonomous vehicle is for the vehicle to be able to traverse a course consisting of two right turns as quickly and reliably as possibly. The goal is to make a vehicle that is both highly reliable and capable of navigating the course without fatal collisions or termination states. Furthermore, our team chose the extra challenge of avoiding a rolling blue ball while c9om

1.2 The Approach

To create an autonomous vehicle that is both reliable and capable, the team decided to implement a switch controller. This allows the vehicle to rely on data from the less reliable IR sensors during straightaways and rely on more accurate local data from the IMU sensor during turns. This also allowed the team to have varying speed depending on what stage of the course the vehicle is in. To minimize the course time, the team was able to speed up during the straightaways and slow down to take the turns.

2 Methodology

2.1 Mechanical Processes

The mechanical design in this project, while not obviously critical, is still very relevant. The placement of sensors and controllers is very important because, if they are not located in the correct place, or not located where the team thinks they are located, it can lead to issues that are not simply addressable by software (or potentially much more difficult to address with a software change, compared to a hardware change). The team attempted to address these challenges from the start to avoid issues down the road.

To begin with, the baseplate was designed to accommodate all of the necessary modules to permit this $\frac{1}{10}$th-scale car to operate autonomously and on offshore power. The baseplate, depicted in Figure 1, secures the ODROID XU-4, oCam, both infrared sensors, PhidgetSpatial IMU, USB Hub, Pololu Servo Controller, and the voltage regulator. All of these components are critical to the functionality of this car and require proper positioning to ensure reliable data gathering.

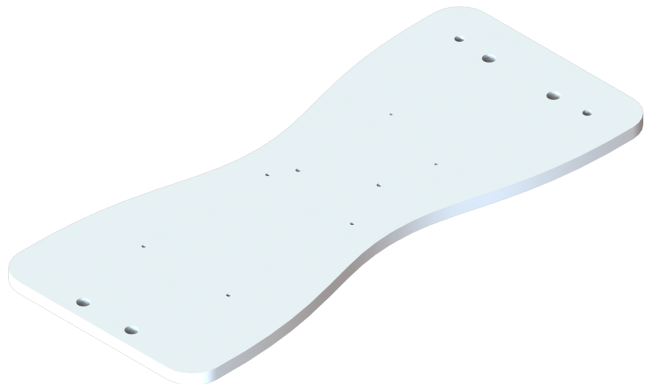


Fig. 1. Baseplate

The baseplate was designed to take up as little space as possible. Given that it was necessary to secure a large suite of sensors, controller, and various other modules, the baseplate was designed to mount these units on both faces. This improved the amount of usable surface area, as well as minimized the footprint.

The initial version of the baseplate, which was used to complete the midterm, was a laser cut piece of wood that only had locating holes to secure the baseplate to the vehicle. This gave the team the ability to map out the locations of the various modules, drill holes very easily, and then modify the contact points as necessary. This was an important step for the team because it permitted slight modifications without the need for remaking an entire part or parts. Once the locations of the individual components were solidified, the team purchased a 1/4" sheet of acrylic and laser cut it, including the locating holes this time. This new baseplate was then taken to the machine shop and all relevant holes were tapped. This allows the units, mentioned above, to secure directly to the baseplate, yielding a cleaner assembly overall.

Another important step in the mechanical side of this project is the development of adapters to secure several sensors to the baseplate. Based on the documentation for the Adafruit IR Distance Sensors (GP2Y0A710K0F), the team realized that these needed to be mounted vertically. This created an interesting problem because the mounting holes are now located in a plane that is unreachable by the baseplate. To solve this, an adapter was designed (see Figures 2 and 3) to secure the IR sensors to the baseplate, while maintaining several degrees of freedom that would aid in fine-tune adjustments. These sensors can rotate about their mounting points in two planes, allowing the team to change the angle of the sensors in the vertical and horizontal directions (relative to the baseplate).

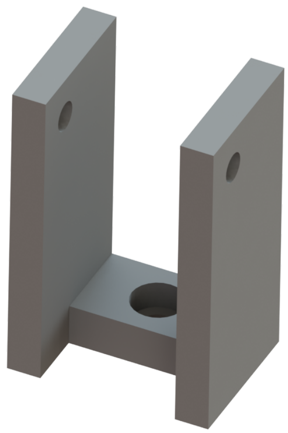


Fig. 2. Infrared Sensor Mount

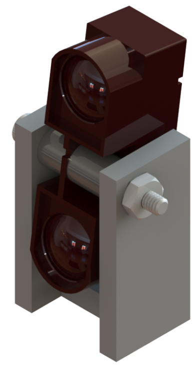


Fig. 3. Infrared Sensor Assembly

Finally, the oCam also required a unique mounting solution. There are no mounting points with easy access, relative to the baseplate. To accommodate this, another adapter was designed that secures the oCam to the baseplate in a secure manner. As seen in Figure 4, the slot on the base allows the unit to slide forward and backward on the baseplate, giving the oCam a variable standoff distance from the edge of the baseplate. This was important because, should the car run into a wall head-on, the oCam needs to be protected by the bumper of the car. Also, the team was unsure about the oCam's angle of view in relation to the IR sensors, so some modularity was added.

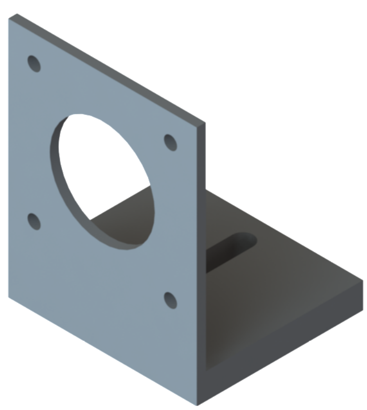


Fig. 4. oCam Mount

To fabricate these adapters, they were 3D printed using a Fused Deposition Modeling (FDM) printer. This manufacturing process provided the necessary structure and dimensional accuracy to mount the sensors to the baseplate in a reliable manner.

Once this baseplate is fully assembled, it mounts directly to the car and is ready to be hooked up to the various modules to permit full electrical integration. Figures 5 and 6 demonstrate the fully-assembled configuration.

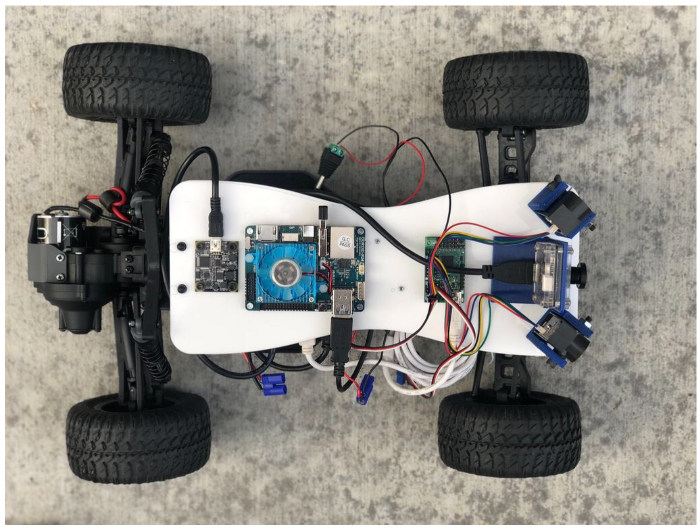


Fig. 5. Assembly 1

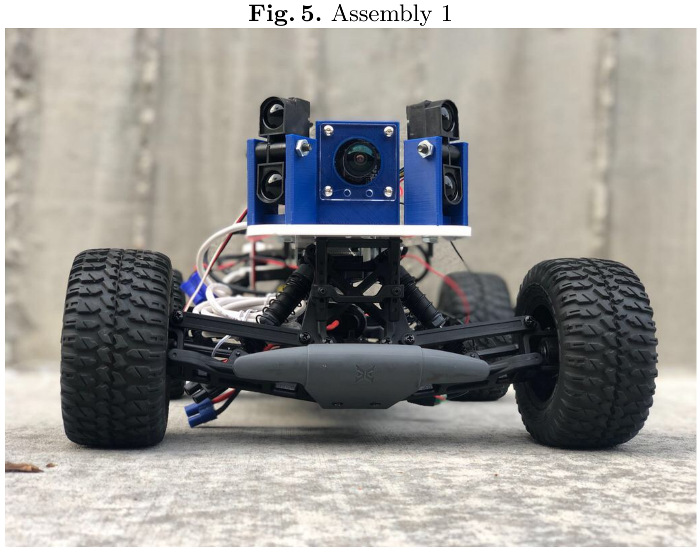


Fig. 6. Assembly 2

2.2 Sensing

Any given reading from an IR sensor is noisy and unreliable. Therefore, the team keeps a rolling average of sensor readings, which can then be sent to the controller. The team also decided to angle the IR sensors at 45% with respect to the front of the car (see Figure 6). This ensured that the distance readings always decreased when the car steered toward a wall within the constraints of the track. Another benefit of this choice was that the vehicle was able to recognize the turn from farther away, and therefore be able to drive faster before and through the turn.

The orientation data that comes from the Phidget IMU also has a tendency to drift easily, so the team used a madgwick filter to obtain more accurate data. However, due to magnetic interference from the motor near the IMU, the team only used the accelerometer and gyroscope data in the filter. Because of its unreliability to track the orientation of the IMU in the long-term, the team only used data from the IMU for a short term, to orient the robot through the turns on the course.

2.3 Controller

For the purpose of deciding the controls of the vehicle, the team implemented an abstract state machine which was comprised of two PID controllers. The two states developed were the "straightaway" and the "turn" state.

When the vehicle is launched, it is initialized to the straightaway state. This state uses the distance estimations from the two IR sensors to get distance estimates from the left and right walls in the corridor. The PID controller then estimates the width of the hallway using the median of a stored number of estimates taken during the run. Then, the error term is computed to be in the interval $[-1, 1]$ which represents the percent distance away from the desired position relative to the walls. The desired position was set to have about 60% of the hallway on the right of the vehicle and 40% to the left of the vehicle. This created a better approach angle for the car to make a right turn more reliably.

The straightaway controller also contains many optimization's that keep the car moving straight and prevent it from making dramatic turns. This is achieved by ignoring the data that would suggest bad sensor readings. Particularly, if the derivative of the distance to the wall with respect to time passes a threshold, we do not decide any new controls for the vehicle in that control loop. Also, if the left IR sensor has a distance reading that is greater than the width of the corridor, it defaults to the right IR sensor to calculate the error term.

The IR sensors were also used to detect the time for the vehicle to turn during the straightaway state. This was decided by detecting a right distance estimate and distance derivative that lie in an interval that was empirically determined to represent a turn. When the straightaway controller detected these values, it passed control to the turn controller.

The second PID controller was developed to handle the sharp turns of the course at high speed. It uses the error between the current orientation around the

Z-axis (upward direction) of the car and a desired orientation to make steering decisions and a constant forward velocity. For the turns on the course, the desired orientation was taken to be the initial orientation of the vehicle when control returns to the controller, plus a fraction of $\pi/2$. This was to avoid over-steering, but may have been avoidable with better-tuned gains. The controller would return control to the straightaway state when the error of the current orientation was sufficiently small. We found this controller to be more precise than the other due to the high short-term accuracy of the IMU data.

3 Results

After three consecutive successful runs on the course, the team recorded a best lap time of 26.5 seconds and an average time of 27.2 seconds.

4 Discussion

The team successfully implemented an autonomous $\frac{1}{10}$th-scale vehicle that traversed the designated course in a reasonable time period. The implementation of a proportional – integral – derivative controller allowed the vehicle to navigate the course with ease. The vehicle did not oscillate through the straight hallways, which demonstrated the robustness of our controller and improved the time performance since it took a more direct path. The 3D printed components allowed all of the components to be securely mounted onto the vehicle. The customized base plate allowed the team to find the optimal positioning for all of the components. Finally, the vehicle would slow down when going through the turns to ensure control was maintained. These aspects, among others, ensured the reliability and functionality of the vehicle.

5 Conclusion

Overall, the vehicle came together really well. The mechanical components provided us with the necessary structure to develop upon. The sensors, filters, and controller permitted the team to manage the way in which the vehicle moved through the course. In addition, all of the state transitions also occurred smoothly. These features yielded a functional, efficient vehicle.

One possible improvement that could be made to the vehicle would be to add functions that make the controllers more robust to battery charge. During testing, it was necessary to switch out the motor and steering battery when it was depleted to about half of the charge. This is because the lower battery charge would change the dynamics of the car, and the turn detection would no longer be as robust. To allow the car to run longer off of the same batteries, the team could use a voltmeter to detect battery charge and tune the controller values for intervals of battery charge. Thus, the vehicle would be more efficient in battery use and therefore more practical.

Another improvement that could be employed in this project is to introduce more controllers to the state machine. An example would be to introduce a "turn planning" state that optimizes the vehicles pose going into the turn. This could greatly decrease the car's lap time on the course as the team believes this component of the race has the most room for further optimization in light of the fact that it is fairly easy to drive down the straightaway at high speed. This improvement also relies on better turn detection techniques so there is sufficient time to run a turn planning state. Although better hardware can allow for more complicated functionality, it also in turn complicates the implementation. From the hardware already available for the project, computer vision using the oCam could have improved the turn detection. This data could be the basis for a turn planning controller to prepare turns to be as quick as possible.

Bibliography

[1] A. Mokhtarzadeh, A., Jian Yangqing, Z.: Human-robot interaction and self-driving cars safety integration of dispositif networks *. pp. 494–499 (08 2018). https://doi.org/10.1109/IISR.2018.8535696

[2] Almalioglu, Y., Saputra, M.R.U., de Gusmao, P.P.B., Markham, A., Trigoni, N.: GANVO: unsupervised deep monocular visual odometry and depth estimation with generative adversarial networks. CoRR **abs/1809.05786** (2018), `http://arxiv.org/abs/1809.05786`

[3] Hayes, B., Shah, J.A.: Improving robot controller transparency through autonomous policy explanation. In: Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction. pp. 303–312. HRI '17, ACM, New York, NY, USA (2017). https://doi.org/10.1145/2909824.3020233, `http://doi.acm.org/10.1145/2909824.3020233`

6 Appendix

6.1 A. Human-Robot Interaction with Autonomous Vehicles

Human path planning is incredibly complex and draws from several different factors, including vision, sounds, social queues, language, etc. Human Robot Interaction (HRI) aims to create autonomous robotic systems that not only aid humans with day-to-day tasks that range from mundane to complex, but to do so in a manner such that the robot is 'socially aware' and can understand social queues from humans as priors into its own path planning or task execution algorithms.

One example of this vein of research is socially aware path planning, where mobile robotic systems, such as the Kinova Movo, learn to navigate safely in an environment with humans. Traditional path planning algorithms and approaches fail to consider how the generated paths interact with or change an environment. For example, a robot that uses any vanilla shortest paths algorithm in a room with humans moving about will potentially generate an acceptable path that avoids collisions with humans and other objects in the environment, but may do so in a way that hinders or contradicts with the goals of other people in the environment, which can lead to frustration. Socially aware path planning attempts to alleviate this by learning behavior models for humans that act as priors for predicting trajectories or semantic goals of surrounding people. For example, if a human's heading and trajectory line up with a coffee machine on the other side of the room, the robot should be able to use its learned model of human behavior as a prior to predict that the human is most likely to follow a trajectory that will lead him/her to the coffee machine. In doing so, the robot agent can develop trajectory predictions for surrounding humans, and perform its own path planning in a way that is the least obstructive to other people.

Autonomous vehicles on the road are another interesting point for investigation into human-robot interactions. For fully autonomous vehicles to be successfully implemented, robots will need to learn to adapt to the way that humans drive. Often times, the decisions that humans make on the road are not logical to robots. It is important for artificial intelligence to be generalized enough that it can understand and adapt to many different situations. Currently, artificial intelligence is able to understand how to perform certain tasks, such as maintaining a steady speed, self parking, and staying within the bounds of a lane. Current research and development into autonomous vehicles aims to improve the robustness path planning algorithms to work in expected conditions, where lanes are clearly defined, cars pass through an intersection on a green light, and set speed limits are followed. However, what does the autonomous vehicle do in the situation where, for example, a person suddenly jumps in front of the car?

Humans have acquired many bad habits on the road. From quick lane changes to speeding through red lights, human behavior on the road is very unpredictable. The sporadic behavior of humans driving makes it hard for a robot to share the road. Unpredictable human behavior is one of the factors that prevents level five autonomy. Before autonomous cars can reach level five autonomy, robots and the artificial intelligence behind them need to be able to learn to adapt to any situation they are faced with. Currently, most autonomous systems are able to sense and adapt to other internal attributes. One example of such a system that is built into cars is cruise control. Basic cruise control only relies on a signal from the vehicle to tell the system to either accelerate or ease up on the gas. Since this is a completely internal system, the car is able to do it all automatically. When the car needs to perform tasks that involve external input, the situation becomes a lot more challenging. Driving requires continuous evaluation of the surrounding environment, as well as always thinking about what to do in an unforeseen event. For an autonomous system to perform these tasks, the system needs to understand all of the parameters that affect how the car should drive. Some parameters can be easily handled through sensors, such as road conditions or the speed of the car and surrounding cars. However, there are many unexpected events while driving, such as a car suddenly cutting in front of another car, a child running after a ball that rolls in front of the car, or everyone slamming on the brakes due to an object falling off a car on the highway. An autonomous vehicle needs to be able to adjust its path based off of multiple input parameters.

One of the issues with the current sensors and sensor data is the way that it is being processed. The human mind is not able to fully grasp how the control logic in the machine learning algorithms work. Currently, the artificial intelligence behind autonomous vehicles is trained with machine learning frameworks and artificial networks [1]. While this method allows the artificial intelligence to continuously improve upon itself, it can make it hard for humans to understand what is happening within the network. These understanding gaps must be closed to ensure that autonomous vehicles will perform reliably and safely. In addition, it is important to fully understand the system so that humans know why the system makes the decisions that it makes and so that the parameters can be tuned appropriately to get the desired results. It is important that the human behind the wheel is able to understand the vehicles perception of its surroundings so that the human can complement the vehicles awareness with their own.

Safety is the top priority when designing fully-autonomous vehicles. When designing a robot, it is very important to consider how the robot will be controlled, plan paths, and make predictions. Improving the robots controller and effectively communicating what is happening in the control is one way to improve the safety and reliability of autonomous vehicles. The current sensors used for robotics range from cameras, to LIDAR sensors, or regular RADAR sensors. By improving these sensors and how the data is used, full autonomy becomes more achievable. The vehicle must be trained to handle different situations. For example, one situation it needs to be able to handle is what to do if a person appears in front of the vehicle. The vehicle must be aware of its surroundings so that the vehicle knows the best action to take. For example, if the vehicle is surrounded by traffic, the vehicle must be able to make a decision that will minimize harm. In addition, humans must consider how the artificial intelligence was trained. If the artificial intelligence was trained to make protecting the people inside the vehicle its top priority, humans need to be aware of that and how that can affect the driving performance.

The understanding of how the autonomous vehicle was trained is very important as it will impact the way the humans interact with autonomous vehicles. When an autonomous vehicle is designed to protect the person inside of it, the car is destined to face some very hard-to-make decisions. A common moral issue with autonomous vehicles is what decision does the vehicle make when both options are not ideal. For example, if an autonomous vehicle is heading towards an accident and needs to get out of the way, but there is a family to the right of the car and an elderly person to the left of the car, which way does the vehicle swerve. In this scenario, the autonomous vehicle must choose to either hit the family, the elderly person, or allow the person driving the vehicle to drive into the accident. This decision is not easy for a human to make, nonetheless an algorithm. It is important that humans understand the algorithms behind the decisions that autonomous vehicles make so that situations like this can be avoided.

Currently, the artificial intelligence behind autonomous vehicles is not enough to achieve level-five autonomy. Humans need to understand that algorithms behind autonomous driving are not sufficient enough to make all of the right decisions while driving and that the person behind the wheel still needs to be present while the vehicle is driving itself. For this reason, it is critical that humans are able to interact with autonomous vehicles and understand the algorithms and controls behind the system. Humans must be able to interact with autonomous vehicles and systems in order to have a successful implementation of level-five autonomous vehicles.

6.2 B. Deep Learning to Improve Performance

For many of the challenges presented in this competition, traditional computer vision methods are not only well suited for tackling said challenges, but also require much less computational resources, a problem that commonly plagues deep learning. However, there are many complex perception and path-planning related problems that deep learning can solve well, where a traditional computer vision system would fail. For perception and localization, YOLOv3, a well known Convolutional Neural Network (CNN) architecture, can perform multi-class object detection and camera relative 2D pose estimation in real time on constrained hardware. YOLO takes inspiration from Fast-RCNN, a CNN that using convolutional layers in tandem with region proposal layers that use anchors to predict bounding boxes for different objects in an image. However, Fast-RCNN uses a sliding window to make predictions, which makes it unfeasible for real-time object detection. YOLO performs singleshot predictions using a similar anchor strategy, which allows for realtime multi-class detection. This potentially allows autonomous vehicles utilizing YOLOv3 to detect a vast and varied number of objects in a scene, as well as localize against detected stationary objects. If computational constraints are still a concern, the neural network can be trained and run offline by streaming images off the autonomous vehicle to another, more powerful computer, that then returns class predictions, bounding boxes, and 2D pose back to the vehicle. Finally, transfer learning can be used to detect custom objects that may not be part of the data set the network was initially trained on. By freezing most of the layers in the network, and training a fully connected layer with a new, custom data set containing the desired objects to detect, an existing architecture like YOLO can be trained to detect a broader range of objects, which autonomous vehicles can utilize to have a richer understanding of its surroundings with visual data alone.

For real world applications involving autonomous vehicles, visual data is often noisy, and objects humans want to detect are often occluded. For example, on a busy highway, cars in their respective lanes will be partially or fully occluded by trucks, other cars, etc. Here, using a region proposal network will lead to poor performance, since the anchors in region proposal layers cannot reliably handle occluded objects. In this case, networks that perform semantic segmentation such as Mask-RCNN and Region-based Fully Convolutional Networks (FCNET), are much more suitable for autonomous vehicle perception. Both Mask-RCNN and FCNET perform pixel-wise classification, which means each pixel is classified as either one of the predefined object classes, or as background. This technique is robust against partially, or even fully occluded objects in a cluttered scene. This architecture can also be used to understand 3D or even 6D pose plus orientation information, which can further assist an autonomous vehicle in its decision making process. For example, networks such as Mask-RCNN can be used to understand pose information of pedestrians, which gives insight into their heading and trajectory when considered over a time interval.

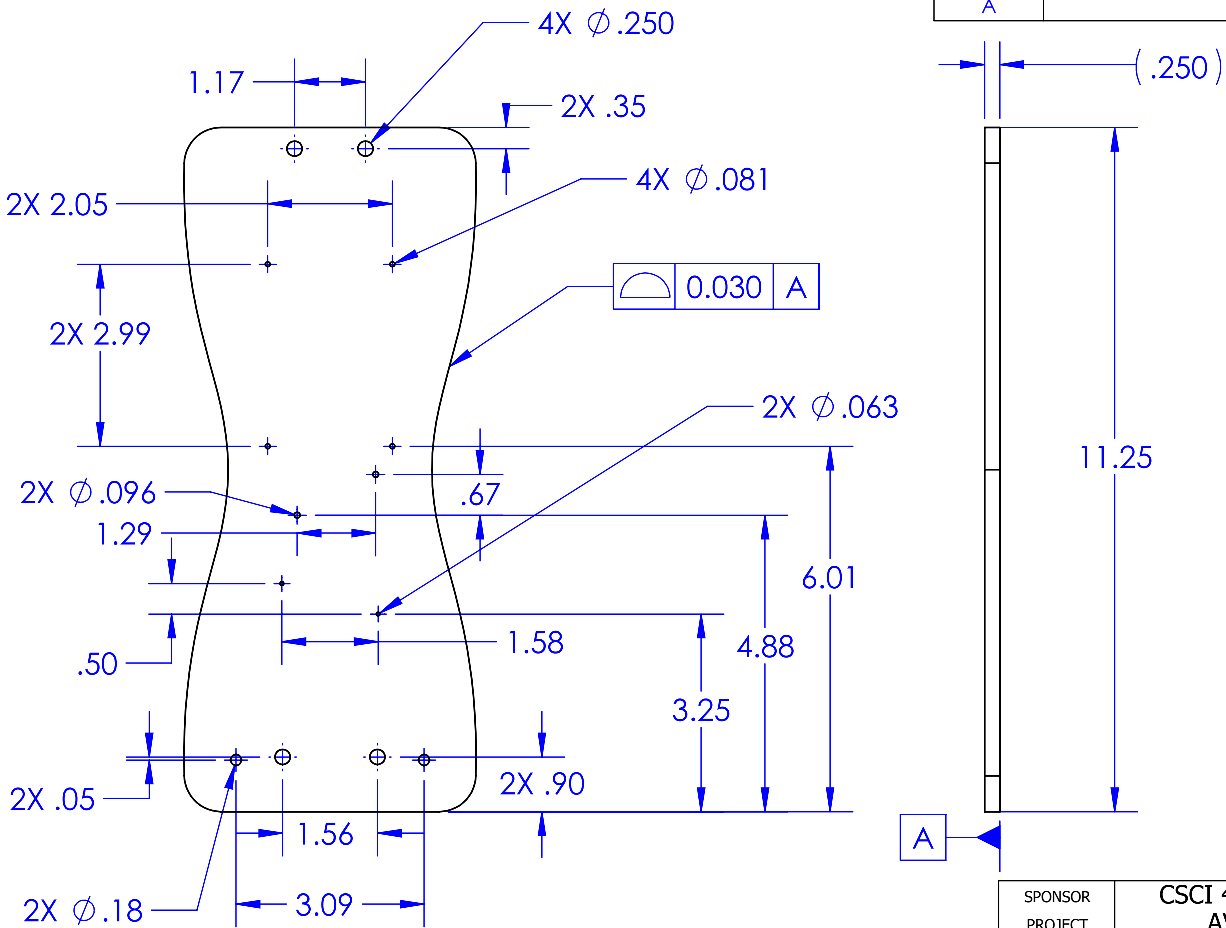
In regards to deep learning for localization and mapping, there have been many attempts to use supervised deep learning methods to do Visual SLAM; however, robotics applications involving deep learning often suffer from a lack of labeled data, including the Visual SLAM task. Using Generative Adversarial Networks (GAN) alleviates the lack of data problem and has been proven to be useful for tackling the Visual SLAM task[2]. In this approach, raw RGB images are fed into system. The generator network attempts to create realistic depth maps from the RGB images that are indistinguishable from real depth maps. A discriminator network attempts to classify fake depth maps created by the generator from the real depth maps. This back and forth between the discriminator and generator forces the generator to learn latent variables that are responsible for creating realistic depth maps. In this way, autonomous vehicles can utilize this GAN architecture to do Visual SLAM without the need for labeled data.

Finally, the task of autonomous navigation is still a difficult challenge, since it involves a continuous state space with lots of uncertainty, due to many moving objects, such as other cars and pedestrians, in a cluttered scene. Prior attempts to tackle this challenge involve simplifying the state space into a computationally tractable discrete space, making simplifying assumptions, and then employing some type of learning or planning algorithm. Here, Deep Reinforcement Learning (Deep RL) can be used to solve the navigation task with respect to more complex action-state spaces. While traditional RL involves more hand engineering of certain aspects of the problem, such as the cost/reward function, observations, transition probabilities, etc., Deep Reinforcement learning makes the assumption that these latent variables can be learned via a neural network, and a suitable policy can be found with said learned variables.

6.3 C. Drawings

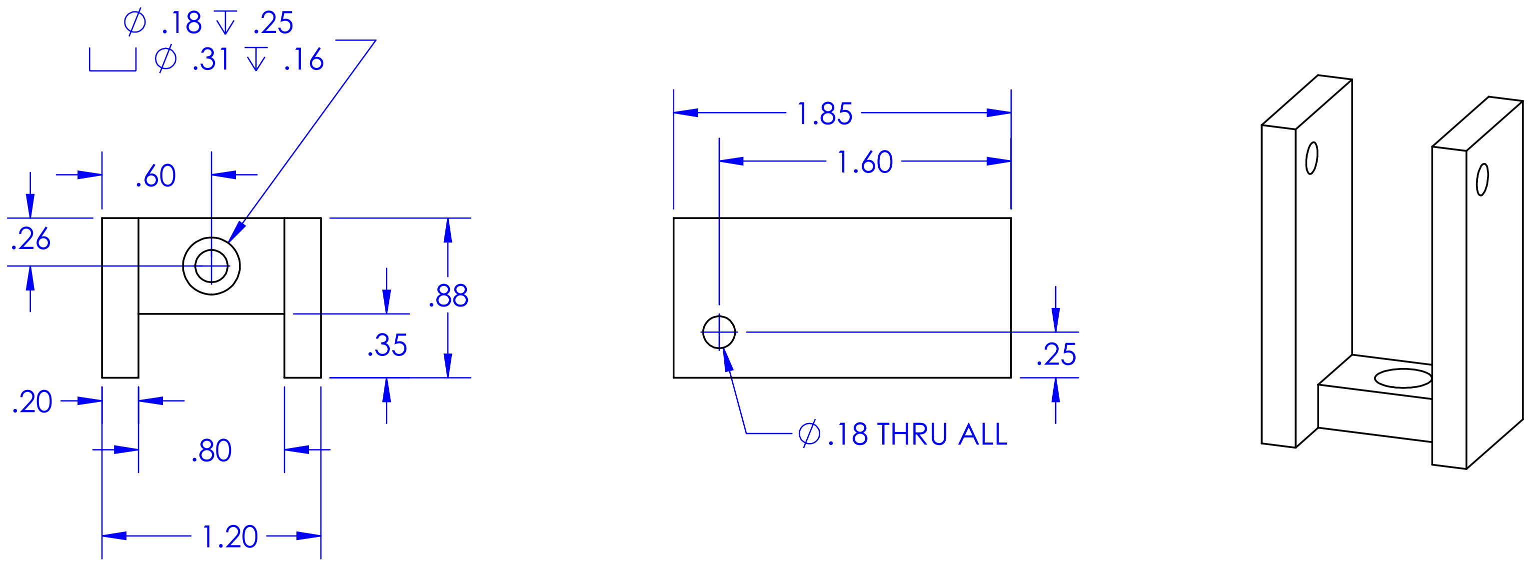
The following pages contain 2D orthographic drawings for the mechanical components of the baseplate assembly. Note that drawings of standard fasteners (such as those available on McMaster-Carr) are not included because they are easily accessed via third-party vendors.

REV.	DESCRIPTION	DATE
A	INITIAL RELEASE	4/20/2019



SPONSOR PROJECT TEAM NUMBER	CSCI 4302 AVC ODROIDEKA	DESIGN CENTER COLORADO
TOLERANCES UNLESS NOTED: UNITS: INCHES	X.X ±0.050 X.XX ±0.010 X.XXX ±0.005 X.X° ±0.5°	FILE NAME RobotBaseplate APPROVAL
DRAWN BY CAMERON CASBY	DRAWN DATE 3/20/2019	MFD BY
MATERIAL ACRYLIC		DESCRIPTION BASEPLATE
STOCK SIZE 1/4 SHEET		PN N/A · REV A · SHEET 1 of 1

REV.	DESCRIPTION	DATE
A	INITIAL RELEASE	5/4/2019



SPONSOR	CSCI 4302
PROJECT	AVC
TEAM NUMBER	ODROIDEKA

DESIGN CENTER COLORADO

TOLERANCES UNLESS NOTED:

X.X	±0.050
X.XX	±0.010
X.XXX	±0.005
X.X°	±0.5°

UNITS: INCHES

FILE NAME
IR_Mount

APPROVAL

DRAWN BY
CAMERON CASBY

DRAWN DATE
4/17/2019

MFD BY

MATERIAL
PLA

DESCRIPTION
IR MOUNT

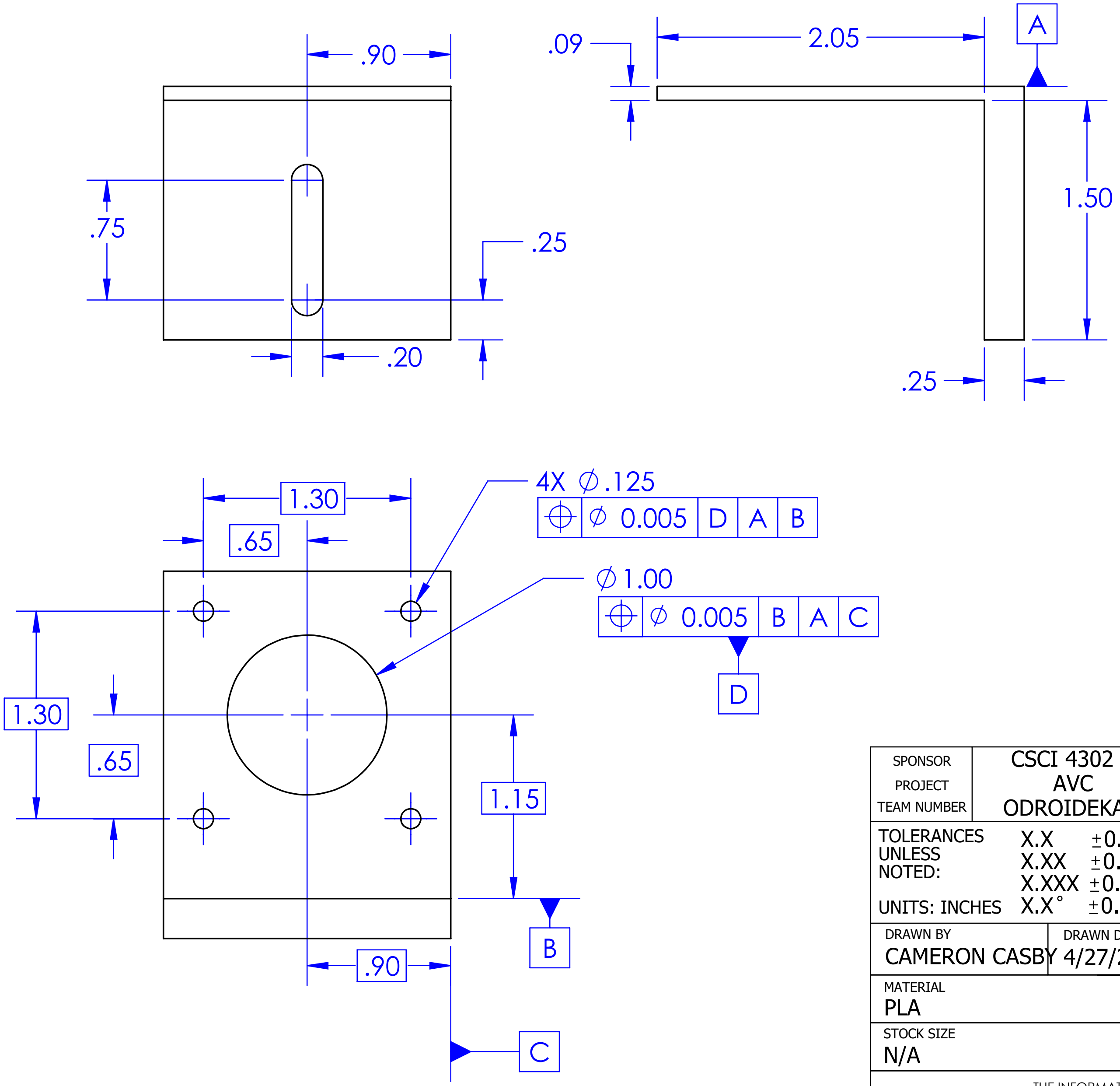
STOCK SIZE
N/A

PN
N/A

REV
A

SHEET
1 of 1

REV.	DESCRIPTION	DATE
A	INITIAL RELEASE	5/4/2019



SPONSOR PROJECT TEAM NUMBER	CSCI 4302 AVC ODROIDEKA
TOLERANCES UNLESS NOTED:	X.X ±0.050 X.XX ±0.010 X.XXX ±0.005 X.X° ±0.5°
UNITS: INCHES	

DESIGN CENTER COLORADO

FILE NAME: oCam_Mount

APPROVAL

DRAWN BY: CAMERON CASBY

DRAWN DATE: 4/27/2019

MFD BY

MATERIAL: PLA

DESCRIPTION: OCAM MOUNT

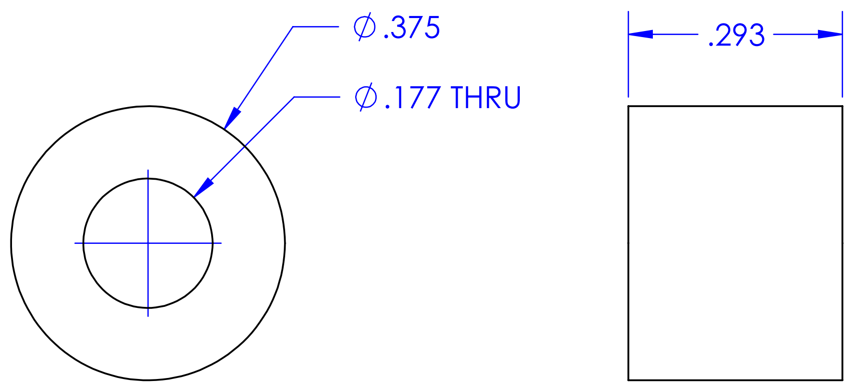
STOCK SIZE: N/A

PN: N/A

REV: A

SHEET: 1 of 1

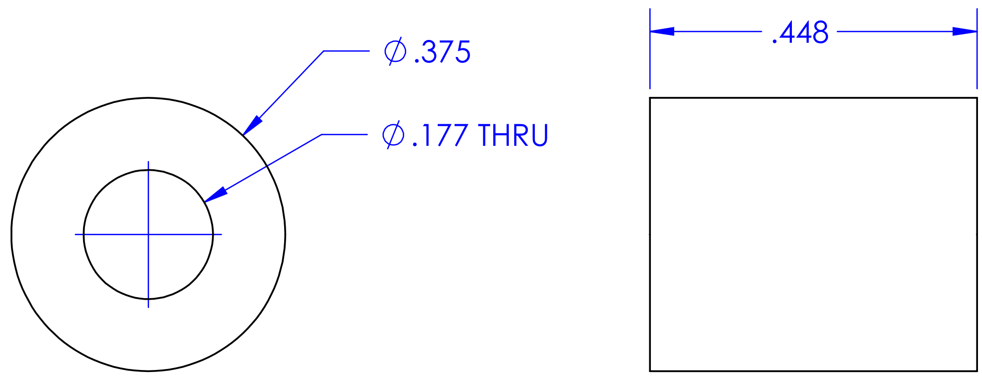
REV.	DESCRIPTION	DATE
A	INITIAL RELEASE	4/9/2019



SPONSOR	CSCI 4302	DESIGN CENTER COLORADO	
PROJECT	AVC		
TEAM NUMBER	ODROIDEKA		
TOLERANCES UNLESS NOTED:	X.X ±0.050 X.XX ±0.010 X.XXX ±0.005 X.X° ±0.5°	FILE NAME Spacer1	
UNITS: INCHES		APPROVAL	
DRAWN BY CAMERON CASBY	DRAWN DATE 4/9/2019	MFD BY	
MATERIAL DELRIN		DESCRIPTION SPACER 1	
STOCK SIZE 3/8" ROUND		PN N/A	REV A
		SHEET 1 of 1	

PROPRIETARY AND CONFIDENTIAL

REV.	DESCRIPTION	DATE
A	INITIAL RELEASE	4/9/2019



SPONSOR	CSCI 4302	DESIGN CENTER COLORADO			
PROJECT	AVC				
TEAM NUMBER	ODROIDEKA				
TOLERANCES UNLESS NOTED: UNITS: INCHES	X.X ±0.050 X.XX ±0.010 X.XXX ±0.005 X.X° ±0.5°	FILE NAME Spacer2			
		APPROVAL			
DRAWN BY CAMERON CASBY	DRAWN DATE 4/9/2019	MFD BY			
MATERIAL DELRIN		DESCRIPTION SPACER 2			
STOCK SIZE 3/8" ROUND		PN N/A	REV A		SHEET 1 of 1