University of California, San Diego

Halıcıoğlu Data Science Institute

Data Science Capstone

Final Report

Neghena Faizyar Garrett Gibo Shiyin Liang

Domain Mentor: Dr. Jack Silberman

Abstract

Self-driving vehicles are revolutionizing the automotive industry with companies like Tesla, Toyota, Audi and many more pouring a substantial amount of money into research and development. While many of these self-driving systems use a combination of cameras, lidars, and radars for local perception and navigation, the fundamental global localization system that they use relies upon a GPS. The challenge in building a navigation system around a GPS derives from the inherent issues of the sensor itself. In general, GPS's tend to suffer from issues of signal interference that lead to infrequent positional updates and lower precision. On the 1/5th car scale, positional inaccuracies are magnified, so it is crucial that we know the location of our vehicle with speed and precision. In this project, we compare the performance of different GPS's in order to determine what level of performance is best suited at the 1/5th scale. Using the best-suited GPS, we design a navigation system that can mitigate the shortcomings of the GPS and provide both a reliable autonomous vehicle.

1/10th Autonomous Car (JACK-E)

Contents

1 Problem

1.1 Challenges of GPS Navigation

Our team's project in the context of our domain is to build an autonomous GPS-based navigation system for a 1/5th scale vehicle that will be robust and reliable enough to be competitive in a race. The challenge in doing so primarily comes down to understanding the shortcomings of the GPS being used so that its problems can be mitigated through supplementary methods. In addition to the differences in accuracy between GPS products at different price points, GPS's in general tend to suffer from issues of signal interference which lead to delays in positioning updates and lack of precision as well as oscillating data even when left at a fixed position. To build a navigation system for a vehicle that will travel at high speeds, it is crucial that vehicle positioning is provided accurately and quickly. The hurdle for our project, then, is to create a reliable navigation system using GPS that can update instantaneously and precisely despite these being the characteristic issues that plague the GPS.

1.2 Goals

In order to complete our main goal, we created several smaller tasks to complete over two quarters. These subtasks can be roughly categorized into GPS quality testing, sensor fusion, and waypoint navigation. Finding a GPS accurate enough was crucial as a navigation system built on a faulty GPS can only be so reliable. To do this, we first tested the quality of a lower-end GPS—u-blox NEO-M8N—and developed the methods needed for benchmarking GPS performance. Once we proved the necessity of a higher-end and more accurate GPS, we proceeded to benchmark the performance of the aforementioned GPS to test if it is truly a better fit for our project. Even with more accurate GPS sensors, there are still a few shortcomings that are unavoidable, such as slower frequency of messages, and lack of orientation data. To overcome these shortcomings we fuse the sensor data of GPS, IMU, and odometry together in order to get a more accurate localization system. Once we can provide reliable data to our race vehicle of its current position, speed, and orientation, we can move on to the final subtask of waypoint navigation. The main challenges in this area are determining the waypoints needed for the vehicle to effectively and efficiently navigate around a track at varying speeds along with designing the algorithm to control steering and throttle to navigate to those waypoints.

Creating an autonomous navigation system that is only reliant on GPS is essentially the same thing as driving blindfolded. While it is important to have other systems in place to determine both the position of the vehicle locally as well as the position of any nearby obstacles, we will be focusing on the core ideas behind a system that is primarily reliant on a GPS. The full navigation stack will contain a global path for the vehicle to navigate around a track with zeros obstacles and perfect accuracy. While this path can be computed independently of the vehicle on the track, the fully integrated navigation system will also need a local trajectory system that will use sensors such as cameras and lidars to make sure that the vehicle both stays on the track and avoids any collisions.

It is evident that our sub project requires a lot of expertise in areas beyond GPS, so it is important for our team to collaborate with the teams working on IMU and path planning as we build GPS-based navigation. We will need to integrate our work with every other teams' in order to create the full autonomous race vehicle.

2 Methods

2.1 GPS Comparison: NEO-M8N vs. ZED-F9P

In order to compare the performance of the u-blox NEO-M8N that we tested last quarter against the u-blox ZED-F9P that we received this quarter, we replicated the testing procedures we previously designed on the new GPS. So, again, we have chosen the Circular Error Probable (CEP) and 2D Root Mean Square (2DRMS) to be the evaluation metric used to compare these two GPS units.

Besides allowing an ease of comparison between our previous and current work, CEP and 2DRMS are also good measures of accuracy when it comes to evaluating the performance of a stationary GPS. CEP represents the accuracy radius from a ground truth coordinate 50% of the time and 2DRMS is an accuracy radius for 95-98% of the time. Since we do not have the equipment to ascertain the ground truth, we will be using an average of all the coordinates as our ground truth. We chose to use the mean of our data as a substitute for the ground truth because the mean of a sampling distribution (the collected GPS coordinates) should be able to represent the population mean (the true coordinate).

To compare the NEO-M8N with the ZED-F9P, we reused the data we collected on NEO-M8N last quarter and collected ZED-F9P data by leaving in a stationary position for approximately one hour at a rate of 4 Hz, or 4 coordinates per second. We chose to test the ZED-F9P in two different location types for this — in a neighborhood and in a park. This serves to gauge the accuracy within a setting that has obstacles which can cause signal interference, such as buildings, compared to that of an unobstructed and open area respectively. Since it is a well documented problem that the environment around a GPS can drastically affect its performance, testing the ZED-F9P in multiple locations will give a clear picture of its overall limitations.

2.2 Sensor Fusion

In order to navigate with a GPS, an accurate vehicle position and direction are needed at all times. By integrating odometry, IMU, and GPS data together, we can overcome the weaknesses of each individual method to obtain a position and orientation that is overall more accurate. To accomplish this task, we used an Extended Kalman Filter (EKF) that takes in the positional information from the GPS, the acceleration and orientation data from the IMU, and the velocity measurements from our odometry and IMU. EKF is a state estimation algorithm that consists of a prediction step and an update step that measures the actual state. Estimates for the new current state of the vehicle are improved with the previous estimate and previous measured actual state and this continues in a cycle until the uncertainty between predicted state and actual state converges to zero. Using an Extended Kalman Filter allows us to filter out noisy readings from odometry, IMU, and GPS and provides us with an estimate of the current vehicle location more accurately than these three sensors can provide on their own. In addition to filtering out the noise, the EKF allows us to directly combine the three separate data streams into a single source that contains both an accurate position and orientation.

2.3 Navigation

Navigation between a series of waypoints is the concluding subtask of our project which allows us to tie into the overall goal of creating an autonomous racing vehicle. For an autonomous system that is based purely on GPS, the first requirement is to obtain a detailed map of the desired track to race on. This navigation map is a binary masked image where white indicates allowable driving area and black represents non-drivable areas. For the Thunderhill Raceway track in Willows, CA we obtained high resolution satellite images that served as a basis for a navigation map. From this satellite image, a combination of both automatic and manual image processing techniques can be used to create the final binary mask. Once a mask is created, it is possible to implement navigation algorithms that will generate a feasible path for the vehicle. In order to test the vehicle's capabilities, we also created a map of the UCSD tent track in Warren Mall.

The navigation stack can be divided into three components: path planning, waypoint determination, and waypoint navigation. The main goal of the path planning step is to generate a viable map for a vehicle to utilize given a provided map. Our primary objective was to test the GPS's effect on navigation methods, so a path was generated by logging vehicle position as it was manually controlled. One of

Figure 1: Thunderhill Raceway Satellite Image and Navigation Map

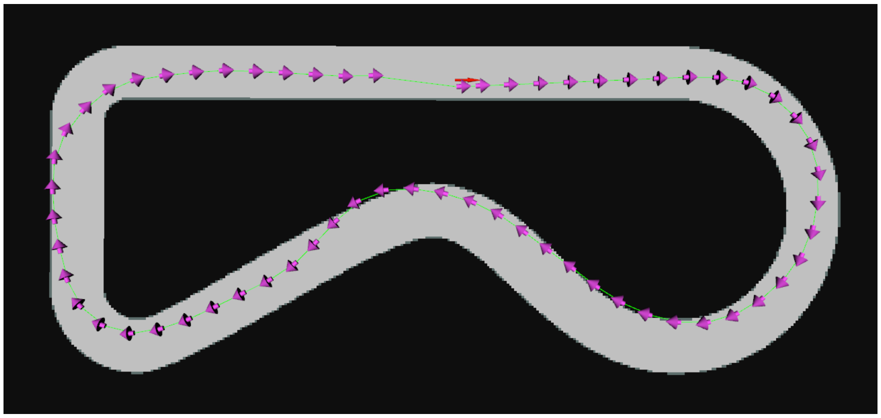

Figure 2: UCSD Warren Mall track map with Global path and waypoints

the major challenges in waypoint determination lies in the noisy nature of a GPS. Even if the vehicle reaches the exact position of the waypoint, the localization output may not match the waypoint coordinates, which could result in the vehicle going off-track or even turning around. To circumvent this issue, we set a buffer radius from the waypoint coordinate such that if the distance between the current vehicle position and waypoint is within this radius, the vehicle is considered to have reached the waypoint and can head towards the next. **Figure 3** shows the overall concept of the buffer radius. The buffer radius is based on the calculated CEP/2DRMS of the GPS in use as well as the distance between consecutive waypoints in a generated path. If the generated path has waypoints that are 0.1 meters apart with a buffer radius of 0.2 meters, then the vehicle may end up skipping waypoints.

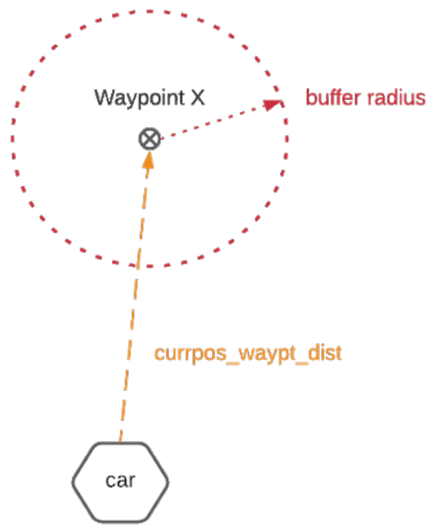

Figure 3: Diagram of buffer radius

To determine if a waypoint has been reached, we first determine a lookahead point, a waypoint that is k waypoints ahead, for each waypoint. We also calculate distance between the current vehicle position and lookahead point, distance between current waypoint and lookahead point, and distance between current vehicle position and current waypoint. If the distance between current vehicle position and lookahead position is greater than the distance between current waypoint and lookahead point, then the vehicle has not reached the current waypoint yet. The vehicle should continue heading towards this waypoint and should continuously update the distance between its current position and the waypoint to determine if it is within the buffer radius. Once it is within the buffer radius, the vehicle has reached the waypoint and can start at step 1 again for the next waypoint. This algorithm continues until all waypoints have been reached. **Figure 4** contains an overview of the algorithm.

Once there is a reliable method for the vehicle to determine if it should still head towards waypoint X or if it should now head towards waypoint X+1, autonomous navigation can be simplified into determining steering angle and velocity values to send to the vehicle. When waypoint X has been reached, a new steering angle should be calculated so that the vehicle can point towards waypoint X+1 and can now, theoretically, just travel down the path between waypoint X and waypoint X+1 to reach waypoint X+1. We do a simple check to see if the current vehicle heading is the same as the upcoming waypoint's orientation. If they are equal, there is no need to change the current steering angle, otherwise, we use a PD controller

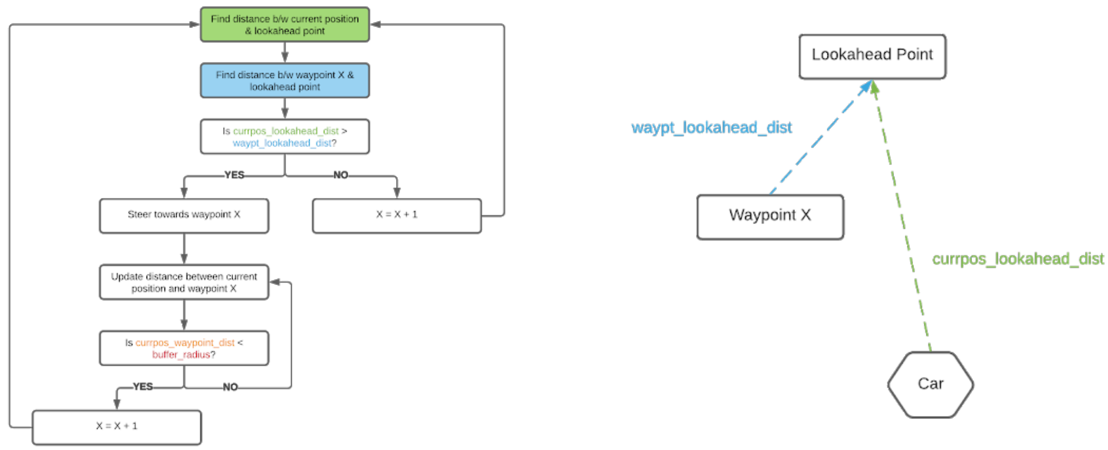

Figure 4: Waypoint Determination Algorithms and Distances Algorithm

to determine the steering angle for the car based on the heading it needs to go to. This steering angle is continuously updated based on the current position of the vehicle in relation to the upcoming waypoint.

In order to test both our navigation system and our calculations for how the GPS noise affects navigation, we created a Gazebo simulation environment of the UCSD Warren Mall track. By testing in simulation, we were able to generate a vehicle with the necessary sensors and kinematics that allowed us to evaluate overall performance of the system. Furthermore, the Gazebo environment allows for the customization of the amount of noise of the GPS sensor, so in essence we were able to change the CEP of the GPS being used on the test vehicle. By changing the CEP of the GPS, it was possible to directly observe how a fixed navigation system's performance changes based on the accuracy of the localization sensors.

3 Results

3.1 NEO-M8N vs ZED-F9P

Testing the ZED-F9P in two different locations yielded two very different results. From our previous research we knew that the NEO-M8N had a CEP of approximately 1.532 m and a 2DRMS of approximately 3.710 m. The results are shown as the first figure in **Figure 5**. When tested within a neighborhood that had obstructions such as buildings, the result was that the ZED-F9P had a CEP of approximately 3.548 m and a 2DRMS of approximately 9.050 m. The results are shown as the third figure in **Figure 5**. The results were erratic, most likely due to the strength of the GPS signal being interfered by the tall buildings. Thus, the graph of the results look visually different compared to the other two. On the other hand, when tested in an open space that emulates our race tracks more accurately, the results were drastically different with the ZED-F9P reporting with a CEP of only approximately 0.097 m and a 2DRMS of approximately 0.233 m. The results are shown as the second figure in **Figure 5**. This is a huge difference from the performance in the neighborhood with a 189% difference in CEP and 190% difference in 2DRMS. Comparing the better performing (open area) CEP to our old NEO-M8N, we find a 176.648% difference in CEP and a 177.151% difference in 2DRMS.

3.2 Sensor Fusion

To test if we can get accurate vehicle positioning and orientation, we simulated odometry and IMU data and introduced noise into the y-axis specifically. Each of the red arrows in **Figure 6** is a reading of the vehicle's positioning and orientation as the vehicle remains in a stationary point. As designed, the readings are erratic along the y-axis even though the vehicle has not moved. The blue arrow is the noisy odometry and IMU data after it has been filtered by EKF. Although it is hard to tell, there are actually multiple blue arrows stacked on top of each other as EKF filters out each noisy reading (the red arrows) to produce a consistent vehicle positioning. Compared to the raw odometry and IMU data, EKF filtered data is much more accurate to the vehicle's positioning in both the x and y direction.

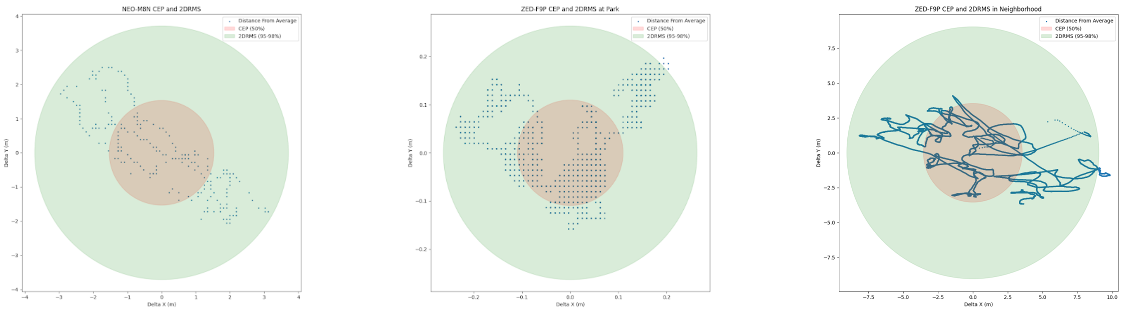

Figure 5: Comparision of CEP and 2DRMS for GPS
Left: NEO-M8N, Middle: ZED-F9P in Neighborhood, Right: ZED-F9P in Park

While we were unable to test the navigation system on finalized hardware due to integration issues, simulated tests of the vehicle proved to be successful. Over the course of a full lap around the simulated UCSD Warren track, the EKF algorithm was able to reduce the error in localization error essentially down to zero. **Figure 8** demonstrates how the simulated noise affects the predicted location of the vehicle. The purple sphere indicates possible locations for the car; thus the NEO-M8N sphere is much larger than the ZED-F9P.

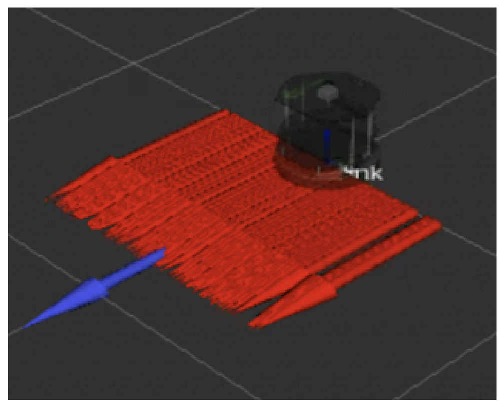

Figure 6: Improved heading through the use of EKF

3.3 Navigation

Utilizing the waypoint selection and navigation algorithms outlined above, the simulated vehicle was able to successfully navigate paths generated by the manually controlled vehicle. Testing in this way drastically reduced the need to run comprehensive tests of all conceivable paths, because we directly limited the test space to that of possible paths of the real vehicle. Provided that there are no obstacles, the path that is provided is guaranteed to be feasible, and the localization data adheres to the specifications that we benchmarked, the GPS based navigation system that we created will successfully navigate to every point supplied. For a demonstration of our vehicle completing a lap autonomously, please go to: https://neghena.github.io/GPS_Autonomous_Nav/

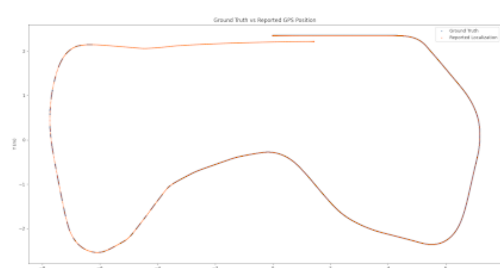

Figure 7: Ground Truth vs Estimated Position from EKF

4 Discussion and Analysis

Through our statistical and visual analysis, we are able to arrive at the conclusion that the performance of the ZED-F9P would be much more effective for autonomous navigation than the NEO-M8N at the 1/10th and 1/5th vehicle scale. The CEP and

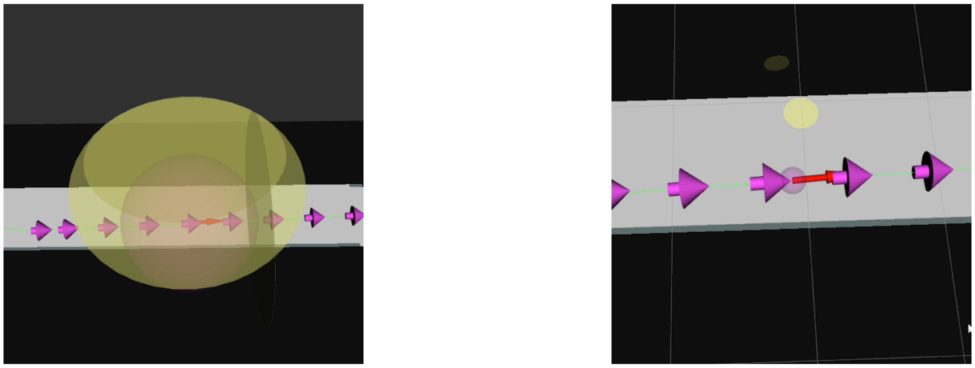

Figure 8: Left: Simulated noise indicating relative position for NEO-M8N
Right: Simulated noise indicating relative position for ZED-F9P

2DRMS of the NEO-M8N were around 1.5 and 3.7 meters respectively, which indicates that it would not be able to adequately identify what position the vehicle was at — it wouldn't be able to identify which side of the track it is on and could even possibly report that the car is off the road altogether. On the other hand, the ZED-F9P had a CEP and 2DRMS of 0.09 and 0.23 meters respectively. One of the most important parameters for autonomous racing/navigation is being able to accurately determine which side of the track the vehicle is on, so our analysis on the accuracy of the ZED-F9P indicates that it would be sufficient. It should be noted that the ZED-F9P did not perform particularly well in areas with obstructions such as buildings, however, we can deem this as negligible for our purposes since the Thunderhill track does not have obstructions which cause satellite interference.

While our localization system using the EKF was very effective in simulation with simulated noise for the required sensors, the same performance was not achieved with the physical IMU and GPS. The primary benefit of using an IMU was to obtain a heading and provide local corrections to the GPS; however, inadequate mounting of the IMU resulted in inconsistent readings of the linear acceleration data. Because the noise being outputted from the IMU was not uniform or even Gaussian the EKF was unable to provide accurate predictions and actually resulted in worse performance than the GPS alone. The UCSD track is a fairly inconsistent environment with large tents and buildings surrounding the track on all sides, so relying solely on GPS for localization would likely result in poor navigation performance. However, on the Thunderhill track, it is likely that even without the improvements provided by an IMU, the ZED-F9P would provide adequate localization accuracy for successful navigation.

5 Future Improvements

One potential path to improving our project is to use a 2 unit RTK rather than just 1 unit as this could also potentially increase accuracy while having less of a computational cost on the vehicle than other methods of improving accuracy. Our team initially made some headway on this topic during the course of our project but due to many challenges with connectivity and lack of customer support, we were unable to deliver solid results. Nonetheless, it is a path worth looking into for GPS-based autonomous navigation if we were to enter our vehicle for competitions.

Another idea to improve accuracy is utilizing different state estimation algorithms such as the Unscented Kalman Filter (UKF) for sensor fusion. EKF uses a single point, the average of all the coordinates, to approximate where the vehicle's position truly is and UKF uses multiple, weighted points to approximate. The idea is that the more points are used, the more precise the approximation will be. So, theoretically, the UKF can provide us with a more precise estimate of the true vehicle position than the EKF can. However, precision also comes at a cost as UKF is more computationally intensive than EKF. In our project, we chose to use EKF as we are designing autonomous navigation for cars on the 1/5th scale travelling on full size lanes so there is more leeway with not having the most accurate localization than there is for full-sized cars. However, if this project was to be expanded for full-sized cars, it may make sense to prioritize localization accuracy over computing costs.

Lastly, in our current project, we design autonomous navigation in the context of racing; the goal is to complete a lap as fast as possible. However, as self-driving cars become more prevalent in the automotive industry settings, designing autonomous navigation to be applicable in urban traffic settings would have a larger and more significant impact. To do so, one of the most important areas we would have to improve on would be to include safety guards for speed. Since the context of our problem was rooted in racetrack settings, we did not design any safety features such as reducing the speed when making turns to prevent accidents from happening. Thus, to help our current project be applicable in more general settings like urban traffic and for full-sized cars, we would need to do more work regarding the speed thresholds we allow our vehicle to travel at, and improve localization accuracy via GPS selection and sensor fusion to design a safe autonomous driving experience for humans.

6 Conclusion

To conclude, in order to create a GPS-based autonomous vehicle at the 1/5th scale, it is necessary to use the ZED-F9P or module with a comparable CEP. We have been able to successfully have our vehicle navigate through any predefined sets of waypoints based solely off the localization of the GPS, IMU, and odometry. Although we succeeded with our team-specific task, it is still imperative to integrate our work with other groups who are working on aspects such as computer vision/obstacle avoidance to create a well-rounded autonomous vehicle that can handle scenarios that a navigation system built around GPS can not. While we as a class did not end up integrating all our work to build one robust race vehicle, our team-specific work is capable of completing full laps around a racetrack and can be a starting point for future work towards integration.