# Lab 3 Report: Safe Wall Following in the Real World

Team 15

Bilal Asmatullah
Jing Cao
Megan Tseng
Michelle Wang

Robotics Science and Systems

March 15, 2025

## 1 Introduction

**Author(s): Jing Cao**
Autonomous mobile robots require robust navigation strategies to operate safely and efficiently in real-world environments. This lab focuses on designing a reliable wall-following system alongside a robust safety controller, both of which are critical for autonomous navigation.

Building on foundational control principles, this lab introduces safety mechanisms essential for real-world deployment. By integrating sensor feedback, motion control, and dynamic safety constraints, we develop a system that enables smooth and collision-free navigation—a crucial step toward fully autonomous operation.

The technical challenge lies in precisely controlling the robot's position relative to a wall while adapting to obstacles and sudden environmental changes. Our solution employs a PD controller for accurate wall-following, a hard-turning mechanism to handle sharp turns and wall loss, and a safety controller that dynamically adjusts stopping distances based on velocity. Together, these components create a balanced system that prioritizes accuracy, adaptability, and safety, preparing the robot for more complex navigation tasks in future labs.

# 2 Technical Approach

**Author(s): Jing Cao**
In this lab, we addressed the following objectives:

1. Maintain a steady distance from the wall using a PD controller for smooth and precise wall-following.

2. Handle sharp turns and wall loss recovery using a hard-turning and look-around mechanism.

3. Build a robust safety controller that stops a predefined distance away from an obstacle dependent on the robot's speed.

## 2.1 Initial Setup

**Author(s): Jing Cao**
We began with a pre-configured ROS2 environment on the racecar. A command mux with multiple priority levels was provided, which we used to implement our wall-follower and safety controller. The available topics, in descending order of priority, are teleoperation, safety, and three navigation channels. Our wall-follower published to `/vesc/high_level/input/nav_2`, while the safety controller published to `/vesc/low_level/input/safety`.

Before the lab, each team member independently wrote a wall-follower implementation. After testing, we combined two versions that demonstrated the most desirable properties: Jing's steady wall-following and Megan's effective corner-turning mechanism.

## 2.2 Wall Follower

**Author(s): Megan Tseng**
Our wall follower aims to maintain a steady distance from the wall and adapt quickly to obstacles in a messy environment. To accomplish this, we elected to use proportional and derivative (PD) control. Figure 1 shows the block diagram for PD control, sensor feedback, and an additional turning mechanism.

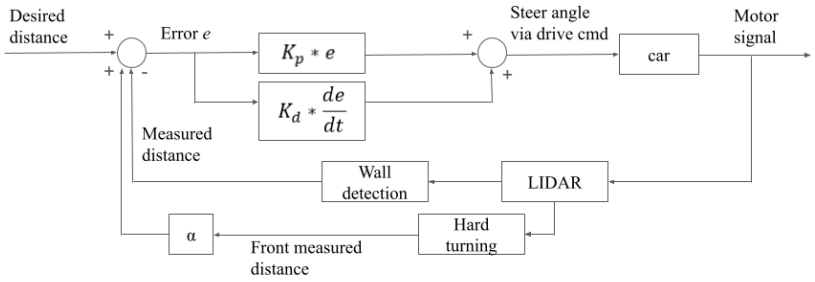


Figure 1: **Block diagram of the PD controller.** It takes desired distance from the wall as a step input and outputs the necessary steer angle through an AckermannDrive command in the /vesc/high_level/input/nav_2 topic.

A PD controller contributes only phase lead compensation, so we were able to optimize the system's response for transient characteristics like rise time, settling time, and overshoot. Phase lead creates an increased phase margin at high frequency inputs, which gives our robot a greater resistance to disturbances in feedback from sudden obstacles or periodic errors in wall detection. As a result, it enjoys improved stability and fast adjustments to its environment.

While we originally considered incorporating integral control, large variations in wall detection resulted in integral windup from high accumulated error, forcing our car to overcorrect and oscillate. We ultimately decided to neglect integral control and prioritize a near-critically damped response, since it was most important to safely navigate past obstacles without unsteady behavior that could lead to a crash. The tradeoff is the presence of steady-state error. To justify lack of an integrator, we specified that our controller be tuned to achieve an average steady-state error within 10% of our desired distance from the wall (input).

### 2.2.1 Wall Detection

The controller uses linear regression to evaluate the robot's distance from the wall (Fig. 2). It takes in a "slice" of LIDAR data within the 60 to 120 degree range and performs linear regression to determine a linear fit. Then, we calculate

the perpendicular distance $d$ from this line via

$$d = \frac{|b|}{\sqrt{1+m^2}}$$

We experimentally tuned this range to produce a line that would react appropriately to obstacles but not oscillate with small changes in wall pattern. In the block diagram, this measured distance to wall is subtracted from desired distance to produce the controller's error input.

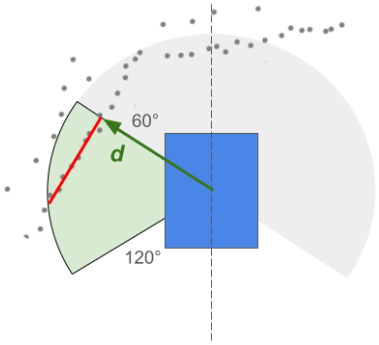


Figure 2: **Diagram of the robot's wall detection via linear regression.** The linear fit is shown in red.

### 2.2.2 Turning Mechanisms

We incorporated two modules from Controller M to assist the robot with turning. One of them is a "hard-turning" mechanism (Fig. 3a) for handling obstructed turns when approaching a wall or obstacle. The robot evaluates a -29 to 29 degree slice of LIDAR within a 2.0 meter range and calculates mean distance from the points. This "front measured distance" is multiplied by an experimentally tuned gain $\alpha$ and summed into the controller error (Fig. 1). The error contribution is inversely proportional to distance, allowing the commanded steer angle to amplify significantly until the robot clears the turn. This enables the robot to handle sharp turns without requiring parameter tuning for different turn radii.

Another is a "look-around" mechanism (Fig. 3b) for handling unobstructed turns when the wall falls out of the 60 to 120 degree range utilized by the wall detection module. The look-around essentially overrides the controller and directly commands a 114 degree steering angle towards the side that the robot is following. This continues until it has turned close enough to detect the wall. This steering angle was tuned so that the robot would relocate the wall quickly without losing sight of the wall.

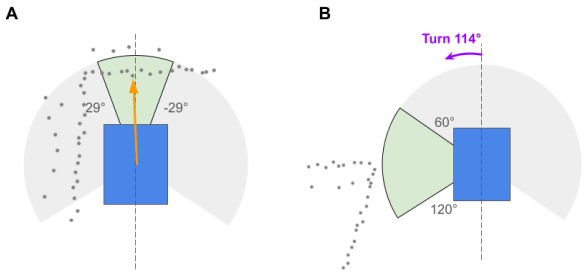


Figure 3: **Diagram of the robot's turning mechanisms.** A) Hard-turning enables obstructed turns. B) Look-around enables unobstructed turns.

## 2.3 Safety Controller

**Author(s): Bilal Asmatullah**

When moving from simulation to reality, damage becomes irreversible. Real collisions can permanently harm the race car, whether from coding bugs or unforeseen obstacles like pedestrians. While our safety controller aims to protect against both, it must allow sharp, precise maneuvers when commanded by correct code, ensuring that safety measures don't compromise performance.

To monitor the distance to potential obstacles directly ahead of the car, we chose to extract LIDAR scan data within a narrow range of -5 to 5 degrees and compute the average distance of points within this range. This narrow field of view allows us to detect not only large obstacles but also smaller objects such as cones, enhancing our controller's robustness to a wider variety of obstacles.

However, simply setting a fixed safety distance for all speeds would lead to suboptimal performance. This could cause the car to stop too early at low speeds, reducing agility, and too late at high speeds, compromising safety. To address this, we developed a model that dynamically adjusts stop commands based on the car's current speed.

### 2.3.1 Finding the Right Model

To ensure reliable stopping, we collected data on the distance traveled after a safety stop command. Our results showed a linear increase from 0.5 to 2.5 m/s, beyond which stopping distance remained constant. This linear trend is captured by stopping distance (cm) $= 76.7 \times$ speed (m/s) $+ 43.7$. We introduced a maximum stopping distance threshold, and we also applied a minimum threshold as speeds under 0.5 m/s produced negative predictions. Figure 4 depicts this model.

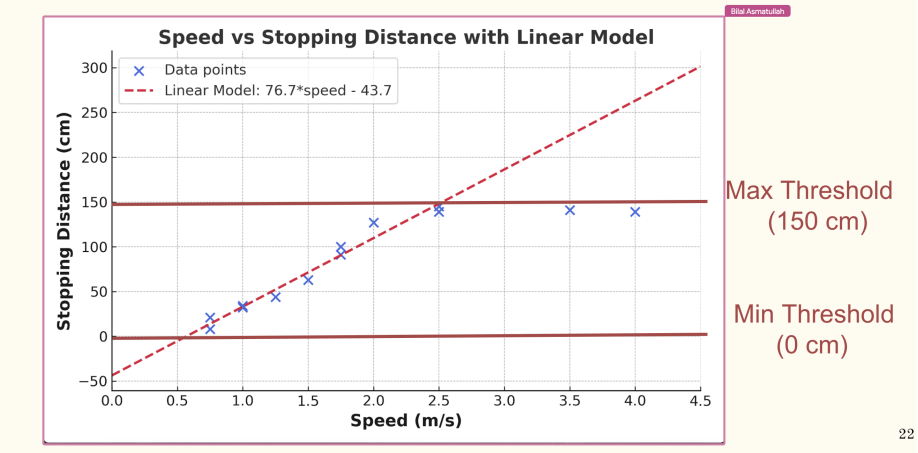


Figure 4: Modeled relationship between the car's speed and its stopping distance, incorporating both minimum and maximum thresholds to ensure reliable stopping behavior.

### 2.3.2 Adaptive Safety Control and Wall Follower Compatibility

To enhance flexibility, we implemented a modular safety controller that lets users set a goal distance—the remaining distance from an obstacle when stopping. This allows adaptation to different environments, with a higher goal distance in open areas or zero for precise maneuvers. The stop command is dynamically issued based on the goal distance and model-predicted stopping distance.

$$d_{\text{stop}} = d_{\text{goal}} + f(v)$$

where:

- $d_{\text{stop}}$ is the total distance at which the stop command is sent.
- $d_{\text{goal}}$ is the user-defined goal distance.
- $f(v)$ is the model-predicted stopping distance as a function of velocity $v$.

In ensuring compatibility with wall follower behavior which relies on sharp turns made close to obstacles, we set our goal distance to zero. However, at higher speeds, our model predicts a stopping distance exceeding the car's turning radius, preventing sharp turns. We avoided hard-coding exceptions for the wall follower, as we wanted to maintain the generalizability of our safety controller across a variety of implementations. Instead, we prioritized ensuring that the controller allows high-agility maneuvers at low speeds while maintaining collision safety at higher speeds.

# 3 Experimental Evaluation

## 3.1 Wall Follower

**Author(s): Michelle Wang**

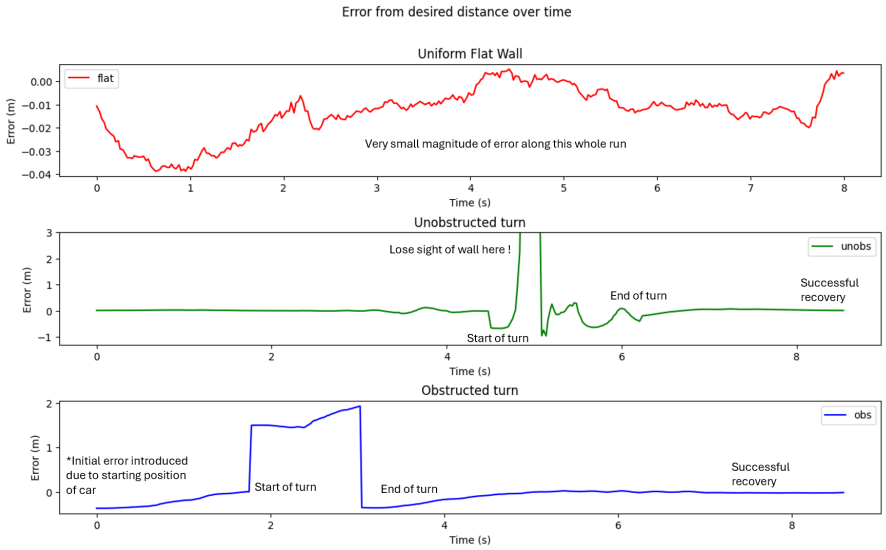


Figure 5: The system exhibits very low error (difference between actual and desired distance from wall) when following a straight wall. When performing turns, deviations in measured distance are expected as the lidar turns, but the system recovers well in both scenarios

Figure 5 validates the successful performance of the combined wall-follower in 3 main scenarios: following a straight wall, an unobstructed turn, and an obstructed turn. On a straight wall, we achieve sub-5 cm error for a desired distance of 1 m away. In the turning cases, we observe a return to the desired distance shortly after successful turn completion.

To compare the performance of individual team members' wall-followers and the combined controller, experiments were conducted for following a flat uniform wall, following a messy wall, and performing an unobstructed turn. To quantify performance, we recorded the error (difference between the actual distance and the desired distance) during testing. For each trial, the root mean square (RMS) error (strongly penalize large deviations) with respect to the fitted line, the mean absolute error (MAE) w.r.t. the line, and the MAE computed using a LIDAR slice on the desired side were recorded. We report errors both relative to the fitted line and the LIDAR since the line is more robust to short-lived environmental disturbances, while the lidar is unaffected by any potential

error introduced during linear regression. The results are summarized in tables 1-3.

| **Controller** | **RMS wrt line** | **MAE wrt line** | **MAE LIDAR** |
|---|---|---|---|
| Controller J | 0.0951 | 0.0434 | 0.0494 |
| Controller M | 0.186 | 0.179 | 0.119 |
| Combined | 0.132 | 0.0627 | 0.0715 |

Table 1: Flat uniform wall

| **Controller** | **RMS wrt line** | **MAE wrt line** | **MAE LIDAR** |
|---|---|---|---|
| Controller J | 0.211 | 0.150 | 0.176 |
| Controller M | 0.327 | 0.262 | 0.207 |
| Combined | 0.161 | 0.114 | 0.151 |

Table 2: Messy wall

| **Controller** | **RMS wrt line** | **MAE wrt line** | **MAE LIDAR** |
|---|---|---|---|
| Controller J | 1.302 | 0.396 | 2.49 |
| Controller M | 0.327 | 0.268 | 0.153 |
| Combined | 0.225 | 0.143 | 0.263 |

Table 3: Unobstructed turn

This study supports the overall superior performance of the combined controller. We now discuss a few exceptions. In the turning test, the combined controller had a higher MAE with the LIDAR. This can be explained by the small slice of LIDAR data having high variability during the turn, and the fitted line being more stable. Also, Controller J outperformed the combined controller in the uniform wall test; this is due to the lack of turning mechanisms in Controller J. However, the difference in performance is not significant, and the combined controller's ability to generalize is far more valuable.

## 3.2 Safety Controller

**Author(s): Jing Cao**

We evaluated the safety controller by measuring its accuracy in maintaining a target stopping distance of 0.2 m across different speeds.

The robot was tested at five different speeds: 0.5 m/s, 1.5 m/s, 2.5 m/s, 3.0 m/s, and 4.0 m/s. At each speed, we performed three trials and recorded the actual stopping distance. The mean error at each speed was calculated as the absolute difference between the actual stopping distance and the goal distance.

The results are summarized in Figure 6 and Table 4. At lower speeds (0.5 m/s and 1.5 m/s), the mean error remained small (1.3 cm and 1.0 cm, respectively). However, at 2.5 m/s, the error increased significantly to 9.7 cm, likely due to it being a boundary case in the piecewise safety controller function. At higher speeds (3.0 m/s and 4.0 m/s), the errors were 2.7 cm and 5.3 cm, respectively, indicating some variability in the controller's effectiveness.

The larger errors at higher speeds suggest that the controller does not fully compensate for increased momentum, leading to overshooting. This is likely due to sensor response time limitations or control loop update rate constraints.

Overall, the safety controller performed well at lower speeds but exhibited higher error at increased velocities. To improve performance, future iterations should refine the function to reduce stopping distance error at boundary cases (e.g., 2.5 m/s) and collect more data at speeds above 2.5 m/s to optimize control across a wider range.

Additionally, when the stopping distance exceeds the car's turning radius, the safety controller activates. To handle this scenario, we propose two approaches:

1. Align maximum stopping distance with the car's turning radius to enable smoother transitions and reduce abrupt stopping, though it may require a wall-following mechanism to avoid collisions.

2. Force turns when the forward distance equals the turning radius to improve maneuverability in constrained environments, though this may cause unintended behavior outside of wall-following scenarios.

These refinements will enhance the controller's reliability and adaptability across different speed ranges and environments.

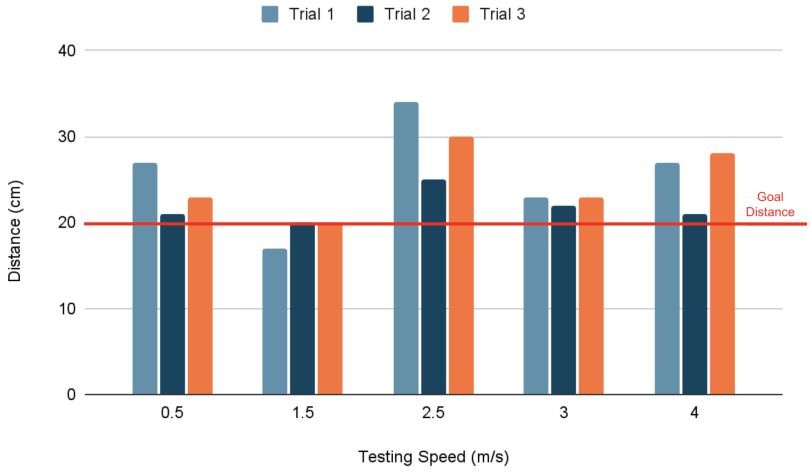


Figure 6: Actual stopping distance of the car given a desired stopping distance of 20cm.

| Speed (m/s) | 0.5 | 1.5 | 2.5 | 3.0 | 4.0 |
|---|---|---|---|---|---|
| Mean Error (cm) | 1.3 | 1.0 | 9.7 | 2.7 | 5.3 |

Table 4: Mean error at each speed.

# 4 Conclusion

**Author(s): Michelle Wang**

In summary, this project successfully developed wall follower and safety controllers for safe and accurate hardware deployment. Starting from different implementations of a wall-follower in simulation, we combined 3 elements of different team members' work: a PD-controller tracking distance to an estimated line, a hard-turning term that encourages sharp turns in front of an obstacle, and a look-around mechanism that searches for a wall if no close wall is detected. This final wall-follower was successful in following a straight wall and performing obstructed and unobstructed turns. It also generalizes to a messy wall, and a turning scenario. It achieved sub-10cm error from the target distance on the uniform wall and had higher overall performance across tasks as compared to the original wall followers of the individual team members.

During real world testing, a need for a safety controller was quickly recognized. Yo generalize to diverse scenarios, specifying a stopping distance is highly de-

sired. However, modeling a real-world system using ideal mathematics is difficult, so we fit our stopping model to the real world instead. By collecting stopping distances for different commanded speeds and target distances, we used regression to fit a model that allowed us to reliably stop the car within 10cm of the target stopping distance.

In the future, we may consider fine-tuning our current PD controller to incorporate an integral term to account for steady state error. Additionally, we discussed the possibility of adding additional logic to our safety controller to allow for making sharp turns in front of obstacles without stopping the car. As we move towards labs involving navigation in more complex environments, these additions may become critical.

# 5 Lessons Learned

**Michelle Wang**: This lab was my first time working intensively in a team setting with git. It encouraged me to be a lot more cogniscent of my version control practices. As projects become more complicated, I expect we will take fuller advantage of git's many features, and I look forward to learning more. Additionally, this lab reinforced hardware debugging practices in my brain: is the power on, is the router working, is my computer on the right wifi, does everything just need to be restarted? In the real world, not everything can be solved with print statements, so the ramp up to working with the new hardware system was a great learning experience.

For collaboration skills, this lab taught me the immense value of planning. From the timeline to the story to exactly what data we need to collect, I realized careful planning is key to an efficient and intelligent workflow. We had to collect the same data three different times because our plan wasn't fully fleshed out until the end; adding on the overhead of working with hardware, this was a lot of time that could have been better spent.Additionally, I learned the value of taking a long pause to think with a clear mind; we had a bit of a misguided exploration with the combined wall follower initially, which could have been avoided if the team collectively paused and pondered for just a minute. Going forward, I aim to have a more diligent plan going into every work session and encouraging lots of space to simply stop and think.

**Megan Tseng**: I learned a lot about controllers in the real world throughout this lab. Previously, I've studied control theory in designing mechanically isolated systems. However, transferring our algorithms from simulation to racecar proved that controls for autonomous navigation is much more complex. One such challenge was tuning parameters for our PID control; while our car worked steadily in simulation, integral control led it to be fairly unstable in practice. In evaluating the wall follower, we considered the effects of factors like variations in drive command rate, lag in error accumulation, and the car's drivetrain

drift. Learning to think about design tradeoffs and ways to experimentally evaluate the performance of a controller were important technical lessons in this lab.

We worked a lot on combining code and taking apart certain features that worked well in individual controllers, so I learned a lot about communicating my work to team members. In the same vein, I gained an appreciation for taking the time to ask clarifying questions and understand other members' work rather than just telling them to implement it on their own. At several points in the project, a bug that felt impossible to figure out was quickly resolved by having all members pool ideas together. Overall, it was a good lesson on how to collaborate effectively as a team instead of individually handling parts of the project.

**Jing Cao**: This lab provided key insights into real-world control systems, particularly in designing and evaluating a safety controller for autonomous navigation. Implementing the controller on hardware revealed challenges like sensor noise and response delays. While it performed well at lower speeds, higher speeds introduced significant stopping errors, highlighting the need for refined tuning and adaptive control. Iterating on the stopping function reinforced the importance of rigorous testing and data-driven optimization.

Beyond technical aspects, the lab underscored the need for careful planning and structured debugging. Small factors—network connectivity, sensor calibration, or overlooked settings—often dictated success or failure. Even minor parameter adjustments significantly affected system behavior, emphasizing systematic testing.

Collaboration was crucial in integrating system components. Balancing individual debugging with team discussions helped resolve unexpected behavior efficiently. A key takeaway was the importance of structured pauses—stepping back to reassess design choices instead of relying on trial and error. Moving forward, I plan to prioritize structured planning, robust controller design, and an efficient debugging workflow.

**Bilal Asmatullah**: This lab was an incredibly insightful experience, particularly in balancing the interplay between software and hardware in real-world control systems. While I have experience in software debugging, working with a physical system added layers of complexity that forced me to develop a more structured debugging approach. Simple things—like checking power connections, ensuring sensor data was correctly processed, and validating real-time behavior—became crucial steps that I hadn't fully appreciated in purely software-driven projects. It reinforced that in practical systems, unexpected issues often arise from overlooked hardware constraints rather than purely code-based errors.

From a collaboration standpoint, this lab reinforced the importance of clear communication and synchronization when working with a team. Using Git in a

more rigorous manner allowed me to appreciate the necessity of version control discipline, especially when multiple people are working on different features simultaneously. I also learned the value of taking a step back to reassess progress rather than charging forward with trial and error.

Beyond the technical lessons, this lab emphasized the importance of trust and flexibility in a team setting. Late-night debugging sessions and last-minute tweaks to our presentation highlighted how crucial it was to support each other and accommodate everyone's schedules. Dividing tasks efficiently and trusting that each team member would execute their part well made the workload much more manageable. Moving forward, I aim to integrate these collaborative habits into future projects by emphasizing structured debugging, thoughtful planning, and effective teamwork.