Lab 3 Report: Wall Follower and Saftey Module on the Robot

Team #13

Kaleb Blake
Lucian Covarrubias
Seth Fine
Jesse George
Ivory Tang

6.141 Robotics, Science, and Systems - Spring 2022

March 4, 2022

Editor: Seth Fine

1 Introduction

By: Lucian Covarrubias

For this lab, we were tasked with setting up the robot and implementing a PD wall follower and safety system for the robot. These tasks laid the groundwork for the rest of the semester, as being able to quickly and effectively interface with the robot is going to be crucial for later labs where we will be implementing very complex systems. When setting up the robot, we had to make sure that all hardware and software were functional, and get familiar with the development cycle for sending files to the robot. To accomplish this task, we made sure to inspect every piece of hardware on the robot, and also test out the features that should be existing already, like TeleOperation and ssh capabilities. For our wall follower, we had to implement a system similar to the wall follower from lab 2, but much more robust to handle extra noise and the intricacies of a physical system. To this end, we combined two of our existing wall following simulation codes and performed extensive testing to tailor our coefficients so that our robot's motion was smooth and stable.

2 Technical Approach

By: Lucian Covarrubias, Jesse George, and Ivory Tang

The robotic racecar platform we used throughout our testing was equipped with a Hokuyo UST-10LX laser rangefinder. The laser scan data from this sensor was the main input for our algorithms that autonomously controlled the racecar's movement through motor commands. The first algorithm, the wall follower, enabled it to maintain a certain distance from the wall on either the left or right and travel along its length. The other algorithm was a safety controller that detected the appearance of obstacles too close to the front racecar and halted movement to avoid a crash while overriding the drive commands of the wall follower. We will explain our approach with implementing these two algorithms below.

2.1 Wall Following

By: Jesse George

The main goal of following a wall accurately and reliably was initially pursued by porting over a wall follower algorithm that previously verified in simulation using fake laser scan data. After running the algorithm and identifying undesirable behavior, our team made corrective adjustments to increase the efficacy of our experiments with the main metrics of improvement being less average error, less variability, and stability in undesirable circumstances. Several cycles of this process were repeated to create a wall-following algorithm that accurately followed a side wall at an a programmable distance, with robustness to sudden wall shape or orientation changes (corners, bends, gaps, etc.).

The core principle behind our wall follower implementation was the segmentation and fitting of planar laser scan data to create approximations of the wall the robot was meant to follow. Depending on the value of a set parameter, the robot would either analyze a portion of the left or right sections (relative to the robot) of the laser scanner field of view and fit a line to the points it detected. This line was the "wall" the robot would attempt to keep a consistent distance from while moving forward. PD control was employed to track this error and achieved satisfactory results when properly tuned after successive trials.

In order to avoid collisions due to rapid changes in the wall shape, a slim angular range of the frontal points of the laser scanner on the robot were fitted to an additional line to identify a "front wall". This "front wall" would cause the robot to turn if it was detected to be within a certain range, assisting the robot in avoiding the corners and pillars that it could run into with a more naive approach.

2.2 Safety Control

By: Lucian Covarrubias and Ivory Tang

The purpose of the safety controller on our robot is to prevent unexpected collisions with obstacles. If an obstacle falls in front of our driving robot, we want to be able to apply an emergency stopping function so that we can avoid damaging the vehicle.

Initially, we wanted to implement a safety bubble that could change shape and size with the velocity and steering angle of our robot. We would model our safety bubble as a parabola around the front of the car, which would grow longer at higher velocities and rotate along with the steering angle of the car.

A dynamic safety bubble has multiple advantages over a static bubble:

- Allows our robot to take tight turns without unnecessarily activating the safety feature
- Gives the robot ample time to stop when moving at high velocities
- At low speeds, doesn't interfere with normal motion

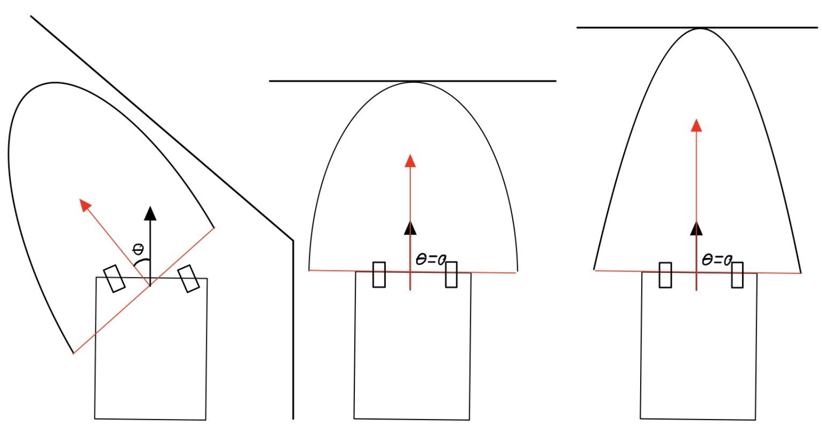

Figure 1: Dynamic safety bubble at angle, low speed, and higher speed.

As can be seen in **Fig. 1**, the ability to turn the bubble is vital for taking turns at high velocities without crossing our safety bubble. Additionally, you can see how the shape of the bubble changes based on the velocity vector (red) of the car.

In order to obtain the dynamic behavior that we wanted, we needed to model the parabola that defined our bubble in terms of the velocity of our car, and use the angle that the wheels were pointed (steering angle) to determine where in our LIDAR scan to compare distance values to the parabola.
Firstly, we need to define some terms.

- x_w^2 is the desired width of our parabola, which is a constant.
- v is the current velocity of our robot, also treated as a constant.
- a is the scaling factor for the parabola.
- b is the vertical offset from the car to the parabola.

With these definitions, we can now formulate an equation for our safety bubble:

$$y = ax^2 + b$$

where

$$b = .2v \text{ and } a = \frac{-b}{x_w^2}$$

This formulation allows the parabola to change height without becoming so wide that it interferes with the normal operation of the vehicle.

Now that the safety bubble can change size, the values of the parabola in cartesian coordinates are converted to polar form. These polar form values can then be efficiently compared with any 180° segment of lidar data to see if the points are within the boundary. To account for the steering angle of the car, we chose a 180° segment centered at our steering angle(θ).

In practice, however, we had some difficulty implementing this dynamic safety feature in our short timeframe, so in the meantime we implemented a much simpler static safety bubble which doesn't change shape at all. As seen in **Fig. 2**, this bubble was just a set radius of 20cm in the angle range of $-30°$ to $30°$. If more than 5 LIDAR readings from these angles were inside this range, we activated our stopping procedure. By setting the threshold at 5 data points, we balanced making the safety controller robust to outliers but still effective.

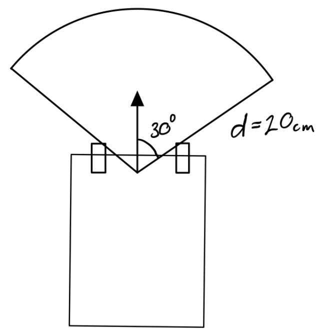

Figure 2: Static bubble in relation to the car.

3 Experimental Evaluation

By: Kaleb Blake and Seth Fine

Since we implemented wall following and safety control algorithms on our racecar, we had to conduct tests to evaluate their performance. In order to test our wall following algorithm, we conducted two trials. In both trials, we set the desired distance to the wall to be 0.5m and the velocity of the racecar to be 0.5 m/s. The wall the racecar followed caved in by about 0.15m for about a 1m length before its corner. We let the racecar follow the wall all the way to the corner, and then past the corner for about 1m. We placed the racecar about 0.5m from the wall to begin. Then we activated the wall follower until the racecar made it to our predetermined finish line. We tracked the error, which is the difference between the current distance from the wall and desired distance from the wall. This identified if the wall follower algorithm was pushing error towards zero over time, meaning our algorithm is behaving successfully. The results from the two trials can be seen below.

From the graph of Trial 1, we see that the magnitude of the error is able to stay below 0.25m for the duration of the trial with little variance. In Trial 2, the error is also able to stay below 0.25m for the majority of the trial with some areas of spiking and variation. We believe these areas were caused by disturbances to the LIDAR, which would make our linear regression wall detection inaccurate then causing the current distance from the wall calculation to be inaccurate. The racecar was able to recover from the spikes in error.

Our tests on the safety controller were less quantitative than our tests on the wall follower because it is easier to tell if the safety controller is functioning properly. We only needed to see if the wall follower could perform normally with the safety controller active, and see if the robot would stop if an obstruc-

Trial 1 Error

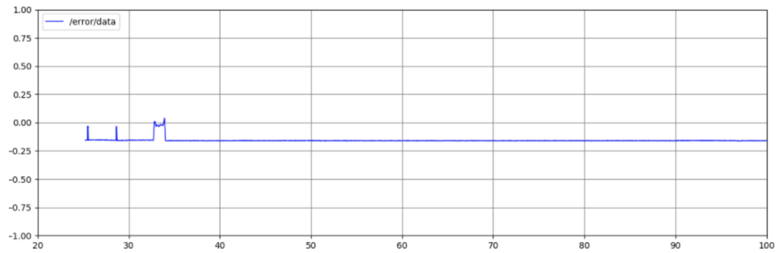

Figure 3: Error [m], which is the difference between the current distance from the wall and desired distance from the wall, at every timestep during trial 1. This measurement was made to evaluate the functionality of the wall follower algorithm.

Trial 2 Error

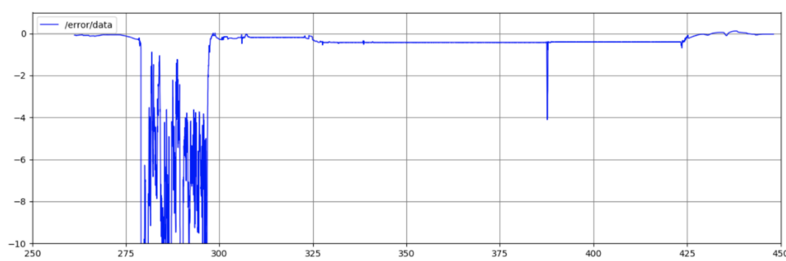

Figure 4: Error [m], which is the difference between the current distance from the wall and desired distance from the wall, at every timestep during trial 2. This measurement was made to evaluate the functionality of the wall follower algorithm. There are much more spikes in error in this trial, presumably caused by LIDAR disturbances.

tion was encountered at any time. The wall follower tests above were conducted while the safety controller was active, so we know the wall follower performed adequately with the safety controller. Then to test the the performance of the safety controller itself, we had a teammate place their foot in front of the racecar during a separate wall following trial from the two described above. An image of this safety controller test is shown below. The racecar was able to stop once our teammate placed his foot in front of it, and once he took his foot away the racecar began wall following again.

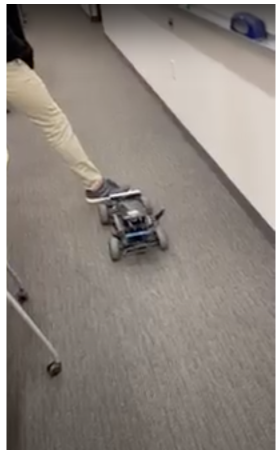

Figure 5: Our teammate placed his foot in front of the racecar while it was executing the wall follower to test our safety controller performance. The racecar successfully stopped in front of our teammate's foot.

Our tests of our wall following and safety control algorithms showed that they are functional together. Our wall follower can keep the desired distance error magnitude below 0.25 m and is robust against possible LIDAR disturbances. Furthermore, our safety controller does not interfere with our wall follower, even when it takes on close corner turns, and it can successfully stop when obstructions are dangerously close to being hit. Once there are no obstructions the racecar begins its task again.

4 Conclusion

By: Ivory Tang

Thus far, we have successfully set up our racecar, implemented a version of the wall follower that works in real life, and enacted a simple but working version of a safety controller. This covers all the basic requirements of this lab, as the main goal was to get familiar with connecting to the racecar and implementing some basic autonomous wall following. We additionally linked our github repository to the racecar while designing and implementing a more advanced version of the safety controller; however, we were not able to fully debug this advanced implementation. Finally, our team gave our first briefing and received feedback on improving how we communicate our results.

Although the core aspects of this phase have already been laid out by our current progress, there are a few more tasks that can be completed to ensure smoother development in subsequent labs. Firstly, the safety controller we implemented was quite basic in that it did not take into consideration angle with respect to wall or initial velocity. This could be problematic in future situations where the racecar is traveling at higher speeds or turning sharp corners. While we tested a few different scenarios in which the safety controller did its job, there are definitely a lot more edge cases that our current controller would fail at. Secondly, our wall follower could be improved further. The current version appears to be robust at following a smooth wall and turning 90 degree corners. It can also correct itself if at the wrong angle or too far or close to the wall. However, in case of irregularly shaped walls or breaks in the wall, the lidar will not predict the wall correctly. With more time, we could definitely improve our wall following code. Lastly, regarding the communication aspect of this lab, we could improve upon our presentation and video making techniques. Instead of simply recording one trial that we did, we could display the data across multiple trials. Using markers such as arrows or circles could also point out more clearly for the audience what is being shown in each video demonstration.

The team's efforts this lab will sufficiently prepare us for the next phase of this project. In future work, we will use the camera to enable the racecar to autonomously park in front of a colored cone and follow a line. The concepts learned in wall detection will certainly be applicable to the line following portion of this work. Additionally, slowing the racecar to park in front of the cone is similar to our safety controller implementation in which we take into account the speed and direction of approach to determine the right time and rate for deceleration.

5 Lessons Learned

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

Kaleb: In completing this lab one of the most important skills I got to practice was problem-solving. There were many times our team had problem-solve different situations including dysfunctional hardware, ROS and rviz intricacies, and code debugging. In addition to problem-solving, I learned many to use new software tools such as ROS, Git, Linux, and docker. Aside from technical skills I also had practiced by communication and collaboration skills. I made sure to be highly communicative with my teammates and listen to them when they communicate with me, which helped with the problem-solving a lot. Splitting up the tasks to get completed in parallel improved our efficiency greatly, as well. Our team was very committed to getting our robot to perform the wall following and safety task well so they can be useful if needed in the future, which helped everyone to be very thoughtful and contribute ideas until a successful one was formed.

Lucian: Through this lab, I learned about how to effectively interpret lidar data and convert between Cartesian and Polar operations. I also learned that working with a real system involves much more noise, which makes the use of high quality linear regressions even more important in order to effectively understand our environment. Additionally, I learned more about how the race car works, such as how to effectively ssh into the car and send files via rsync. I additionally learned how to set up ssh keys for the robot so that we could simply pull from the cloud to update our robot.

The communication skills I've learned through this lab are how to effectively divide up work so that we can complete tasks in parallel. We split up the wall following and safety measures, along with testing and taking data so that they could be completed as effectively and quickly as possible. We additionally had to navigate remote collaboration because we couldn't meet in person for the first portion of the lab, so we made sure to emphasize active participation in communication.

Seth: It was very exciting to be able to work with the physical robot cars for the first time this lab. However, this also led to great realizations that such code that works in simulated environments with very controlled parameters and limitations does not perform well in the real world which is inherently more random. I saw how I could have better generalized many parts of my wall follower code that was made too specific to work well in the simulator and can utilize this generalization style in future labs in order to make changes easier to debug. Additionally, having to integrate Git with scp commands to transfer code to the car solidified my understanding and working knowledge of Linux and specifically the scp command. Lastly, our team communicated well throughout the lab development cycle, but we learned that coming together, raising potential issues, and collaboratively debugging earlier in the development cycle should make our team more robust in handling multiple deadlines in the future.

Jesse: This lab was an excellent opportunity for me to work hands on with a real system that had complex issues. For example, the processing delay of running the callback function for our wall-follower had a significant effect on it's performance, making it necessary to remove any components that were too time-consuming. Despite being a relatively simple problem to fix, the symptoms of this problem were not apparently obvious at all times, since occasionally the script would run as intended with no issues. It reminded me that often times there are many more levels of complexity to consider when adapting software from a purely simulation based perspective to a real-world environment one.
Effective communication with my teammates was also incredibly crucial for this lab assignment, since there were often many parts that needed to be dealt with at once. By working together efficiently by parallelizing certain tasks and assigning each other certain roles, the process of completing the assignment became much easier.

Ivory:
After completing this lab, I learned how to set up the racecar so that we can control it using the joystick or autonomously with our code; additionally, I learned to debug various errors in controlling the real life robot that were absent in the simulation. I also learned important skills in teamwork and collaboration such as how to divide up work efficiently as to fully utilize the different skill sets on the team and also to avoid interference between each other's tasks.

My understanding of ROS and the wall follower code as well as my github skills greatly improved after successfully implementing the real life wall follower. Figuring out how to ssh into the racecar and move files between my local machine and github and the racecar gave me a better understanding of how the different pieces fit together. I had a fairly good grasp of the wall follower implementation through the previous lab, but having to tune for different parameters to account for the real environment changes gave me an even better understanding of what each constant term of the PD controller did. Finally, being able to link github directly to the racecar and merging our team's changes gave me greater confidence in my github skills that will undoubtedly be useful in the future.

I learned the importance of constructive feedback and cooperation through this lab. One example is when we realized all of us had a slightly different wall follower implementation. In trying to modify our code to work on our real life wall follower robot, we had to communicate with each other and understand each other's implementations. We were very careful to take into consideration everyone's opinions on what we should change; this also involved a good amount of explaining our own code and listening to other people explain their code.

Overall, I was satisfied with my contributions of offering my wall follower code as a baseline, tuning the parameters to work in real life, coming up with the more involved (and also simplified) version of safety control, and actively participating in all the team discussions. In future labs, I would like to work on starting the lab earlier than we did this time, improving our presentation slides to be more explicit, and being even more open about exchanging ideas between team members.