Lab 3 Report: Developing a Robust Wall-Following System

Team 13

Inimai Subramanian (6-2)
Insuh Na (16)
Russell Perez (6-2)
Xavier Sanchez (2A)

RSS

March 15, 2025

1 Introduction - Xavier

Lab 3 required the development of a robust wall-following algorithm and safety controller, to be deployed on an RC race car. These technical problems built on work in Lab 2 for refining a wall-follower in simulation, but in translating these foundational algorithms to fragile hardware, an emergency braking system was introduced as a necessary precaution. Looking ahead to the continued development of our autonomous system, a reliable algorithm for wall following will prove immensely valuable as a framework generalizable to other trajectory tracking problems. For example, Lab 4 will require computer vision to track an arbitrary line.

Technical requirements of Lab 3:
(1) construct an optimized wall follower algorithm for the physical race car, capable of performing at a variety of speeds and wall distances
(2) implement a safety controller to prevent crashes without hindering wall-follower performance
(3) use experiments to collect quantitative performance data of the wall-follower and safety controller, to be used for optimization and evidence of robustness
(4) demonstrate the successful operation of our wall-follower and safety controller on the race car

Our implementation of the wall-follower feeds LIDAR data collected by the race car to a PD controller, which tracks a set distance to the wall using gains we tuned from experiments at different driving speeds. Additional weighting logic is used to perform precise turns. For the safety controller, we built a buffer zone around the dimensions of the car, capable of triggering a full stop when an obstacle is detected within its range. In front of the car is an additional 'warning zone' that brakes according to both obstacle distance and current vehicle speed.

2 Technical Approach

2.1 Wall-Follower Formulation

2.1.1 Main Algorithm (from sim) - Russell

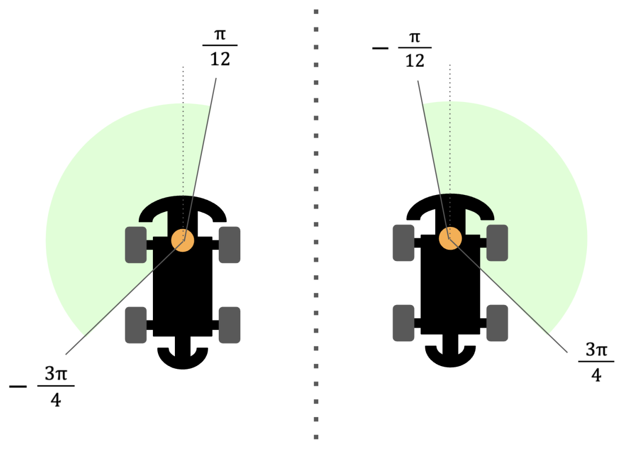

Figure 1: Angle Ranges for Wall Scan

For our initial approach to lab 3, we used the wall following code from Russell's approach to lab 2. In this algorithm, the car subscribes to LaserScan data, which allows it to see an array of distances from the car along an arc. From the data, an angle $\theta = 0$ indicates a scan point in front of the car, with an outer range defined from $\theta_{min} = \frac{-3\pi}{4}$, to $\theta_{max} = \frac{3\pi}{4}$. Depending on which wall we want to follow, either right or left, we filter out the LiDAR scan values we don't want to see. To follow the left wall, we only care about the LiDAR scan in an angle range $\frac{-\pi}{12}$ to $\frac{3\pi}{4}$ which captures data from the car's left side to a little beyond the front of the car as seen in Figure 1. We chose these values to see as much of the environment we needed to created an informative turning context as needed, for both outer and inner turns. Experimenting with other angles proved that this was the best arc range. One problem that arises from this approach is that the car will include distance values to walls that are very far away from it. To solve this, the code incorporates a range filtering algorithm which removes any distances greater than three meters, using a mask. Once we have the angles and distances we want, we convert them into x-y coordinates and feed these values into a linear regression function. This generates an approximate line of our wall, and we can determine how far away we are from this wall by finding our perpendicular distance to it. This distance is fed into the PD controller.

2.1.2 PD Control - Russell

Once we have a function to determine an approximate wall to follow, and the distance from that wall, we are able to implement a proportional-derivative controller. The first step is to determine the difference between our current distance and our ideal tracking distance. We call this term our error. We also keep track of our previous error, our current time, and our previous time. From this we are able to get: $\Delta error = current\_error - previous\_error$ and $\Delta time = current\_time - previous\_time$. For our proportional controller, we multiply our *error* term with a K_P value. To incorporate a derivative controller, we multiply a K_D value with $\frac{\Delta error}{\Delta time}$. We send the sum of these two terms to the steering angle in our drive command, which is executed by the race car to turn.

2.1.3 Following the Wall - Russell

In simulation, we found that having a K_P value of 4 and a K_D value of 2 worked really well, with a score of 97% achieved in Lab 2. However, even in simulation when making an inner turn the car came too close to the wall for comfort, and if we incorporated a safety controller we knew it would stop during a turn like this. In tuning our gains, we prioritized minimizing oscillations, which we would be able to see very well in simulation. However, when transferring the wall following algorithm to our car, we saw poor performance when using just our Lab 2 algorithm, so knew we needed to adjust the algorithm and re-tune our k values.

2.1.4 Inner Turns - Insuh

It was apparent in both the wall follower simulations and in real-world testing that the car was executing 90 degree inner turns too late. This would cause the car to get too close to the wall during turns and sometimes even create collision scenarios. The approach that was taken for this problem was to apply a scaling factor to the distance to the linear regression 'wall' that the controller perceives, based on the scanned distance to the front wall at $\theta = 0$. The scaling factor increases as this front distance decreases, so the car turns more aggressively as it enters an inner turn.

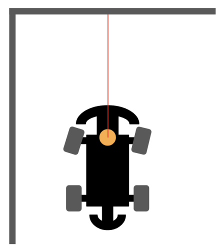

Figure 2: Inner Turn Diagram

2.1.5 Outer Turns - Insuh

In simulation, the linear regression for estimating the car's distance to the wall seemed to extend well to outer turns. However, to improve reliability and keep our turning logic consistent with inner turns, we added an explicit case check. A scan of the leading wall at $\theta = \frac{\pi}{3}$ is used to scale the calculated distance to the linear regression 'wall'. The scaling factor increases as this leading distance increases, so the car turns more aggressively as it approaches an outer turn.

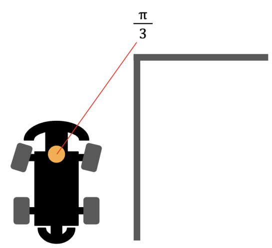

Figure 3: Outer Turn Diagram

2.2 Final Wall-Follower - Inimai

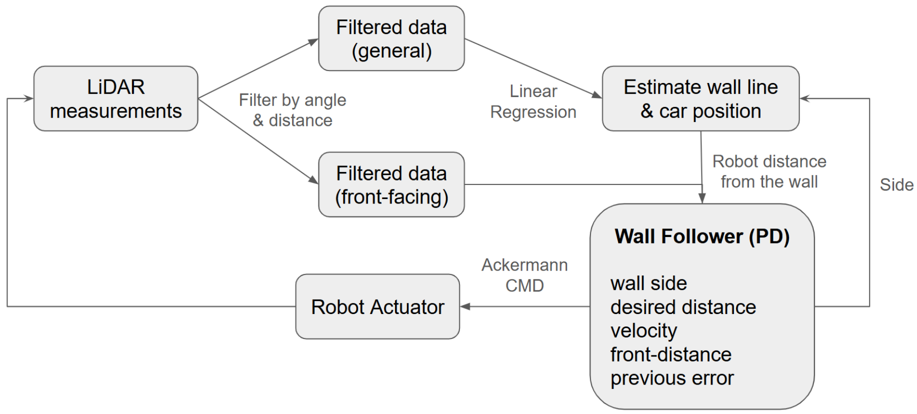

Figure 4: Final System Diagram of Wall-Follower Implementation

2.3 Safety Controller Formulation - Insuh

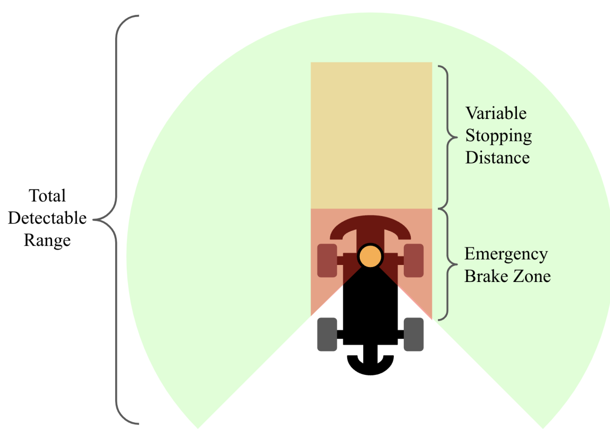

Figure 5: Safety Controller Layout

2.3.1 Main Considerations

A robust safety controller is essential for testing and operating autonomous robots outside a simulated environment. Our team created a node called safety_controller that publishes Ackermann commands to /vesc/low_level/input/safety, which takes priority over any command that the wall_follower code publishes. This means that whenever a safety-critical event occurs, the safety controller takes precedence, ensuring that emergency actions are executed immediately. Properly utilizing this command priority through accurately defining and efficient stopping algorithm is the main challenge for this safety controller.

2.3.2 Emergency Brake Zone

A key aspect of our safety controller is its monitoring strategy. Instead of focusing on a small range of the environment around the vehicle, it evaluates the full scope of sensor data to identify potential obstacles. Within this overall detection field, it defines a close range boundary box that serves as an emergency brake zone, as seen in the red bounding box, figure 5. Should any obstacle be detected within this immediate zone, the controller triggers an instant stop, providing a rapid response to sudden hazards.

2.3.3 Variable Stopping Distance

Beyond this immediate response area, the controller also defines a variable stop distance zone that extends ahead of the vehicle, as seen in the orange area, figure 5. This zone is not static; its length is determined by a linear function of the current speed of the vehicle. In other words, as the vehicle's velocity increases, the stopping distance zone grows longer. This adjustment is crucial because it accounts for the increased distance required to safely stop the car at higher speeds. The exact relationship - defined by a braking constant was determined through experimental tests, in which stopping distances were measured in a range of speeds to ensure the reliability and safety of the system under various conditions.

3 Experimental Evaluation - Inimai

3.1 Evaluating the Wall-Follower

We started with a simple PD controller to determine the best LiDAR filter angle ranges to approximate the wall. After widening and narrowing the ranges (and comparing the calculated line to the actual wall), we determined that the best angle ranges were $-\frac{3\pi}{4}$ to $\frac{\pi}{12}$ for the right wall and vice versa. This includes points to the side and front of the car to aid in turning or other obstacles.
Next, we quantitatively tested the controller as follows:

1. Implement an extremely simple barebones PD controller.
2. Test the racecar on a simple flat wall.
3. Set $K_P = 0.5$. Calculate the average absolute loss according to the formula: $loss_{avg} = \frac{1}{N}\sum_{i=0}^{N} |d_{desired} - d[i]|$ where $d_{desired}$ is the preset desired distance and $d[i]$ is the robot's distance to the wall for timestep i.
4. Plot the distance to the wall vs. time.

5. Repeat steps 1-4 but increase the K_P until the loss stops decreasing. Set that K_P value constant and incremently increase the K_D starting from 0.1 until the loss stops decreasing.

6. Repeat steps 1-5 for velocities of 0.5 m/s, 1 m/s, and 2 m/s.

3.1.1 Racecar Evaluation & Metrics

Thus, we completed the following evaluation to tune the gains and turning angles:

1. Complete steps 1-6 from the previous section for a simple wall-following scenario to tune K_P and K_D.

2. Inner turning: We divide the distance at 0 rad by a factor (to force a turn). We performed several trials at various speeds.

3. Outer turning: If the distance in the turning direction is large compared to the current distance to the wall, the current wall will wrap around as a corner. We also divide the distance by a factor and perform several trials.

3.1.2 Racecar Performance

Wall-Following (Gain Tuning). For gain tuning, we collected the data shown in Table 1 displaying average absolute error. A K_P of 0.8 provided the lowest value for the different velocities. We subsequently tuned the K_D with error results shown in Table 2.

K_P	0.5 m/s	1 m/s	2 m/s
0.5	0.079	0.061	0.199
0.8	0.044	0.052	0.245
1.1	0.065	0.129	0.149
1.4	0.026	0.122	0.224

Table 1: K_P tuning: Average Absolute Error of Racecar Straight Wall Driving ($d_{desired} = 0.5$m and $K_D = 0$)

K_D	0.5 m/s	1 m/s	2 m/s
0.1	0.051	0.118	0.284
0.2	0.054	0.043	0.172
0.3	0.048	0.075	0.188
0.4	0.047	0.059	0.146

Table 2: K_D tuning: Average Absolute Error of Racecar Straight Wall Driving ($d_{desired} = 0.5$m and $K_P = 0.8$)

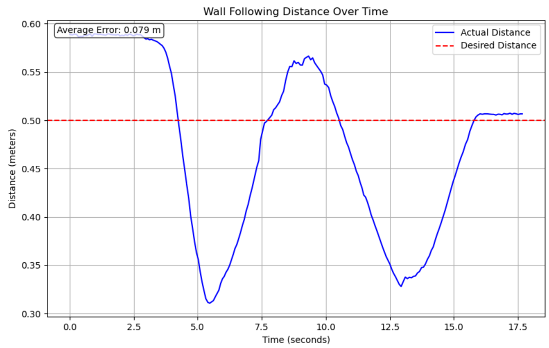

Figure 6: Racecar Distance to the Wall vs Time ($K_P = 0.5$, $K_D = 0$, $v =$ 0.5m/s, $d_{desired} = 0.5m$)

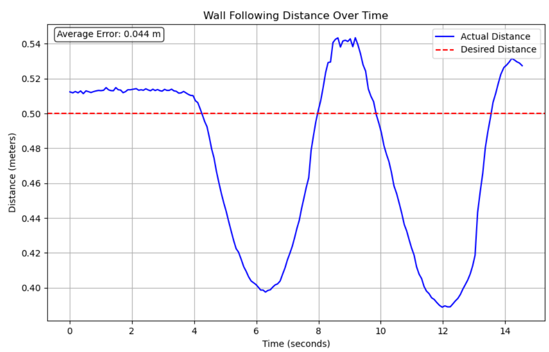

Figure 7: Racecar Distance to the Wall vs Time ($K_P = 0.8$, $K_D = 0$, $v =$ 0.5m/s, $d_{desired} = 0.5m$)

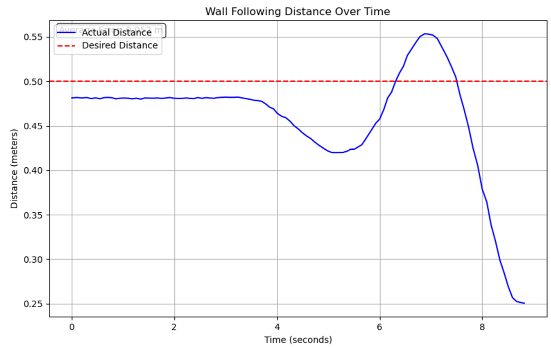

Figure 8: Racecar Distance to the Wall vs Time ($K_P = 0.8$, $K_D = 0$, $v = 1$m/s, $d_{desired} = 0.5m$)

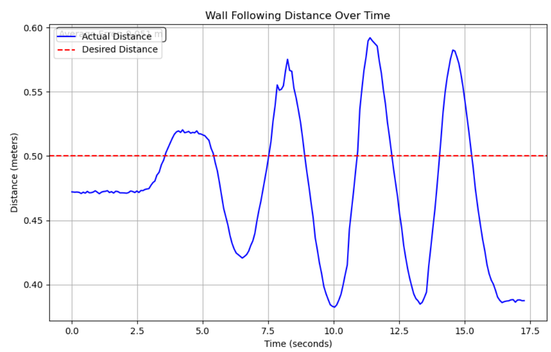

Figure 9: Racecar Distance to the Wall vs Time ($K_P = 1.7$, $K_D = 0$, $v = 0.5$m/s, $d_{desired} = 0.5m$)

The distance vs time plots show the stability and number of oscillations for each run (Figures 6 - 9). Compared to Figure 9, the oscillations of Figures 6 - 8 are fewer in number and have smaller amplitude, showing that a K_P of 1.7 (Figure 9) is too high. Figure 7 ($K_P = 0.8$) has smaller amplitude oscillations compared to Figure 6 ($K_P = 0.5$). Thus, we used the stability of the controller, as shown in the graphs, to validate the average absolute error metric and tune the gains. We settled on final gains of $K_P = 0.7$ and $K_D = 0.4$.

Inner Turning. We determined the inner turning parameters qualitatively. We conducted several tests where we divided the distance by factors of 2, 3, 4, and 5. We created a constant proportional to the velocity of the car to accommodate for its speed. The results are summarized in Table 3, where $3v$ produces the best results (v = car velocity).

Inner Turn Factor	0.5 m/s	1 m/s	2 m/s
$2v$	Successful Turn	Successful Turn	Too Late
$3v$	Successful Turn	Successful Turn	Successful Turn
$4v$	Too Early	Too Early	Successful Turn
$5v$	Too Early	Successful Turn	Successful Turn

Table 3: Inner Turn Factor Tuning: Qualitative Inner Turn Performance (v = racecar velocity)

Outer Turning. When the car receives a point from a specific angle in the direction of turning that is 5 times its current distance to the wall, the car should turn in that direction. We use the same turning factor of 3 and tuned the distance-receiving angle qualitatively, by observing how well the car follows the turn. The results are shown in Table 4, with $\frac{\pi}{3}$ producing the best turns.

Outer Turn Angle (rad)	0.5 m/s	1 m/s	2 m/s
$\frac{\pi}{6}$	Too Early	Too Early	Too Early
$\frac{\pi}{4}$	Too Early	Too Early	Successful Turn
$\frac{\pi}{3}$	Successful Turn	Successful Turn	Successful Turn
$\frac{\pi}{2}$	Too Late	Too Late	Too Late

Table 4: Outer Turn Angle Tuning: Qualitative Outer Turn Performance

3.2 Evaluating the Safety Controller

3.2.1 Racecar Evaluation & Metrics

We tested the safety controller as follows:

1. Determine the stopping distance of the car for a range of velocities.
2. Collide the car with the wall head-on and at 45 degrees at a variety of speeds.
3. Throw random obstacles in the car's trajectory.

3.2.2 Racecar Performance

Stopping Distance. The stopping distance of the car is proportional to its velocity squared. This allows the car to stop in time when moving faster. We predict the distance based on drive conditions with the fitted data (Table 5) in Figure 10.

Car Velocity (m/s)	Stopping Distance (m)
0.25	0.06
0.5	0.08
0.75	0.1
1	0.11
1.25	0.24
1.5	0.43
1.75	0.41
2	0.47
2.25	0.558

Table 5: Stopping distance of the car based on its velocity

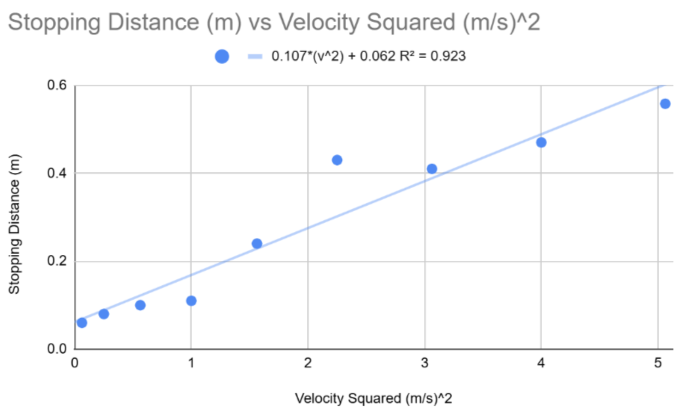

Figure 10: Racecar Stopping Distance vs Velocity Squared

Collisions with Walls. We attempted to collide the car at 0.5, 1, 1.5, 2, and 3 m/s and steering angles of 90, 45, and 30 degrees relative to the wall. The car was able to stop for each of these 15 scenarios with enough room to spare.

Random Obstacles. We tested whether the car could stop at a stationary object in front of it, a horizontally moving obstacle (a person walks in front of the car's trajectory), and a vertically moving obstacle (setting down an object from above in the trajectory). The car was able to successfully stop for all these scenarios.

4 Conclusion - Xavier

In Lab 3 we set out to achieve two goals:
1) to use previous work to optimize a wall follower for hardware deployment under a variety of speeds
2) to create a safety controller for automatically modulating the race car's speed to prevent collisions

During the design phase, we settled on a PD controller that tracks a set distance to the detected wall, with additional front and side wall proximity weighting to perform turns. Our safety controller uses a static buffer zone with an additional, velocity-dependent 'warning' zone to modulate braking.

From our performance metrics, we see that our safety controller meets the design requirement of preventing collisions. The race car successfully slows and stops before hitting the wall when driving towards the wall at various angles (90, 45, 30) and speeds (0.5, 1, 1.5, 2, 3). The car also stops in time when an obstacle is suddenly introduced to it's static buffer zone .

In a similar thread, our metrics demonstrate that our wall-follower is fairly competent, capable of performing effective turns and decent wall following with minimized oscillation after our gain-tuning protocol. However, there are areas we see as open for future improvement:
1) shortening time to steady-state convergence, and reducing oscillations, at variable speeds
2) being robust finding the wall in the event of tracking loss
3) making sharper turns
These refinements to our generally effective wall-follower will assist in the design of an even more effective line tracking controller in Lab 4, and for the Final Challenge.

5 Lessons Learned

5.1 Inimai

Technical: This lab taught me a lot about the importance of iterative testing and debugging. Everything was built off of data-driven decisions that informed our next steps in the process. I also became more comfortable with feedback control mechanisms, actually driving and programming on the robot, and writing simple readable code.

CI: I learnt how to communicate my ideas in a simple manner in this lab. The briefing and report were both very useful in allowing me to distill the most important messages and data, and present them in a way that is understandable and interesting to the audience. I also became more fluent in stating and supporting claims to support technical ideas and design decisions. Finally, I became more comfortable with working in my team and communicating our strengths and interests.

5.2 Insuh

Technical: This lab has provided me a rigorous experience that helped me familiarize myself with interfacing with linux and remote connections (ssh). This is something that I have never done before, so the repeated usage really helped me understand the platform as a whole.

Teamwork: I have learned the importance of planning ahead and properly scheduling with my team. Especially for lab 3, if it was known that people weren't going to be available during the weekend, we could've pivoted and scheduled to work more towrads the beginning of the week.

5.3 Russell

Technical: Throughout this lab, I learned many times how much simulation does not agree with real life. There were many times when we would convert from simulation to real life where I thought to myself, "Why doesn't this work as well as it did in the simulation?" Having to incorporate new aspects into the wall-following controller, such as navigating inner and outer turns, proved to be a learning moment. It is also important to ask these questions so that we can identify the problems that come with transitioning to real life and accurately adjust our algorithms to better suit the actual car.

CI: One very important lesson I learned in this lab is that communicating with each other and better managing our time is essential to getting the work done. We also need to have everyone present when we meet so that when we do run into problems, we are all able to help diagnose the issue. Having everyone communicate which parts they are working on and where they are in the process is also very important to ensure we understand each person's progress throughout the labs.

5.4 Xavier

Technical: I learned the importance of debugging, including proper tuning of any parameters, and of seeking help/collaborative insight. In Lab 2, I was unfortunately unable to create an effective wall follower, but after looking at my teammates' code I saw that I had implemented most, if not all, of the strategies very similarly in my own code. However, I recall that after running into performance issues, I tried to make the controller more complex, which introduced more bugs and ultimately led me *away* from a working solution.

Teamwork: I am still learning the importance of seeking help and collaborative insight. By asking for help from TAs or my peers on Lab 2, I could have found the original errors in my code, and see that my gains simply needed tuning, saving me time and allowing me to be successful in the lab. I am also learning the importance of being my own advocate for scheduling time to work on the project which fit my schedule. There were several instances during Lab 3 when I agreed to a less-than-ideal meeting time, and could only show-up for a fraction of the time my teammates had dedicated to the project. In cases where this is unavoidable, a better solution would be to come to a consensus with my team on how and when I could work on the project independently, allowing me to continue contributing equally to the project.

CI: From the CI Forums I definitely learned about the importance of communicating ideas with simplicity. Regardless of how complex an idea is, visual aids, clean slides, a conversational tone, an emphasis on takeaway ideas, and a story-like presentation format can help an audience understand what you are talking about.