Reactive Wall Following Using Spline-Fitted LiDAR Scans with Pure Pursuit Control

Christian Cassamajor-Paul
MIT AeroAstro
chrisscp@mit.edu

Alicia Ramierez
MIT Meche
aliciar@mit.edu

Leo Sun
MIT EECS
leosun@mit.edu

Raymond Wang
MIT Mathematics
rayhw@mit.edu

I. Introduction - Christian

This lab covers various aspects of autonomous system development, including wall following and safety control. These features are important for any self-driving car, or more broadly, any autonomous vehicle, to possess as they increase safety and navigation capabilities.

Wall following refers to the process of gathering external sensor data and inputting it to a controller to move forward at a specified lateral distance from a ”wall” (any surface perpendicular to the ground plane) while maintaining a parallel heading relative to the wall. This method of navigation can be used in any case where there is a surface to travel along, e.g, building interiors, mazes, or caves. A successful wall follower should be robust to changes in wall curvature and able to operate at various speeds.

Beyond wall following, this lab also focused on creating a safety controller. The safety controller acts as an emergency brake system that stops the vehicle from crashing into any obstacles. By detecting objects with sensor data and sending a stop command, the vehicle is able to avoid collisions that may be fatal to the obstacle and/or the vehicle itself. A successful safety controller should be able to stop the car before any crashes occur, regardless of speed, without impeding other navigational processes like wall following.

Additionally, this lab sits at a pivotal moment in the RSS journey, as this is the first time that algorithms were implemented and evaluated on the racecar hardware.

II. Technical Approach - Wall Follower - Leo

A. LiDAR Preprocessing

Let a LiDAR scan consist of N range measurements

$$r_i \in \mathbb{R}, \quad i = 1, \ldots, N$$

with corresponding angles

$$\theta_i = \theta_{\min} + i \cdot \Delta\theta$$

where $\theta_{\min}$ is the minimum scan angle and $\Delta\theta$ is the angular resolution. Each valid polar measurement is converted to Cartesian coordinates in the vehicle frame:

$$x_i = r_i \cos(\theta_i), \qquad y_i = r_i \sin(\theta_i)$$

To follow a wall on one side of the vehicle, only points within a 90° sector are retained. For left wall following:

$$0 \le \theta_i \le \frac{\pi}{2}$$

and for right wall following:

$$-\frac{\pi}{2} \le \theta_i \le 0$$

Invalid points (infinite or outside a maximum range) are discarded. To reduce computational load and avoid oversampling, the point set is uniformly decimated to approximately 20 points.

B. Spline Fitting

The filtered points $\{(x_i, y_i)\}$ are used to fit a parametric cubic B-spline

$$\mathbf{p}(u) = \begin{bmatrix} x(u) \\ y(u) \end{bmatrix}, \qquad u \in [0, 1]$$

using least-squares spline fitting with smoothing parameter s. The spline provides a continuous approximation of the wall boundary. The spline is sampled at uniformly spaced parameter values

$$u_j \in [0, 1], \quad j = 1, \ldots, M$$

producing a smooth curve:

$$(x_j, y_j) = (x(u_j), y(u_j))$$

C. Corner Detection

The forward LiDAR region

$$|\theta| \le \frac{\pi}{16}$$

is checked for nearby obstacles. If the mean forward distance d_f satisfies

$$d_f < d_{\text{desired}} \cdot |v| \cdot k$$

where v is vehicle speed and k is a tuning constant, the system assumes an upcoming corner. When a corner is detected, the controller commands a maximum steering angle against the wall-following direction.

D. Offset Path Generation

To maintain a fixed distance from the wall, the spline curve is offset by the desired distance d along its inward normal direction.

First, the unit tangent vector is computed:

$$\mathbf{t}(u) = \frac{1}{\sqrt{x'(u)^2 + y'(u)^2}} \begin{bmatrix} x'(u) \\ y'(u) \end{bmatrix}$$

The corresponding unit normal vector is

$$\mathbf{n}(u) = \begin{bmatrix} -t_y(u) \\ t_x(u) \end{bmatrix}$$

The offset path becomes

$$\mathbf{p}_{\text{offset}}(u) = \mathbf{p}(u) + d \cdot \mathbf{n}(u)$$

which represents the trajectory the vehicle should follow to maintain the desired wall distance.

E. Pure Pursuit Path Tracking

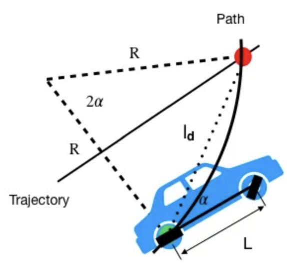

Fig. 1. Pure Pursuit Geometry

The offset path is tracked using a pure pursuit controller. Let (x_g, y_g) be a goal point on the path located approximately a lookahead distance L_d from the vehicle.

The heading error is

$$\alpha = \tan^{-1}\left(\frac{y_g}{x_g}\right)$$

For a vehicle with wheelbase L, the required steering angle is

$$\delta = \tan^{-1}\left(\frac{2L\sin(\alpha)}{L_d}\right)$$

The steering command is saturated to the maximum allowable steering angle

$$\delta \in [-\delta_{\max}, \delta_{\max}]$$

and the vehicle velocity is held constant.

F. System Characteristics

This method exhibits several desirable properties:

- Fully reactive operation without a global map
- Smooth trajectory generation via spline fitting
- Robust corner detection using forward obstacle checks
- Stable path tracking using pure pursuit control

The combination of spline-based wall estimation and pure pursuit tracking provides smooth, stable wall following even in environments with noisy LiDAR measurements.

III. Technical Approach - Safety Controller - Christian

The safety controller is fairly straightforward and only involves 2 significant steps: A) Lidar Filtering and B) Object Detection.

A. Lidar Filtering

First, the lidar data is processed and filtered by finding the data points that correspond to a roughly 60° field of view from the front of the car. Given a lidar scan with N data points, a desired field of view $\theta = \frac{\pi}{3}$ radians, an angle increment ϕ = 0.047576 radians (about 2.73°) and the fact that 0° corresponds to the middle data point, the safety controller will calculate the start index A, and stop index B to mask the lidar data as follows where $K = \frac{\theta}{2\phi}$ is equal to the number that brings the range closest to the desired field of view:

$$\begin{aligned} A &= K - \lfloor\frac{N}{2}\rfloor \\ B &= K + \lfloor\frac{N}{2}\rfloor \end{aligned} \qquad (1)$$

Taking data points r_i such that i ∈ [A, B) filters the data down to the relevant angle range.

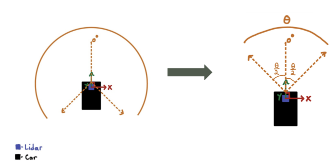

Fig. 2. Lidar Filtering

B. Object Detection

At this point, the safety controller checks each data point going counterclockwise to see if the distance is greater than a specified safety distance parameter. If the number of data points with a reading beyond the safety distance surpasses an empirically chosen safety tolerance parameter, then the controller sends a drive command with a speed of 0 meters per second to the motors. Due to the setup of the vehicle, the safety controller's drive commands override any other drive commands the vehicle may receive from other sources, such as the wall follower. This ensures that safety is prioritized above all else, no matter the situation.

IV. Experimental Evaluation - Alicia and Raymond

To validate our reactive wall-following system, we conducted a structured set of hardware trials on the MIT RACECAR platform. Our evaluation targeted two subsystems—the wall follower and the safety controller.

Together, the tests were designed to characterize steady-state following accuracy, robustness to velocity variation and non-linear walls, and emergency braking reliability. The tests were performed in an indoor environment, Stata basement, with planar well defined walls. Wall distance error was the primary quantitative metric examined, and was calculated by taking a splice of recorded trial LIDAR data and fitting a line to it. More specifically, the distance error was computed as the signed difference between the measured lateral distance and the desired distance, whereby lateral distance was computed by calculating perpendicular distance to the linearly modeled wall.

All together, the implementation was tested against four canonical scenarios that capture the dominant modes of real-world corridor navigation: straight-wall tracking on both the left and right sides of the vehicle, and angled-wall (corner) tracking on both sides. For each scenario, the vehicle was commanded to maintain a fixed lateral 0.5 m offset from the wall while traveling at velocities ranging from 1 to 3 m/s. Additional straight line tests were run to determine whether or not the implementation was robust to different starting angles of incidence with respect to the wall.

A. Straight-Wall Trials

Across all straight-wall trials (including angled trials), the pure pursuit controller achieved a mean steady-state error of -0.0056 m. The negative sign indicates a slight bias toward the wall, though the magnitude is well within acceptable tolerances for a platform of this scale. Qualitatively, these tests were a success. The robot was able to quickly find the wall and return to a reasonable wall-following distance within 1.5 seconds, as the recorded video and graph below suggest.

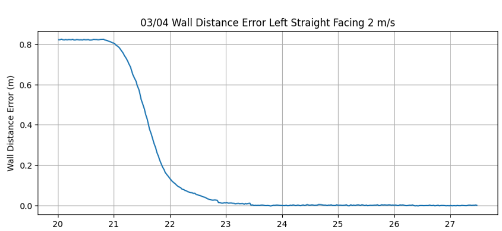

Fig. 3. Facing Toward 2 m/s

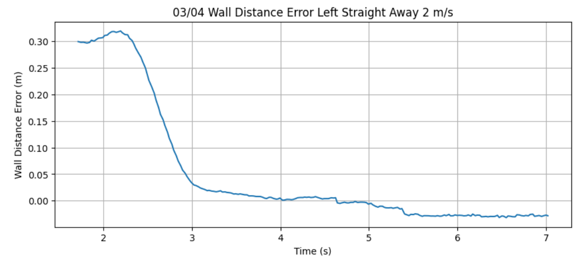

Fig. 4. Angled Away 2 m/s

The time-series error profile for a representative trial (left straight, 2 m/s) illustrates convergence clearly: from an initial offset of approximately 0.30 m, the controller drives error monotonically toward zero within roughly 3 seconds, after which the signal stabilizes. This smooth, overdamped convergence is a characteristic of pure pursuit.

B. Corner Trials

Corner-following trials tested the controller's ability to handle abrupt changes in wall geometry. We sought to test two broad scenarios: turns around a wall (robot on the "outside") and turns against a wall (robot on the "inside").

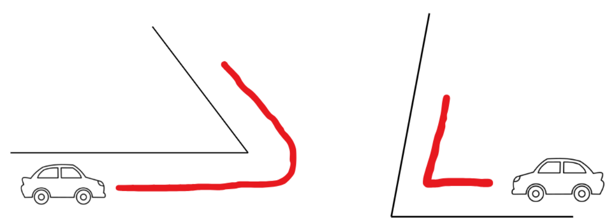

Fig. 5. Obtuse Corner Trial (Left), Acute Corner Trial (Right)

For ease, these corner runs will be referred to as "obtuse" and "acute" trials respectively.

The results of the obtuse-corner trial were promising quantitatively and visually. Here, at 3 m/s, the robot exhibited a negative transient dip to around -0.30 m followed by recovery, with error settling to within ± 0.02 m of the setpoint after roughly 3 seconds.

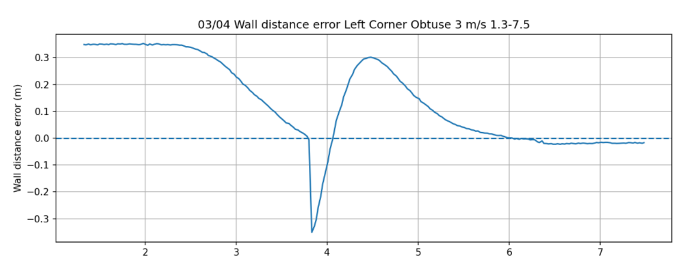

Fig. 6. Left Corner Obtuse 3 m/s

The results of the acute-corner trial were similarly promising, as shown below.

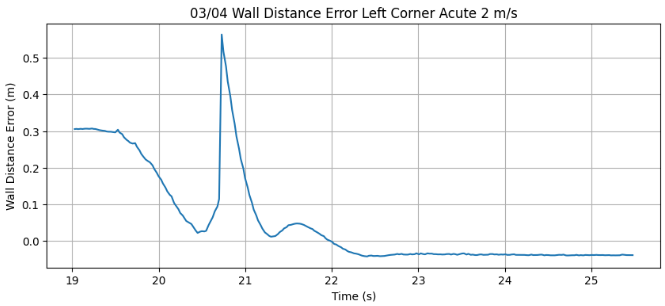

Fig. 7. Left Corner Acute 2 m/s

C. Wall Following Velocity Independence

A key observation of our system's test performance is that wall-following error does not increase as velocity increases within the tested range. We performed a linear regression of mean steady-state error against velocity across all straight-wall trials.. The slope was -0.0063 with an uncertainty of 0.0393 making the slope statistically insignificant, which suggest that the tracking error is independent of velocity, as can be observed in the figure below.

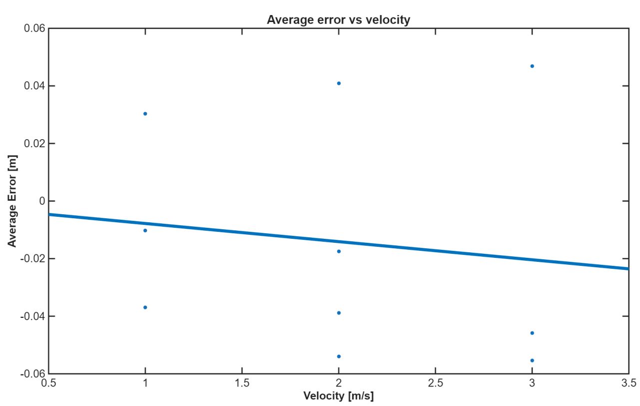

Fig. 8. Linear fit on the average error correlated to the velocity

D. Safety Controller

The safety controller trials had complementary motivations: to determine whether or not the controller itself sensed obstacles and halted the car, and to characterize the real physical stopping distance of the vehicle.

In sim our racecar is able to stop on a moments notice. In practice, several factors contribute to different behavior. For one, there is some small finite time between a LIDAR scan detecting an obstacle and the motor controller commanding zero throttle allows the vehicle to continue at its current speed. The lidar also operates at a fixed scan rate, so at higher velocities the vehicle covers more ground between consecutive detections, further increasing effective reaction distance. Finally, the racecar platform's drivetrain does not produce instantaneous deceleration; mechanical compliance, tire-floor friction limits, and controller bandwidth impose a non-negligible ramp-down period.

The safety controller was evaluated by driving the vehicle directly toward a static, thin board at controlled velocities and measuring the distance at which the vehicle came to a complete stop.

Velocity (m/s)	Safety Range (m)	Measured Distance (m)	Percent Error (%)
1	1.000	0.835	16.5 %
2	1.000	0.395	60.5 %
3	1.000	FAILED	N/A
3	2.000	0.970	51.5%

Fig. 9. Safety Controller Test Results

The results suggest that a safety controller using a fixed distance threshold without accounting for vehicle momentum is suboptimal. This motivates adopting a velocity-dependent safety range in the future, one that potentially scales non-linearly according to our results. Despite this limitation, the controller was confirmed to indeed work under reasonable speed and desired stopping distance combinations.

V. Conclusion - Raymond

All together, our tests suggest a robust wall follower and safety controller. The implementation achieved low loss on nearly all of our tests which were reasonably comprehensive with respect to the types of wall geometries the racecar at any moment may face when following a wall. That being said, there is material further work than could be had, namely surrounding more robust testing — more complex and ambiguous paths, and a dynamic obstacle for the saftey controller test (someone stepping in front of the racecar, for example). Furthermore, running multiple successive tests to ensure that our previously tested results were not the result of variance and are the true deterministic behavior of our implementation is important. That being said, given the robustness of the trials and the implicit correlation across them that mock multiple draws, the team has conviction that our implementation has successfully completed this design phase.

VI. Lessons Learned

Christian - It's important to set up common ground and agreed-upon definitions so everyone is on the same page throughout a discussion or presentation. Additionally, it essential to have comprehensive test planning along with consistent documentation and naming conventions. Even simple things such as saying the name of the test being run at the beginning of a video greatly improve the process of organizing collected data.

Alicia - I learned about the importance of data organization and collection. Without quantitative data it is difficult to arrive at a conclusion about the effectiveness of a physical system. It is also better to have several repeated tests to show that the the accuracy of the wall follower is not a fluke. It is also important to have a good testing plan with previously agreed upon naming conventions in order to streamline data collection.

Leo - I realized that it's important to balance project quality with actually delivering a finished product. I think I was too focused on building a perfect wall follower, and I should have paid more attention to our timeline and the other deliverables we needed to complete.

Raymond - From this lab, I was reminded about the importance of pre-planning and organization. While we rightfully spent much of our time was spent considering wall follower approaches and nuances, we did not consider things in advance as carefully such as what specific data we wanted to construct, how many repetitions of each trial we should run, how to best standardize rosbag recordings and naming conventions such that on a very large scale they were easy to work with, etc.