**Friday, December 8th, 2023**

I continued training the reinforcement learning algorithm in simulation using the TQC algorithm. It was able to converge to a policy which could slowly get around the track after about 30 minutes, and it managed to beat the lap time I trained with PPO after about an hour of training (it averaged about 16 seconds per lap, while the PPO which I trained overnight wasn’t able to improve past 18 seconds). I did tweak the action space slightly: the steering values are limited from (-0.5, 0.5) instead of (-1, 1), because extreme steering values are not really necessary for driving successfully. Constraining the action space definitely helped the algorithm converge to a better solution faster.

While the training was happening, I charged the battery for the car because I realized I hadn’t charged it since I started testing out the motors. To ensure that the battery was charged safely, I used the flame-retardant bag.

I then started planning out how I will train the auto-encoder. First, I will need to collect manual training data – essentially, I will drive the car around myself with a game controller and capture an observation consisting of [(steering, throttle) 🡪 (image)]. Then I will be able to train an autoencoder that boils down the image to a more compressed feature vector. After that, I’ll change the reinforcement learning so that it is given this feature vector rather than the entire picture, which will make the process much more sample-efficient.

**Monday, December 11th, 2023**

Since today’s a short period I continued working with the camera as it is integral to my project. On Wednesday I will take the time to set-up the autoencoder in simulator (or the parts will come and I can do some car assembly).

I had already established programmatic Python control of the RGB camera, so today I checked to make sure the on-device processing was working. I did this by using the camera’s built-in IMU combined with the stereo depth map to create an estimate of the motion of the car using a visual odometry Python library:

A graph of a diagram

Description automatically generated with medium confidenceThe camera was successfully able to localize the car. I further increased the accuracy of the localization by using an AprilTag I had printed a few weeks ago. Since the size and orientation of the AprilTag are known, it constrains the uncertainty of the camera’s position estimate. I may end up using these when doing the reinforcement learning in my implementation of reset(), which will need to reset the car’s position to the starting line regardless of where it currently is.

**Wednesday, December 13th, 2023**

I was not present because I had a doctor’s appointment.

Before ML on Wednesday I talked with Dr. Torbert who said the VESC motor controller will arrive in January. Almost all of the smaller parts have already arrived.