Lab 3 Report: Wall Follower

Team 11

Luis De Anda
Suchitha Channapatna
Amber Lien
Prince Patel

6.4200

March 15, 2025

1 Introduction

Autonomous navigation in complex environments is an essential skill for robots that requires robust perception and control systems. Many autonomous vehicles use LiDAR sensors because they provide a stream of environmental data, enabling the detection of obstacles and the estimation of their distances. This lab explores the challenges of processing raw LiDAR data to enable effective wall-following behavior on a robotic mini racecar.

Building on our simulated wall-following algorithms, this lab focuses on deploying and refining a wall-following controller on the physical racecar. There are two primary goals: first, to accurately parse LiDAR data to extract relevant environmental information, and second, to use this information to guide the racecar along a wall at a desired distance and velocity. Additionally, a safety controller will be implemented to prevent collisions with unexpected obstacles, a new problem not seen in simulation. The motivation behind this lab is to bridge the gap between simulation and real-world robotic systems, addressing the complexities of operating an autonomous vehicle in a dynamic environment. This work contributes to the broader understanding of robust control strategies for autonomous navigation, highlighting the importance of sensor data processing and real-time decision-making in robotics.

2 Technical Approach

As described above, the goal for this lab is two-fold: to follow a wall successfully while having a comprehensive safety mechanism to prevent collisions.

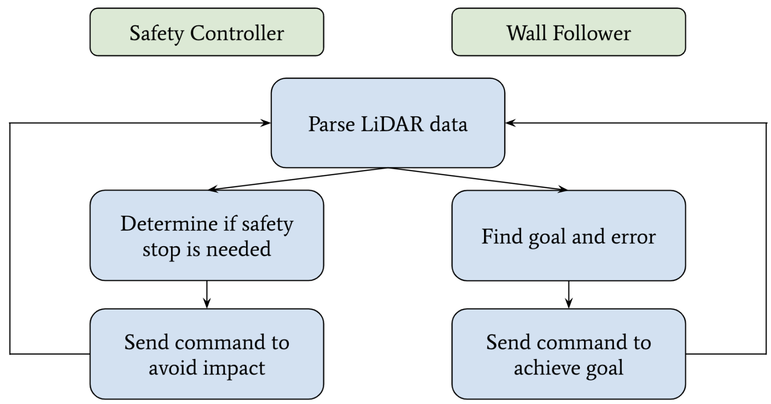


Figure 1: **A block diagram depicting the steps taken to complete the lab.** We parse LiDAR data, using the information to stop or calculate our projected trajectory.

We divided this problem into subproblems, tackling each independently (a choice reflected in our source code) as shown in Figure 1.

To achieve both goals (each implemented as its own module), we use an exteroceptive sensor, LiDAR. Both modules run simultaneously and continuously, limited only by the rate of measurements from the LiDAR. This architecture is supported by the existence of a mux which establishes the hierarchy of commands for our controller. Safety commands are considered higher priority than wall-following drive commands, allowing the safety controller to override the wall follower when needed.

2.1 Safety Controller

The primary goal of the safety controller is to avoid impacts with obstacles. Notably, this should not impact the accuracy or performance of wall following. Overall, this module determines the appropriate obstacle search area, determines if an obstacle is too close, and accordingly sends a stop command to the racecar.

Based on the racecar's dynamics, it will only hit obstacles head-on. We are therefore able to reduce the obstacle search area. We define lookahead distance to be the distance d_l between the LiDAR frame's origin and the furthest point (in cartesian absolute distance) we will consider in our search for poten-

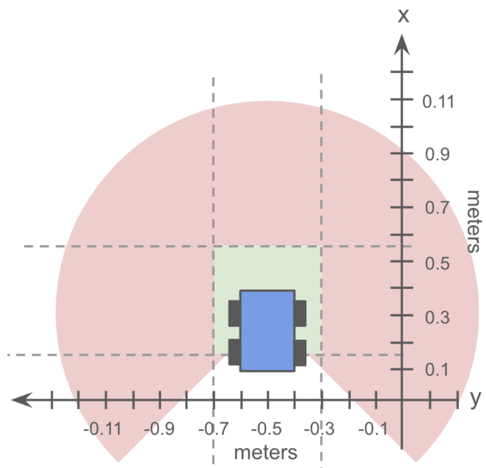


Figure 2: **The angle ranges considered in calculations in the safety controller.** The red region denotes possible ranges from which LiDAR data can be obtained. The green region denotes the portion used in our calculations for the safety controller. Units are in meters.

tial obstacles. We recognize that a faster moving racecar should attempt to break sooner than a slow moving racecar when in danger of colliding with an obstacle. For that reason, within our safety controller module, d_l is defined as $max(d_{l\_base}, d_{l\_base} \cdot v)$ where d_{l_base} is a user defined baseline lookahead distance and v is the current speed of the racecar. d_{l_base} is a tunable parameter, and we have chosen 1.5 meters as a reasonable baseline. By taking the maximum, we ensure a large enough lookahead distance for both velocities less than 1 and greater velocities more typical of racing conditions. Figure 2 depicts the range of LiDAR data considered.

Another important feature of our safety controller is the buffer distance. The buffer distance is defined as the distance from an obstacle at which the racecar should stop. We experimented with making this buffer linearly proportional to velocity, but ultimately found that was too conservative for high velocities. Instead, similar to the lookahead distance, we define a baseline buffer and increase the buffer based on velocity. The formula is:

$$d_b = d_{b\_base} + (\lfloor v \rfloor - 1) \cdot 0.05 \tag{1}$$

We follow the below steps to determine conditions in which we must safely stop.

1. Convert the LiDAR point cloud data from polar to Cartesian coordinates in (x, y).

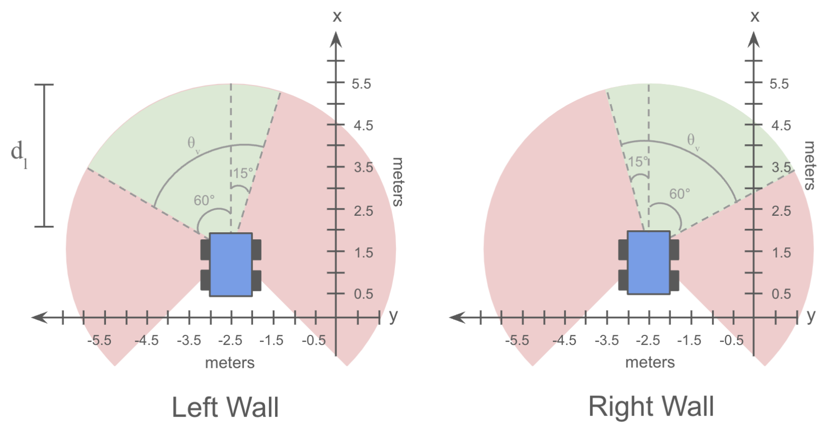


Figure 3: **The laser ranges considered when estimating the wall for the wall follower.** The red section depicts the range from which LiDAR data can be obtained and the green section depicts the range of information we use to estimate where the wall is located. Note the difference in the ranges used when following the left wall versus the right. Units are in meters.

2. Filter out all data that is more than 0.2 meters to the left or right of the racecar (in the y-axis).
3. Filter out all data that is farther than the lookahead distance d_l or behind the racecar (in the x-axis).
4. Check if the minimum point is closer than the buffer distance d_b. If so, send a stop command.

2.2 Wall Follower

The primary goal of the wall follower is to allow the racecar to keep at some fixed distance from the wall (either on its left or right side) while traveling at a desired velocity.

We define the following terms used in our wall follower:

x = actual distance from wall
D = desired distance from wall
e = error between desired distance and actual distance
θ = current heading

Based on Lab 2, we aim to keep a fixed distance from the wall using a PD controller. Here, we employ two separate PD controllers. One minimizes distance error between the current racecar position and desired distance from the wall. The second minimizes angle error between the heading of the racecar and the position of the wall, ensuring that the racecar always attempts to drive parallel to the wall.

The wall follower has eight parameters. One parameter, S, is defined beforehand to inform the racecar of which wall to follow. $S = \{1, -1\}$ where 1 is left and -1 is right. The next three are used to determine the racecar's trajectory by estimating the wall's location: lookahead distance, front distance, and the view angle for parsing LiDAR data. The last four are tuned for optimal wall following: K_p^d, K_d^d, K_p^a, and K_d^a.

The view angle θ_v defines the range to the left and right of the LiDAR frame's origin considered in wall calculations. Note that the origin is defined as right in front of the racecar (x-axis) and positive angles are in the counterclockwise direction.

If $S = 1$, $\theta_v = [-15°, 60°]$.
If $S = -1$, $\theta_v = [-60°, 15°]$.

Figure 3 depicts this asymmetry.

As with the safety controller, the lookahead distance is the distance d_l between the LiDAR frame's origin and the furthest point we'll consider in estimating our wall. The front distance d_f defines the distance at which the racecar will begin to more heavily weight the points in the front, preparing the vehicle to take a turn at a closed corner. The point at which the racecar prepares to turn is dependent on the desired distance from the wall. For this reason, we consider the racecar to be in "turning preparation" when $d_f \leq 2 * D$.

We prepare for a turn by manipulating the racecar's perception of the wall, creating a sloped line that encourages our racecar to turn as depicted in Figure 4. We first parse LiDAR data as follows.

If $S = 1$, then consider points in the range $[-15°, 5°]$.
If $S = -1$, then consider points in the range $[-5°, 15°]$.

We then weigh these points more heavily in forming our line of best fit by adding them again to our initial points of consideration. This skews the line towards the wall in front of the racecar, encouraging a steeper turn.

In a normal, non-turn scenario, these extra points are weighted equally and the line of best fit is formed using points gathered from the LiDAR data in the range of θ_v. This data is used to create a linear matrix equation with the following terms:

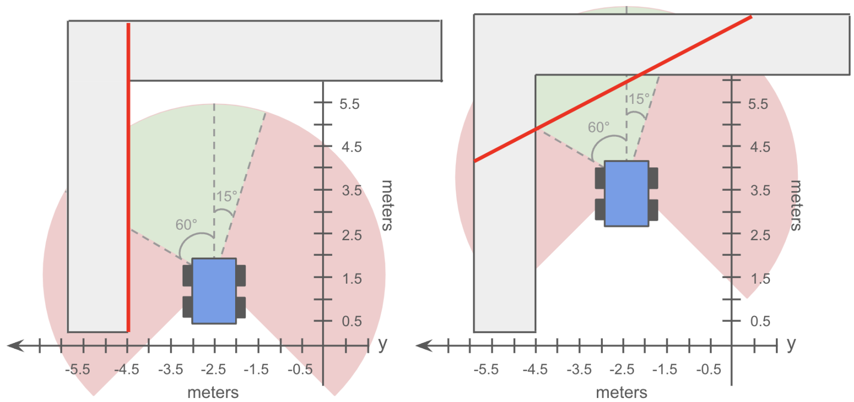


Figure 4: **The difference in line estimation when traveling along an open wall and traveling towards a closed corner.** As the distance to a wall in front of the vehicle decreases, LiDAR data points in front of the vehicle are more heavily weighted, resulting in a more sloped line encouraging a turn. Units are in meters.

$$A = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \quad B = \begin{bmatrix} y_1 \\ y_1 \\ \vdots \\ y_n \end{bmatrix} \tag{2}$$

We solve for the least square solution of $Ax = B$ to obtain our line of best fit with slope m and intercept b. From this line, we compute the current distance from the wall x and error e:

$$e = |b|/\sqrt{m^2 + 1} \tag{3}$$

We additionally calculate the angle to the wall, defined as the arctangent of the slope of the line of best fit:

$$\theta = \arctan(m) \tag{4}$$

We compute derivative terms as the difference between the current and previous state, assuming small time steps between each call of the callback function.

The overall control input angle is a combination of both PD controllers (while accounting for side) defined by the following equation:

$$\text{steering} = -S \cdot K_p^d \cdot e + K_d^d \cdot \frac{\text{de}}{\text{d}t} + K_p^a \cdot \theta + K_d^a \cdot \frac{\text{d}\theta}{\text{d}t} \tag{5}$$

We followed a standardized procedure for tuning these parameters. We tuned both PD controllers independently. We kept K_d^d at 0.0 and increased K_p^d until steady oscillations were visible. We then increased K_d^d until oscillations were damped to a smooth line. The same process was used for the angle controller before combining and making adjustments based on the behavior observed.

3 Experimental Evaluation

We evaluated the performance of our safety controller and wall follower with a combination of qualitative and quantitative metrics in a variety of situations. For qualitative analysis, we recorded the racecar as it followed a wall and stopped in front of obstacles. For quantitative analysis, we used rosbags to record messages on topics of interest. We then extracted the data to form plots and statistics.

3.1 Safety Controller

We ran repeated trials to determine how many times the safety stop would successfully override drive commands in a dangerous scenario. The racecar was commanded to drive at a fixed speed without a controller.

To test a fixed obstacle setting, we had the racecar drive towards a wall. The racecar is expected to stop 0.25 seconds from a collision. Over 10 trials, we obtained the results in Table 1. Note that in the case of collisions, we considered the distance to the wall to be 0.0 meters.

Speed	Times Stopped	Average Distance
1.0 m/s	10	0.24 m
1.5 m/s	9	0.22 m
2.0 m/s	8	0.28 m

Table 1: Number of times the racecar successfully stopped during 10 trials and the average distance at which it stopped is recorded in the above table. A successful stop is defined as not touching the wall.

To test a dynamic obstacle setting, we began to walk in front of the racecar as it was 0.5 meters, 1.0 meter, and 1.5 meters away for the racecar driving at 1.0 m/s, 1.5 m/s, and 2.0 m/s respectively. The safety controller should override the drive commands when the racecar is 0.25 seconds from a predicted collision. This corresponds to a distance of 0.25 meters, 0.375 meters, and 0.5 meters for a velocity of 1.0 m/s, 1.5 ms/, and 2.0 m/s respectively.

We obtained the following in Table 2 over ten trials. Note again that the distance in collisions was considered to be 0.0 meters.

Speed	Times Stopped	Average Distance
1.0 m/s	8	0.20 m
1.5 m/s	6	0.18 m
2.0 m/s	7	0.13 m

Table 2: Number of times the racecar successfully stopped during 10 trials and the average distance at which it stopped is recorded in the above table. A successful stop is defined as not touching the person walking in front of the racecar.

The safety controller exhibited fewer successes in the dynamic obstacle setting compared to the static due to the narrow range in which we considered LiDAR data.

3.2 Wall Follower

In order to maintain consistency between tests while evaluating the wall follower, all of the trials involved following a wall and turning a corner between two fixed points. We varied the speed and side of wall followed during these tests. We use the following metrics to quantitatively determine the wall follower's performance.

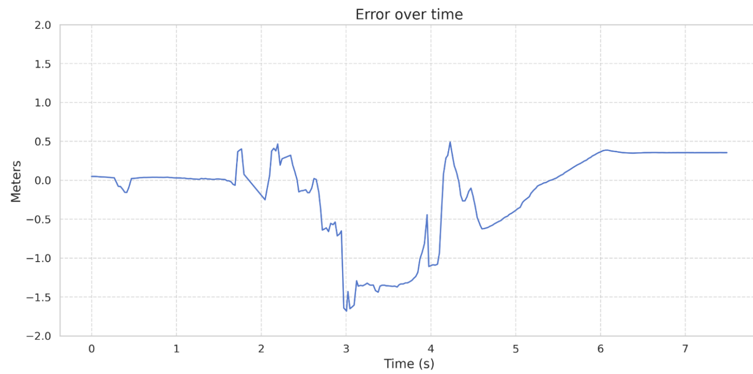
x = actual distance from wall
D = desired distance from wall
e = error = $|D - x|$
a = accuracy = $\max(0, 1 - \frac{e}{D}) \cdot 100$

3.2.1 PD Controller versus Double PD Controller

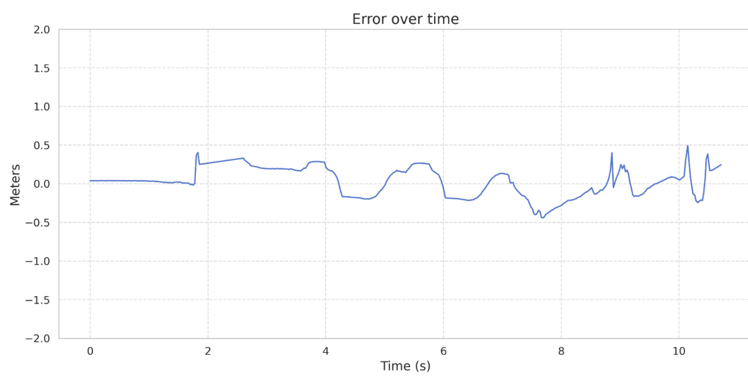
A portion of our analysis also includes a comparison between using a single PD controller to reduce distance error and using our double PD controller to reduce both distance and angle error, results for which are depicted in Figure 5. Note the increased variance in graph (a) compared to graph (b), indicating that using a double PD controller resulted in better performance by reducing oscillatory behavior. The reduced total distance error and increased accuracy in the double PD case validates our choice to control both distance and angle.

3.2.2 Wall Follower Performance

Figure 6 contains plots displaying the error computed over time for each of our scenarios. All trials took place in the same location. We repeated the process



(a)



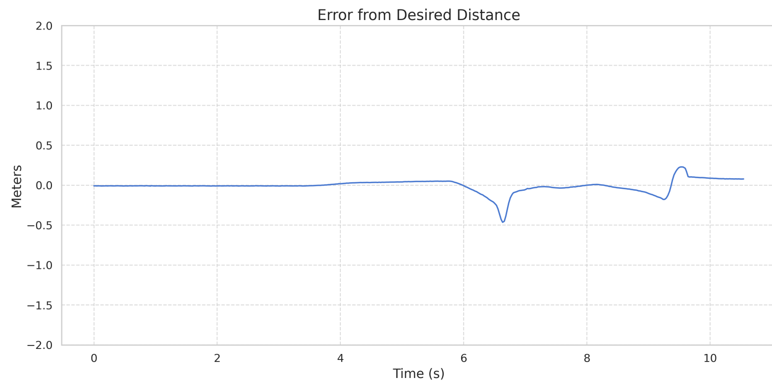
(b)

Figure 5: **A comparison between the error over time of the racecar's deviation from the desired path using a PD controller with distance error and a PD controller with distance error and angle error.** Graph (a) depicts the performance of the single PD controller. Graph (b) shows the performance of the double PD controller. The smaller variance of graph (b) demonstrates the superior performance of the double PD controller compared to the single.

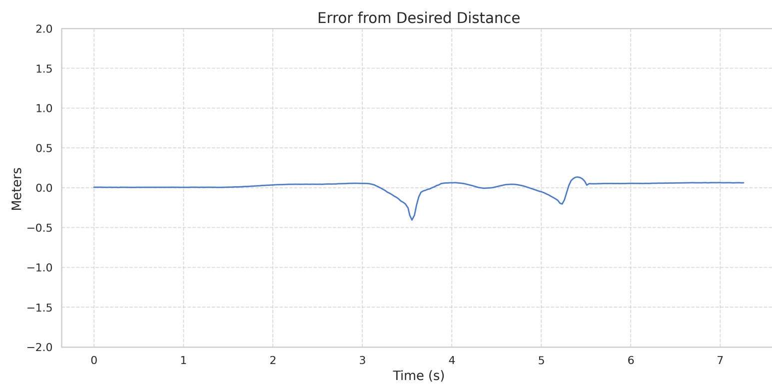
for the right wall as well. The resulting statistics are summarized in Table 3 and Table 4.

Overall, the deviation from our desired path remains near zero, demonstrating the robustness of our wall follower. However, we notice increased error for subplot c, which corresponds to a velocity of 2.0. This may be because the racecar rolls forward when we stop running the wall follower, causing other obstacles to come within the lookahead distance. We would resolve this in the future by finding a more isolated test environment.

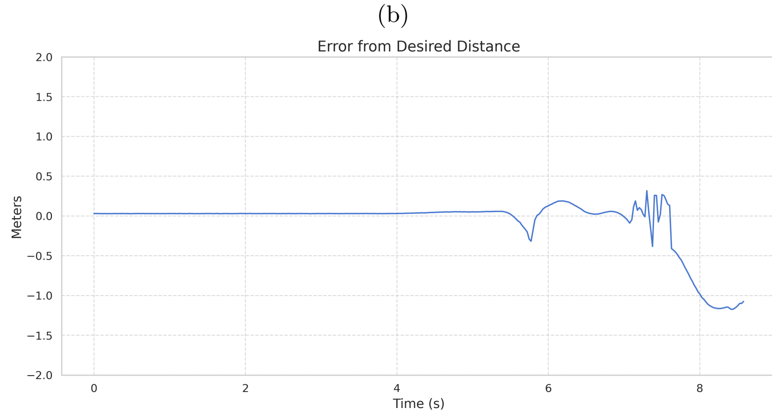
We notice that the mean accuracy when following a right wall is less than the mean accuracy when following a left wall. While we are not completely sure as to the cause, we suspect that it was a result of the testing environment. There was a dip in the wall that was close to the beginning of the run when following on the right side as opposed to at the end of the run when following on the left side. If the racecar was not already driving with close to zero heading, the dip may have impacted the performance negatively. If we were to improve the controller, we would ensure such dips have a smaller impact on the accuracy of the controller.



(a)



(b)



(c)

Figure 6: **The figures depict the racecar's error from the desired path while moving at three different velocities, 1.0 m/s, 1.5 m/s, and 2.0 m/s, following a wall on its left side.** Overall, the error remained close to 0.0 meters, with deviations occurring during turns in which our metric for determining error did not as accurately reflect the ground truth.

Metric	**Fast Left**	**Medium Left**	**Slow Left**
Ground Truth Statistics			
Mean Distance from Wall (m)	0.462	0.484	0.508
Std. Dev. of Distance (m)	0.094	0.068	0.081
Min Distance (m)	0.320	0.366	0.270
Max Distance (m)	1.045	0.905	0.964
Error Statistics			
Mean Absolute Error (m)	0.042	0.047	0.048
Max Absolute Error (m)	0.438	0.405	0.464
Std. Dev. of Error (m)	0.071	0.068	0.081
Steering Statistics			
Variance of Steering Angle	0.024	0.022	0.018
Mean Steering Angle	-0.132	-0.106	-0.040
Accuracy Metrics			
Average Accuracy (%)	89.64	90.53	90.48

Table 3: Left Wall Following Performance for Different Speeds

Metric	Fast Right	Medium Right	Slow Right
Ground Truth Statistics			
Mean Distance from Wall (m)	0.494	0.476	0.484
Std. Dev. of Distance (m)	0.185	0.172	0.179
Min Distance (m)	0.140	0.182	0.211
Max Distance (m)	1.938	0.969	1.112
Error Statistics			
Mean Absolute Error (m)	0.071	0.072	0.063
Max Absolute Error (m)	1.038	0.469	0.862
Std. Dev. of Error (m)	0.105	0.112	0.089
Steering Statistics			
Variance of Steering Angle	0.049	0.037	0.034
Mean Steering Angle	-0.053	-0.095	-0.082
Accuracy Metrics			
Average Accuracy (%)	85.80	85.57	87.4

Table 4: Right Wall Following Performance for Different Speeds

4 Conclusion

We were able to meet the main objectives of this lab. We demonstrated using quantitative and qualitative methods that the racecar can follow a wall at a fixed distance, complete left and right turns, and avoid obstacles while moving in various environments at different speeds. Through the synchronization of the wall follower and safety controller, the racecar will always attempt to navigate around obstacles before stopping if the distance to obstacles gets too close.

Initial testing with the safety controller found that the vehicle continued to roll after the motor turned off, causing it to bump into walls and people moving in front. This issue scaled with the velocity of the racecar prior to stopping. This issue has since been resolved by updating the vesc settings, enabling reverse braking which successfully stopped the racecar in a shorter time frame.

In the immediate future, we will focus on reducing distance error during sharp turns. To better evaluate how well our control performs, we will also develop additional test metrics/environments that more closely reflect ground truth rather than relying on interpolated LiDAR data.

The progress made in this lab will be the foundation for more complex driving and navigation scenarios. Throughout future labs, we will make use of the safety controller to protect hardware when testing. We also expect to modify our wall follower to fit other control-specific tasks, such as line following.

5 Lessons Learned

Suchitha:
Over the course of this lab, I learned many important technical and CI skills. Technically, I became more comfortable with writing ROS2 nodes. I feel more confident when it comes to debugging, especially with using tools like rviz and rqt_graph. There were some hiccups in getting used to the racecar platform, particularly with pushing and pulling code. However, once we set up a github account on the racecar, syncing code between individuals and the racecar became much easier. There are a few aspects I would like to improve for the next lab. Most importantly, I think we need to be more organized with data collection and analysis methods. We initially ran into issues because we would collect rosbags but not remember which trials they represented. Having a more structured, consistent way to collect data would save us a lot of time. This lab gave me opportunities to practice communicating with my teammates, specifically to make sure we parallelized work. I would like to find additional ways in which we can boost efficiency and progress. Additionally, I want to adjust the style of slides used in future presentations based on the feedback and lessons of CI instructors.

Luis:
This lab was an eye opener in both technical and communication aspects. The switch from a simulated race racecar to a real world racecar created an impact much bigger than what I was expecting. Debugging in real life, having to account for more issues like battery life, proper connection of wires, and random noise that can pop up, was much more time consuming than changing a few lines of code. Although one expects this, experiencing it is still quite exhausting. Similarly, I wasn't expecting it to be so stressful presenting in front of a few people. I do typically get nervous before such an event but calm down during it, but this time all the stress seemed to bottle up in me. However, I did learn how to properly work with my team. We divided tasks, set aside time to work on it together, and did our best to get work done despite not always being able to all meet up in person. Also, based on feedback from our technical and CI instructors, I learned that I need to practice more being relaxed when presenting and how data should be formatted.

Prince:
This lab really solidified the sim-to-real gap for me because the same code/process from the previous lab could not be directly reused here. Additionally, I learned how to use and finetune a PD controller to achieve the wall following goal. I think in future labs I want to spend more time on data filtering and accounting for uncertainty in sensor data. On the team side, I think we slowly got better at parallelizing work but hope to see more progress on our efficiency so we can be more than the sum of our parts. On the CI-M side, I learned how to make slides that are direct and act supplementary to the presentation, not act like a mirror. For future weeks I definitely want to start working on the briefing/lab report earlier, perhaps even in parallel with the technical portions

of the lab. This way we know exactly what data we will need to back up our claims and won't have to go back and spend extra time on the lab.

Amber:
Lab 3 has emphasized the importance of written communication when working with a team; coordination through text was key for this lab as we found times to meet and delegated work between members who could make it to meetings and members who had to work remotely. Suchitha communicated frequently, establishing our meeting times and keeping us on track throughout the lab. Additionally, we collectively worked on the lab report and edited one another's sections, keeping others updated through comments. From the lab 3 briefings, I've also learned more clearly my weaknesses while presenting, needing to work to be more relaxed.

Additionally, preparation was necessary for efficient work. Our battery often died while we were working on the lab, requiring us to lose an hour of progress to charge it. We found charging it the night before helped mitigate the frequency of the issue, even if it wasn't entirely preventable. Moreover, having some visualization tool was beneficial, allowing us to debug more concretely. Before creating the tool, we made assumptions about what our logic was doing, unable to know if it was performing as expected and slowing down our progress as we made changes based on false assumptions. After Prince created a visualization tool graphing our estimated wall and angle range, our debugging process went much more smoothly. Finally, we need to be cautious when using logger statements to debug. Though typically seen as harmless, these statements impacted the speed at which our messages were being sent, directly affecting the performance of our vehicle. Luis and Suchitha found this issue while debugging, leading to a much higher performance for our racecar. Overall, I feel more comfortable with debugging and creating ROS2 nodes.