

# Lab #3 Report: Wall-Following on the Racecar

Team #5

Kelsey Fontenot

Yifan Kang

Dante Suazo

He'yun Zhi

6.4200 Robotics: Science & Systems

March 13, 2026

## 1 Introduction (Kelsey)

As autonomous vehicles are introduced into more new cities and become increasingly prevalent within our society, it is important that we as engineers learn and understand this technology so we can make it safer and more efficient. In this lab, we developed a safety controller, iterated on the previous lab's wall following algorithm, and implemented and evaluated both on the physical race car. These were the first steps to developing our autonomous system.

Some of the key goals of this lab were connecting to the hardware, reconciling the differences between simulation and real life on our controllers, and doing data analysis to improve the controllers. We needed to not only make our controllers robust but also evaluate their performance both quantitatively and qualitatively.

For our safety controller, we looked at the Light Detection and Ranging (LiDAR) data and stopped the racecar when it got too close to an object in front of it. For the wall follower, we designed a Proportional-Derivative controller to stay a certain distance from the wall that was estimated from the LiDAR data using a Random Sample Consensus (RANSAC) algorithm.

## 2 Technical Approach (He'yun)

Programmed by Yifan.

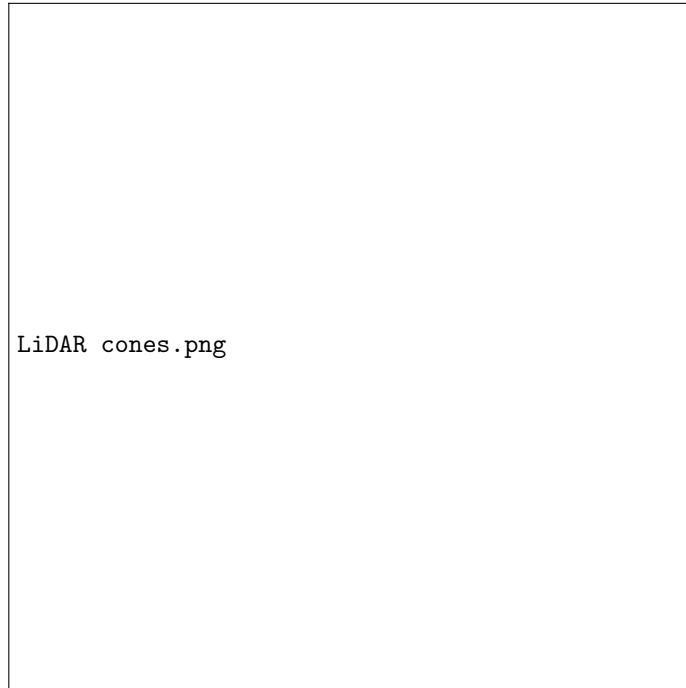


Figure 1: Ranges of LiDAR detection points used for our safety controller (red) and wall follower (green).

## 2.1 Safety Controller

Our safety controller considered the cone of LiDAR detection points from  $-\frac{\pi}{12}$  to  $+\frac{\pi}{12}$  and immediately stopped the car if the closest point was less than 0.5 meters away.

## 2.2 Wall Follower

Our wall follower considered one of two cones of LiDAR detection points depending on the given wall direction: one facing perpendicular towards the left wall and one facing perpendicular towards the right wall, both  $\frac{2\pi}{3}$  wide; thus, we considered either a left cone from  $+\frac{\pi}{6}$  to  $+\frac{5\pi}{6}$  or a right cone from  $-\frac{\pi}{6}$  to  $-\frac{5\pi}{6}$ . These cones are relatively wide to capture enough points for our RANSAC algorithm.

After translating this set of LiDAR points into Cartesian coordinates and excluding cases with fewer than three valid points, we ran 50 iterations of RANSAC to model the points to an estimated line relative to the car (set at the origin).

$$y = mx + b$$

$$d_{\text{measured}} = \frac{|b|}{\sqrt{m^2 + 1}}$$

To improve precision, our calculation of distance to our modeled wall incorporated a term representing the "look-ahead" distance to be traveled by the car in the time interval to the next loop of calculations.

$$d_{\text{measured}} += v \cdot \text{LOOK\_AHEAD\_TIME} \cdot \sin(\text{wall\_angle})$$

We inputted the error  $e(t)$  between our desired wall distance and our calculated current wall distance into a proportional-derivative controller to set our current steering angle, maximizing it at 0.34 radians.

$$e(t) = d_{\text{desired}} - d_{\text{measured}}$$

$$\text{steering\_angle} = \min \left( -\text{SIDE} \cdot \left( K_p \cdot e(t) + K_d \cdot \frac{e(t) - e(t-1)}{\Delta t} \right), 0.34 \right)$$

Our optimal  $K_p$  and  $K_d$  constants in the simulation, determined through trial-and-error, were

$$K_p = \frac{1}{v}, K_d = 0.5.$$

### 3 Experimental Evaluation (Dante)

We evaluated our controller first in simulation and then on the physical racecar. In simulation we mainly used the environment to verify that our wall estimate, PD controller, and safety logic behaved reasonably before testing on hardware. On the physical car we were more concerned with whether the car could hold the desired 1.0 m wall distance, how much steering effort was required, and whether the safety controller preserved clearance near obstacles and corners.

#### 3.1 Simulation

Simulation was primarily used for initial gain tuning and controller validation. We checked whether the wall estimate converged toward the commanded distance, whether the steering command remained bounded, and whether the car could recover from small disturbances. Through this process we selected  $K_p = 1/v$  and  $K_d = 0.5$  because they gave the best balance between responsiveness and overshoot in our simulated tests. However, simulation was more useful as a design tool than as a final predictor of performance because it did not capture the noisier LiDAR returns and corner cases that appeared in physical runs.

## 3.2 Real-Life experiments

We ran eight functional tests on the physical racecar: three right-wall runs, one left-wall run, one longer right-wall run, and three hallway runs. In each run the commanded wall distance was 1.0 m at a nominal speed of 1.0 m/s. The car used onboard LiDAR for both wall estimation and frontal obstacle detection, and each run began with the car aligned near the selected wall and ended when the course segment or test interval was complete.

From the logged wall-distance estimate, steering command, and speed command, we computed

$$\text{CTE}(t) = |d_{\text{measured}}(t) - 1.0|, \quad \text{Score} = \frac{1}{1 + (10\bar{e})^2}$$

where  $\bar{e}$  is the mean CTE for the run. We also recorded the percent of time within 30 cm and 15 cm of the target, mean steering rate, percent of time near steering saturation, signed tracking bias, oscillation rate, and minimum wall and frontal distances.

Across all eight runs, the controller achieved a mean CTE of 0.326 m, an average score of 12.3%, spent 70.1% of the run time within 30 cm of the desired distance, and maintained a mean speed of 0.93 m/s. Straight-line wall tracking was the controller’s strongest case. Right Wall Test 1 achieved 0.133 m mean CTE and spent 83% of the run within 30 cm of the target. Figure ?? shows that the right-wall error distributions are concentrated near zero, confirming tight tracking on continuous walls.

Longer steady runs also showed relatively smooth control effort. The long right-wall run had a mean steering rate of 0.077 rad/s and 7.1% saturation. Speed maintenance was strongest when the frontal obstacle logic was not intervening, and Right Wall Test 3 stayed at full speed 86% of the time.

The largest weakness was corner handling. In the hallway runs, the measured wall distance often jumped when the car approached an opening or turn. This produced temporary 1–3 m spikes in CTE before the controller recovered. Recovery typically took several seconds. Figure ?? shows this behavior across the hallway tests. Hallway Test 2, which was run faster at about 1.5 m/s, had 1.04 desired-distance crossings per second, which suggests that the same PD gains that worked acceptably near 1.0 m/s became underdamped at higher speed.

We also observed a consistent tracking bias. The right-wall runs tracked slightly too close to the wall, with signed errors between about -0.04 m and -0.09 m, while the hallway runs tracked too far away, with biases from +0.33 m to +0.43 m. In practice, this meant that the car often settled around 1.3–1.4 m instead of the commanded 1.0 m in larger spaces. This suggests that our current Cartesian point filter clips useful LiDAR returns when the wall is farther away, biasing



Figure 2: Cross-track error distributions by test category. The right-wall runs are more concentrated near zero than the hallway and left-wall runs.

the fitted wall outward. The left-wall run also underperformed the right-wall runs, requiring 55.2% steering saturation and producing 0.323 m mean CTE.

Finally, the safety controller did prevent the car from ignoring frontal obstacles, but it did not guarantee comfortable corner margins. The closest frontal approach was 0.13 m in Hallway Test 1. Overall, our controller performed well on straight wall segments and longer steady runs, but it remained less reliable in hallways and around corners.

## 4 Conclusion (Kelsey)

In this lab, we developed a controller to stop the car safely as it comes across obstacles during autonomous running, improved our wall following algorithm for physical running, and evaluated both after tests in simulation and the real world.



Figure 3: Best-run deep dive for Right Wall Test 1, showing wall-distance tracking, cross-track error, steering commands, and commanded speed over time.

Our results showed that the wall follower performed best on straight, continuous wall segments, where the car was able to maintain relatively accurate tracking with smooth steering corrections. The right-wall tests in particular showed that our RANSAC-based wall estimate and PD controller worked well when the wall geometry remained consistent. However, our hallway tests also showed that the controller was less reliable around corners and at higher speeds, where abrupt changes in visible wall geometry caused larger tracking errors and slower recovery.

Overall, this lab showed us that success in simulation does not guarantee equally strong performance on the physical platform. Real-world LiDAR noise, wall visibility, and course geometry all had a meaningful effect on controller behavior. Moving forward, the most important improvements will be making wall esti-

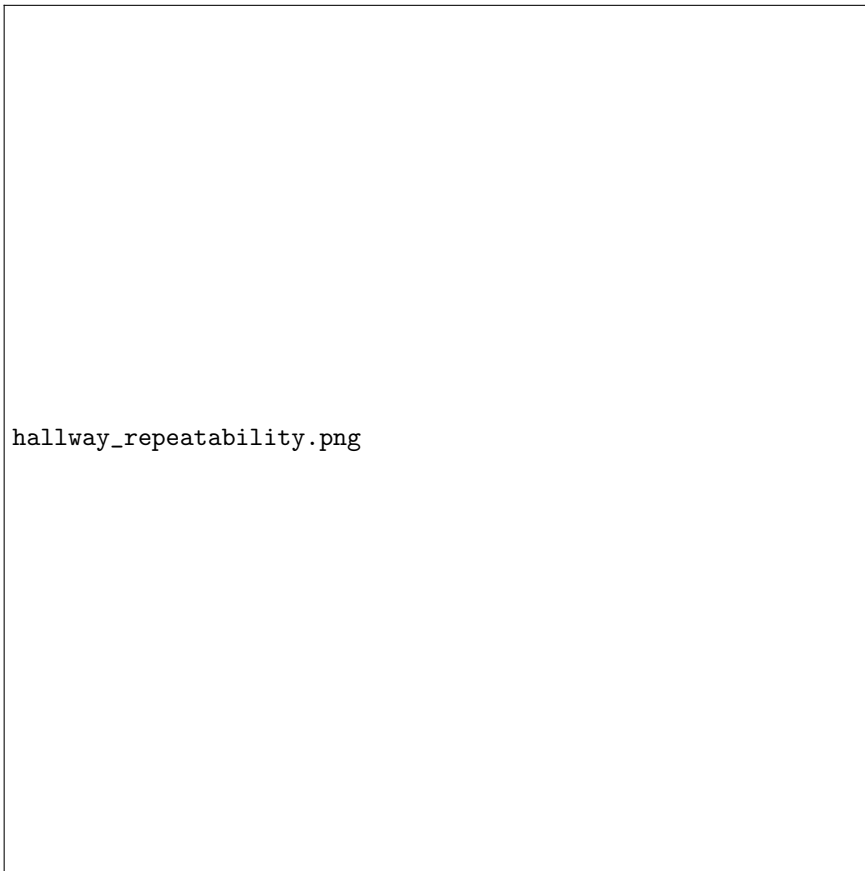


Figure 4: Hallway-run repeatability. The faster hallway run showed larger oscillations and less stable wall tracking than the nominal-speed runs.

mation more robust when the wall is farther away or temporarily lost, and improving controller behavior in corners so that the racecar can navigate more reliably in more complex environments.

## 5 Lessons Learned

### 5.1 Kelsey

In this lab, I learned a lot about the importance of planning and coordination as early as possible, especially since everyone's schedules could be unpredictable and without much overlap, as well as the importance of constant communication. I also learned that issues may be appearing in our controller might be due to the data that is entering it, meaning we need to consider each level when debugging.

## 5.2 Yifan

In this lab, I learned the importance of planning ahead of time especially given a relatively short time frame to work on a problem. It is more challenging to evaluate robots in real life, and we should have a schedule that we can adjust comfortably when unforeseen scenario occurs. I also learned the importance of communication, especially when editing the code. We need to form a more structured way of reviewing and editing code through git.

## 5.3 Dante

One of the biggest things I learned from this lab was that team planning needs to happen much earlier when everyone has conflicting schedules and only a few shared openings. If we do not plan far enough in advance, it becomes much harder to make time for testing, debugging, and writing.

I also learned that high-level design choices should take real-world experimental issues into account, not just whether the algorithm works in theory. In our case, reproducibility mattered a lot, and the stochastic behavior of RANSAC made evaluation and debugging more difficult across runs.

## 5.4 He'yun

I learned that evaluation of performance of a system like this, with many distinct moving parts and entanglement with heavy real-life ethical issues, involves a lot more rigor and attention to detail than I had initially imagined. Discussing the project with our instructors, tackling the evaluation ourselves, and listening to other teams' briefings helped me realize this. I also learned that coordinating a big team project like this takes significant planning and should not be done halfheartedly.