Lab 3 Report: Wall Follower with Safety Controller

Team 9

David Ulloa
Zhenyang Chen
Meenakshi Singh
Kwadwo Yeboah-Asare Jr.

6.141/16.405 Robotics: Science and Systems

March 5, 2022

1 Introduction

Author: Meenakshi
The RSS Lab 3 Wall Follower is an extension of the previous wall follower lab, where we were tasked to design a feedback controller for a simulated racecar using lidar measurements. Given a wall specification (left or right) the resulting controller used steering angle and speed to move forward autonomously while maintaining a target distance from the wall. In lab 3, we moved forward with our controller from the simulation by adapting it to the physical racecar.
The three main goals of lab 3 were to:

1. Familiarize ourselves with the hardware of the racecar. We accomplished this by connecting to the car, transferring the necessary files, and running the tele-op command. We then manually navigated using the joystick controller and visualized and recorded the laser scan and IMU data from the car.

2. Autonomously drive the robot using wall follower code. We redesigned the wall follower code from lab 2 to account for noise from the lidar and corridors of different sizes (edge cases). The main approach of the wall follower code remains the same as lab 2, using line detection to find the wall and then calculating the error of the desired distance using the least squares method. Some of the changes we made involved filtering points from the lidar that are both too close (noisy data) as well as too far in order to detect a more accurate line representation of the wall.

3. Test and design a safety controller that is robust and prevents the robot from crashes in a variety of different crashes. Due to time and hardware constraints, we implemented a very simplistic safety controller of stopping the robot if it detects an obstacle in front (within a certain threshold). The other safety controller design works by checking if a circle drawn using the maximum steering angle of the robot intersects with the detected wall, in which case it stops. We did this by primarily ideating and testing the robot in simulation by driving the robot towards the wall with varying velocity and angles. Later, we were able to t

We were able to gain experience in all three goals by designing, testing, and implementing the wall follower and safety controller on the physical racecar system. In the following sections we detail our approach for the wall follower and safety controller.

2 Technical Approach

2.1 Wall Follower

Author: Meenakshi
The wall follower incorporates the same general approach from lab 2. We restrict the lidar data to a subset that only considers datapoints on the side of the wall we intend to follow. Then, we fit a line to the data to serve as a representation of the wall. Then, we calculate the perpendicular distance from this line to to the robot and calculate an error from the desired distance using a least squares error function. We started with using PD control but then switched to using a pure pursuit steering model in our wall follower.

2.1.1 Filtering Lidar Data

Author: Kwadwo
We filtered lidar data using a specific radius and fraction of the lidar data. We cut the lidar data on one side to give it a bias to that side for either the left or right side of the wall. This would usually be good enough if we were following an infinite line. However since the are corners and other wall in the space we have to limit the distance of the lidar to prevent factoring points on the opposite wall or far away corners into our calculations for the line to fit. To do this we can just take a minimum distance in which to view our lidar points. If we make this minimum relative to the closest point in the laser data we can still detect walls if we happen to be far away from any walls and have the intended behaviour when we are close to a wall or a corner

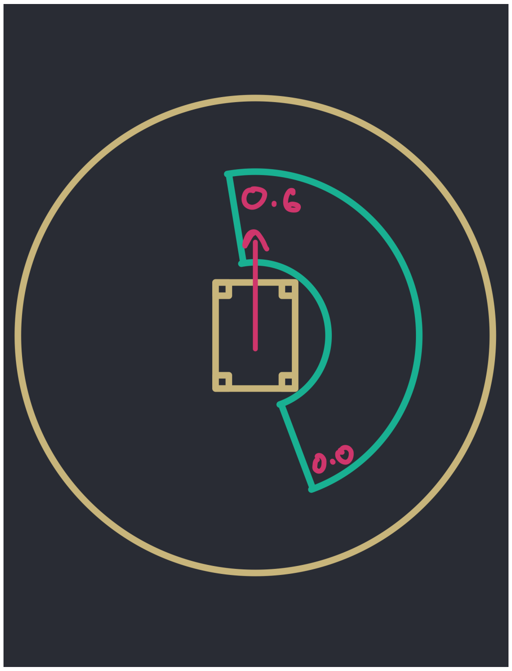

Figure 1: Lidar Representation

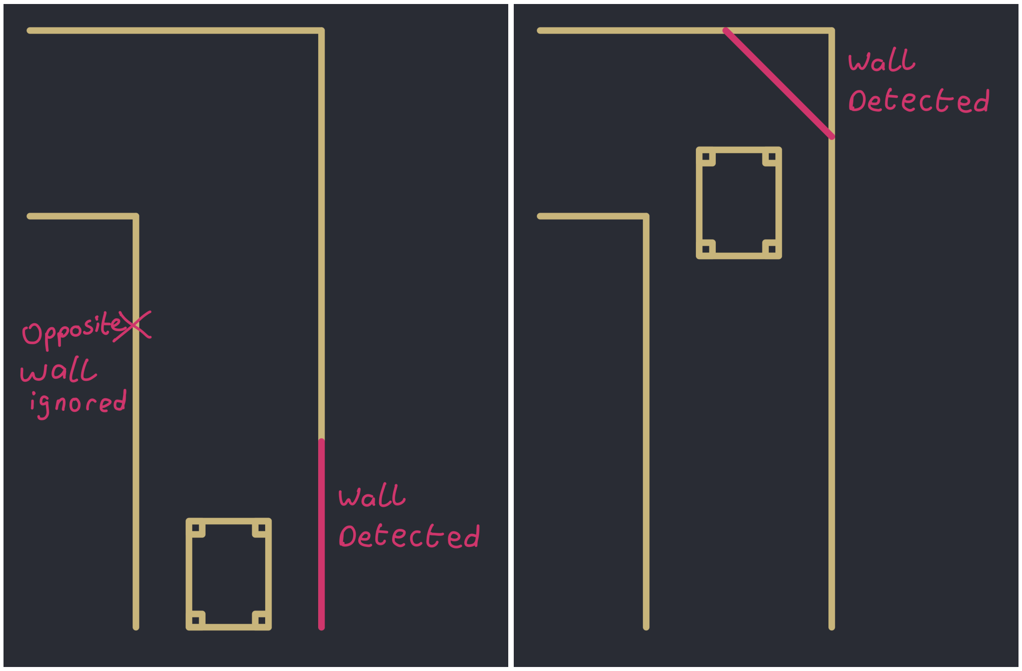

Figure 2: Projected Line far from and close to corner

2.1.2 Line Fitting

Author: Zhenyang

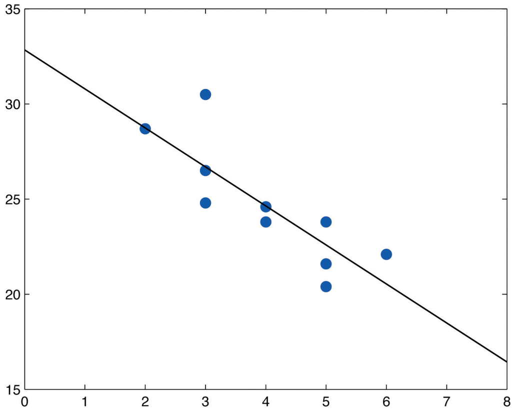

Figure 3: Line Fitting

Given the paired input data of distance and angle, we need to find a line that best fits the data points. We can formulate this problem as a optimization problem. Given data points and the expression of line we want, we want to find the line parameters that minimizes the sum of distances from each data points. There are also other variants and advanced algorithms like RANSAC that can handle more complicated situations like outlier. Here we assume the filter of Lidar has done a good job, and use least-square algorithm to calculate the straight line parameters. We adopt oblique interception to parameterize straight line.

$$y = kx + b$$

For all the data points, we first calculate the y, x using angle and distance data. And formulate the problem into matrix form:

$$\min_{B} ||Y - XB||^2$$

Where B is a column vector contains k, b. This problem has closed-form solution.

$$\hat{B} = (X^T X)^{-1} X^T Y$$

With the efficient matrix operation support from numpy, we now have a fast and reliable line fitting algorithm.

2.1.3 PID Control

Author: Zhenyang

PID control is one of the most common and effective controllers used in industries. It is a feedback controller, so we need to determine which feedback signal we want to use in the control stack. In the wall-follower task, we extract line information from LiDar data and want to make the car aligned with the wall which means the projection of the car should aligned with the desired line and steering angle relative to that line should be zero. So we design a two stage PID controller for steering. For the first stage, we assume the distance error is relatively large, at this stage, we use only distance error as input signal and design a PD controller to output steering angle. If the distance error is less than certain threshold, we will switch the controller to the second stage. Because vehicle is a under-actuated system which means it cannot achieve any state in certain time given any possible control input. Thus, it is hard to avoid the car turning back and forth when the distance error is small. To mitigate this problem, we also introduce steering angle error as control input and integrate the angle error to create smoother turning behavior near the line. We test our method in experiments which will be presented in experimentation part. But due to the hardware issues and time limitation, we fail to implement and test the second stage of the controller.

```
Initialize P$_{gain}, I_{gain}, D_{gain}$
**if** $|d_{error}|$ $\geq$d$_{threshold}$ **then**
    STAGE 1
    steering angle $= P_{gain}d_{error} + D_{gain}\dot{d}_{error}$
**else**
    STAGE 2
    steering angle $= P_{gain}d_{error} + I_{gain}\theta_{errorintegral} + D_{gain}\dot{d}_{error}$
**end if**
```

2.1.4 Pure Pursuit

Author: Zhenyang

When we test our fist stage PID controller in simulation, we realized that it might fail in corner cases such as initial position is far from the desired distance with large offset orientation. In those cases, the PD controller will command a steering angle constantly and lead to car going circles. To fix this problem, we adopt the pure pursuit steering model introduced in 6.141 lectures. We first set the target distance L_1 which indicates how sharp the turn would be to approach the target desired distance. To further account for the velocity and steering relationship, L_1 can be set to a function with variable of velocity. When the car goes faster, the target distance will increase to generate smoother motion for the car.

Here, we assume the error angle is small, and linearize the expression of the commanded acceleration. Finally we have this expression:

$$a_{scmd} = 2\frac{V}{L_1}(\dot{d}_{err} + \frac{V}{L_1}d)$$

The final step is to convert the desired centripetal acceleration to actual steering angle. To finish this conversion, we model the vehicle kinematic using a simple bicycle model: Ignoring the side slip of the tyres, we can calculate the

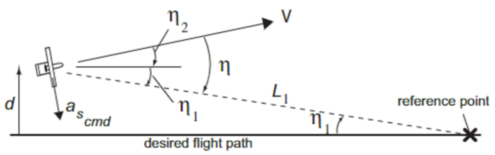

Figure 4: Pure Pursuit steering model

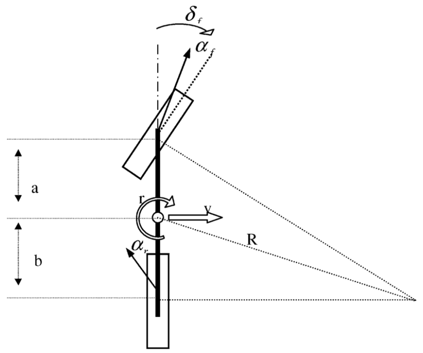

Figure 5: Bicycle model of vehicle dynamics

desired turning radius with the current velocity and a_{scmd}. Also noticing that, the steering angle is a function of the turning radius, we can calculate the desired steering angle combining the pure pursuit model and simplified vehicle dynamics.

2.2 Safety Controller

Author: David
Once the wall follower was implemented, we were tasked with creating a safety controller that would create a fail-safe measure for the wall follower to prevent it from colliding with the wall in the case of some failure. This would prevent any damage to the expensive on-board equipment. However, the precise details of what was to be implemented were left vague. It was unclear whether the car should be made to rectify its path and continue following the wall or simply stop. Our team used this ambiguity to our advantage.

For the implementation, we had two main potential ideas:

1. The first was to prevent the wall follower from giving the car a command that would lead to a collision. This would require either giving some distance buffer to the wall follower or predicting when is the proper time to override the follower's control with the collision prevention code.

2. The second was to simply stop when the wall follower would lead the car to an inevitable collision. This turned out to be the simpler solution and is what we went with for the final design.

Once we had the idea for what we wanted the car to do, we set out to develop the precise details of the safety controller. Though we considered a few ideas, we quickly settled on creating a mathematical formula that would predict, based on the maximum steering angle of the car, whether it could avoid a collision. We decided this was the best option for its potential for straightfoward implementation and slow-growing computational complexity.

2.2.1 Algorithm Statement

```
$L \leftarrow 0.325$ m
$\nu \leftarrow 0.34$ rad
$d \leftarrow$perpendicular distance between the car and the wall
$m \leftarrow$slope of the estimated wall in the car frame
$a_1 = 1 + \frac{m}{\sqrt{m^2+1}}$
$a_2 = 1 - \frac{m}{\sqrt{m^2+1}}$
$d_1 = \frac{L}{\tan \nu} a_1 + L \tan\left(\frac{1}{2}\nu\right)$
$d_2 = \frac{L}{\tan \nu} a_2 + L \tan\left(\frac{1}{2}\nu\right)$
if $d_1 \geq d$ or $d_2 \geq d$ then
  STOP
end if
```

2.2.2 Mathematical Derivation

In this derivation
Before we begin we must first define some relevant quantities in the system. Note that for this whole derivation, we are assuming all the quantities are from the frame of the car, which is at the origin.

Let m and b be the slope and y-intercept of the detected wall. The vector v is the velocity of the car. The wall vector $\mathbf{w}$ is the vector perpendicular to the wall which is defined as

$$\mathbf{w} = \begin{pmatrix} -m \\ 1 \end{pmatrix} \tag{1}$$

Using the dot product, we know that

$$\mathbf{v}.\mathbf{w} = |\mathbf{v}||\mathbf{w}|\cos(\nu) \tag{2}$$

where ν is the angle between the car and the wall. Now consider the diagram below
Given the maximum steering angle of the car, we know that it can turn to avoid a wall at a minimum circle with radius R. We note that the line starting from

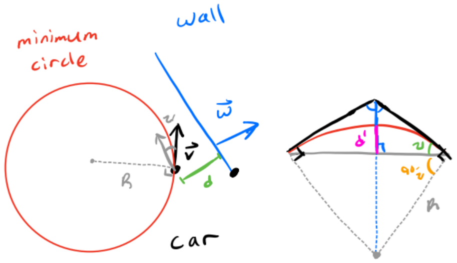

Figure 6: Sketch of the system with parts

the car parallel to the wall cuts a secant through the circle. We label the sector of the circle protruding from the secant toward the wall as having height d'. As you can see, once d' exceeds d, the perpendicular distance between the car and the wall, a collision is imminent. Using our knowledge of secants, we know d' is

$$d' = R - R\sin(90 - \nu) \tag{3}$$

$$d' = R(1 - \cos(\nu)) \tag{4}$$

Using equation (2), we know that

$$\cos(\nu) = -\frac{m}{\sqrt{m^2 + 1}} \tag{5}$$

By the definition of steering angle, we know that $R = \frac{L}{\tan\theta}$ where θ is the maximum steering angle of the car and L is the length of the car's base. Substituting that into equation (4) gives

$$d' = \frac{L}{\tan\theta}(1 + \frac{m}{\sqrt{m^2 + 1}}) \tag{6}$$

Although this is almost complete, all of the calculation are done from the rear axle of the car. We need a correction term K to account for the fact that the front of the car protrudes out from this circle. By the Pythagorean Theorem, we know that

$$R'^2 - R^2 = L^2 \tag{7}$$

Where R' is the distance from the center of the turning circle to the front axle. Since K is the correction term, $K = R' - R$. This together with equation (7) gives,

$$K(R' + R) = R'^2 - R^2 \tag{8}$$

$$K = \frac{L^2}{R' + R} \tag{9}$$

$$K = \frac{L}{\frac{1}{\tan\theta} + \frac{1}{\sin\theta}} \tag{10}$$

$$K = L\tan\left(\frac{1}{2}\theta\right) \tag{11}$$

Thus, this gives our stopping distance

$$d_{stop} = d' + K = \frac{L}{\tan\theta}(1 + \frac{m}{\sqrt{m^2+1}}) + L\tan\left(\frac{1}{2}\theta\right) \tag{12}$$

which we use in the safety controller algorithm.

3 Experimental Evaluation

As a result of various hardware and technical issues, we were not able to conduct experimental trials. Instead we describe our methodology and set up for the empirical tests we conducted of the wall follower and safety controller.

3.0.1 Wall Cases

We ran several tests to verify the accuracy of the wall follower in different physical testing situations and different velocities.

- Inner corner at 1 m/s - The car did very well at following the pure pursuit trajectory stably
- Outer corner at 1 m/s - Just like with the inner corner, with nice enough geometry
- Inner corner at 2 m/s - The car went smoothly around the inner corners, thought it would get very close to the wall
- Outer corner at 2 m/s - Going around the corner at higher speeds typically left the car going very wide and once it had turned the corner the path would be unstable for a long time. It would oscillate around the pure pursuit trajectory for a bit instead of smoothly approaching it
- Narrow corridor at 1 m/s - At 1 m/s the car was always able to navigate the corner
- Narrow corridor at 2 m/s - At 2 m/s the car was only able to successfully complete the turn at certain orientations entering the turn. Otherwise it would get too close the wall and activate the safety controller.

3.0.2 Safety Controller

The experiments designed to test the safety controller's effectiveness were based on how the controller performs under different stopping conditions. We divided the test situations into:

- Static situations, driving into an existing obstacle/wall at 1 m/s and 2 m/s. We also planned to drive towards the wall starting from different initial distances, measuring the total stopping time and the distance it took to stop at different velocities.
  - In static situations, as long as the car was going slow enough to detect the wall as it entered it's field of detection in time, the safety controller worked as expected.
- Dynamic situations, such as a person walking in front throwing a box in front of the robot.
  - As in the static simulations, if the car controller was given enough time to break and the Lidar read distance was high enough it did stop successfully.

Although we did not get to analyze the data, we hope to continue working on this to improve our wall follower and safety controller.

4 Conclusion

Author: Meenakshi
Over the course of lab 3, we were able to learn about the workflow of the physical race car system and gain experience working with ROS and sensor data in real time. We developed and implemented a wall follower that first used PD control and then later implemented a pure pursuit system for steering. We tested the wall follower in different situations and were able to successfully follow both the left and right wall while turning inner corners. In the future, we hope to test more in edge case situations such as outer corners and irregularly shaped corridors.

We were able to test the first iteration of a basic safety controller that stops based on a minimum distance in front of the robot at low speeds. Finally, we prototyped and tested a safety controller that functions by taking into account the car's maximum steering angle and stops if the projected arc intersects with the wall. Due to hardware constraints, this safety controller was tested in simulation and prevented the simulated robot from crashing into the wall when approaching it from different angles. We were also able to empirically test our new safety controller and have it stop in some dynamic situations such as a person walking in front of it. However, the consistency and tuning of our controller remains.

Our team also gained experience troubleshooting different hardware issues, which caused disruptions to the testing and iteration process. As a result, we plan to continue testing the wall follower and safety controller on the physical racecar at different velocities and dynamic stopping conditions in order to improve the robustness of the controller. In the future, we hope to compare the different iterations of our wall followers and safety controller and collect and analyze experimental data to compare the performances of our wall follower and safety controller. Despite the technical issues we encountered along the way, through this lab our team learned how to adapt to the challenges that occurred and use empirical testing data to make improvements to our wall follower and safety controller on the physical robot.

5 Lessons Learned

Presents individually authored self-reflections on technical, communication, and collaboration lessons you have learned in the course of this lab.

5.1 Meenakshi

This is the first time that I have worked with a team that was assigned instead of chosen, so it was a very new experience for me. I feel incredibly grateful that my teammates are so hardworking. I think in terms of collaboration, we're off to a good start but there is always place for growth. I appreciate how communicative we are with each other via group chat. I think the biggest challenge that I felt over the course of the week was the amount of time that was required outside of lab hours. It was difficult to pick up where a fellow team member left off at times and so we need a better documentation system going forward to avoid disruption of work flow.

In terms of technical learnings, I became more familiar with the workflow of the physical racecar system. I was able to log into the car, edit files, run the different teleop and autonomous controllers, and record rosbag files of our car in action. I also learned how to monitor useful topics and troubleshoot various hardware issues, such as the joystick connectivity issues that were happening. Lastly, I started setting up the website and became more familiar with the layout of the website and some of the web development skills needed to maintain such as css.

5.2 Zhenyang

My teammates are smart and reliable, so we manage to deliver some results under the circumstances of frequent hardware failures. But we can also do much better. We don't have a goal that is clear enough, and we don't plan and divide the work very well. For the next lab, we might want to set a quantitative metric to evaluate our future result. And with that result and target in mind, make the timeline of the whole week clear. This can also help divide the work. In short, we want to learn to work in a more systematic and structured way.

For the technical side, I think the method and implementation are generally good, but still room to improve. We can have more discussion on the technical issue. And also, getting familiar with the course material and asking TA for help can be an effective way to finish the task. These also lead to better and more productive technical plan.

5.3 David

Throughout this lab, we worked as a team to create the wall follower and safety controller. This required a lot of coordination and communication, some of which went very well, some of which we're actively trying to improve and I've been learning a lot about working with a team. This includes how to effectively delegate work and create meeting time. We have since created a schedule and been more clear about reporting any work that's done, pushing relevant code, and setting up meetings.

For the technical aspects, I also learned more about working with the ROS environment to improve the wall follower, but also how good simulation performance might not translate to good real-world performance. For example, since I was the primary designer of the safety controller, I got to experience developing a theoretical model and then transferring and debugging that model on the actual robot.

5.4 Kwadwo

The team was always ready to lend a hand in making sure the lab turned out great. I feel we can do much better with fewer hardware failures and a better division of labour as it seemed like half the team was working on things in such a way the other half could not start working if they didn't finish. We also need to make a greater effort to record runs of the lab since we had many instances were we had run tests but collected no data on them.

For the technical aspects I learnt more about car control than I ever wished to know. I have experience with tuning models for the robot and what variables change what behaviour and how to trade things for different performances. I am more familiar with the robots Linux base, transferring files with SCP, editing parameters during testing and generally navigating a file system from terminal. I need to get better with GitHub though.