A6 Desla Report: Autonomous Parking

Sidak Singh Arora, Anirudh Devanand, Anthony Ji, Vishal Thilak, Kei Matsui
The University of British Columbia

Abstract - **Autonomous forward parking remains a nuanced challenge in the field of self-driving vehicles, especially in urban environments where space constraints and limited input data make maneuvering difficult. While significant strides have been made in autonomous parking systems with the development of path planning algorithms using Rapidly-Exploring Random Trees and Ant Colony Systems, forward parking has received comparatively less attention than its reverse counterpart. This paper introduces a LIDAR-based autonomous forward parking system that follows a state machine framework to park a vehicle adjacent to a parked one. The system navigates through five discrete states: initialization, spot detection, alignment, maneuvering, and final positioning. Our results show consistent parking performance within marked lines, supporting reliable autonomous forward parking in constrained scenarios.**

1.0 Introduction

Autonomous parking is a critical, under-researched aspect of self-driving vehicle technology. While progress has been made in areas such as lane keeping, path planning, and obstacle avoidance, low-speed maneuvering tasks like parking in crowded, real-world environments remain a complex problem due to constrained spaces, limited perception, and no room for errors. Forward parking, in particular, introduces additional challenges, as it often requires the vehicle to navigate into tight stalls without the utility of wide sensor coverage or the enhanced field of view typically leveraged during reverse parking.

In recent years, there has been an increasing interest in intelligent parking systems, many of which utilize a combination of vision-based perception and pre-mapped environments. However, such approaches frequently rely on dense infrastructure, prior knowledge of the environment, or costly sensors, including high-definition cameras and GPS. In contrast, we focused on a more deployable approach—leveraging LIDAR to solve the problem of autonomous forward parking into a stall adjacent to a parked vehicle.

In this paper, we describe the development of a LIDAR-based forward parking system deployed on a roughly 1:10 scale car model. Our system uses a state machine FSM approach that enables the car to perform precise, repeatable forward parking maneuvers in a constrained parking lot setting. We navigate through the following discrete states: initialization, spot detection, alignment, maneuvering, and final positioning.

Unlike prior methods, which require a dense semantic understanding of the environment, our approach leverages real-time LIDAR data to identify reference objects and compute state transitions accordingly. We assume that our car is in a reasonably structured environment with at least one adjacent parked vehicle acting as a reference point for localization. Our algorithm was evaluated using a physical test suite simulating a real-world parking lot, with a series of experiments conducted to evaluate tolerance to variation in initial conditions and neighboring vehicle positions.

2.0 Background

Autonomous parking is already a challenge for human drivers, but removing humans from the equation makes the task even more complex for self-driving cars. The difficulty arises from several key factors:

- **Complex Motion Planning:** Self driving cars must compute an optimal path while avoiding collisions with obstacles such as curbs, parked cars, and pedestrians. This becomes harder with obstacles that are irregularly placed, which require advanced algorithms to dynamically adjust parameters while parking [1].
- **Sensor Limitations:** Our self-driving car relies on LiDAR data for decision making . However, this has limitations, such as limited field of view, lack of data in terms of types of objects and a limitation in the number of available sensors on the vehicle. These limitations make it difficult for the car to accurately gauge and process information accurately and precisely while parking [2].
- **Unpredictable Environment:** The car must account for unpredictable elements such as pedestrians walking between cars, cyclists, or other vehicles moving near the parking area. Additionally, there may be human errors in the cars parked in the adjacent lots, which causes the projected space to be misaligned or narrower than expected, requiring real-time adjustments [2].
- **Time Constraints:** In real-world scenarios, self-driving cars may need to park efficiently while other vehicles wait behind them. Many current auto-parking systems are slow because they rely on extensive real-time computations to ensure accuracy and avoid obstacles. This adds to challenges that make implementing a valid parking system for self-driving cars difficult [3].

Automated parking is a key innovation in the development of autonomous vehicles. As cities grow, parking congestion and inefficiencies are becoming major issues. A study states that urban drivers spend an average of 17 hours per year searching for parking, which increases traffic congestion, fuel consumption, and carbon emissions. On top of that, poorly executed parking maneuvers lead to many minor collisions, resulting in a significant number of expensive repairs and insurance claims.

Beyond the environmental and economic impact, auto-parking technology improves accessibility for individuals who struggle with parking due to age-related impairments or disabilities. By automating parking, we can reduce driver stress and enhance road safety. Suppose a future where autonomous vehicles become widely adopted, a safe and efficient auto-parking system will be essential for self-driving cars to park without human control.

The commercial benefits of auto-parking technology are also significant. Large parking facilities such as those in shopping malls, view spots, and corporate campuses can optimize space usage, improve traffic flow, and create a better passenger experience. Given these advantages, developing a reliable and efficient auto-parking system is a critical and much-needed step toward advancing autonomous mobility solutions.

3.0 Analysis

3.1 Ford’s Active Park Assist

Ford's park assist feature is a partial auto-park system. It uses ultrasonic sensors to detect a possible parking spot and then finds the optimal steering path. The system provides automatic steering, but the driver must control acceleration and braking [4].

3.2 Tesla's AutoPark

Tesla uses ultrasonic sensors and cameras positioned around the car. It calculates the best path and adjusts the trajectory in real-time using spatial awareness to avoid hitting other objects. However, the process is slow due to heavy computation, and it is limited to rear and parallel parking [5].

3.3 Aptiv's AI Parking System

Aptiv integrates AI with ultrasonic sensors and cameras to improve parking precision and efficiency. It effectively maps a path directly into a parking spot, reducing unnecessary maneuvers. It supports parallel, front, rear parking, and even autonomous valet parking (Level 4) [6].

4.0 Proposed Approach

This section outlines the step-by-step approach for implementing the auto-parking system for the F1tenth car. The development process is divided into several key phases:

1. **Algorithm and Methodology Review** (1 week): In this phase, we will study existing autonomous parking algorithms for parallel, perpendicular, and angled parking, focusing on strengths and limitations of each approach and understand trade-offs. We will explore open-source implementations as well as relevant research papers to gather proven techniques that we can implement and adapt for the F1tenth car. Additionally, we will research the additional technique requirements for auto-parking such as path planning, localization and mapping algorithms, and control strategies.
2. **Sensor Integration** (1 week): During this phase, we will explore additional sensor options to complement the current LiDAR and camera equipped on the car. We will also focus on the calibration of sensor location and orientations to ensure each sensor is properly positioned to gather data required for auto-parking. Throughout this process, we will collaborate and communicate with instructors to evaluate the available options and make decisions on which sensors to integrate into the system.
3. **Path Planning and Motion Control** (1 week): In this phase, we will develop the path planning and motion control algorithm for the car to navigate to the parking spot safely.
4. **Simulation Testing** (1 week): In this phase, we will conduct testing and debugging on the auto-parking system in simulation to ensure its reliable and safe operation. This will be done using the F1tenth car simulation, where we will create various parking maps to evaluate the system's ability to successfully park in various scenarios. Additionally, we will test the system's reaction to obstacles, environmental changes, and edge cases to ensure the system is resilient.
5. **Deployment and Optimization** (1 week): During this phase, we will deploy the auto-parking system on the physical car. The system will be tested under real-world conditions, and we will assess any aspects of the algorithm affected by environmental factors such as sensor noises and road conditions. Based on the test results, we will further optimize the system's performance to ensure smooth execution of parking.

5.0 Success Metrics

We will measure the success of our autonomous parking system using the following metrics:

1. **Alignment in Spot:**
    - The car should park reasonably within the center of the parking space.
    - Benchmark: We can compare results to existing self-parking systems such as Tesla's AutoPark and Aptiv's AI parking system.
2. **Time to Park:**
    - The system should complete parking within a reasonable time frame (under 60 seconds in standard conditions as required to pass driving license test).
3. **Collision Avoidance and Safety:**
    - There must be zero collisions with obstacles during the parking process of the car (curb, parked cars, pedestrians).
    - Evaluation: Conduct extensive testing with static and dynamic obstacles, logging any contact events.

6.0 What We Did

6.1 Left Forward Park Implementation and Algorithm

We implemented a state machine to forward park the car into an empty stall to the right of a parked car. The state machine uses 4 main states stated as follows:

State 0: Initialization: The car starts straight and is ready to drive forward to find the parking spot (see Figure 1).

State 1: Find the parking spot: The car drives forward until a parked car is detected to the left. It then moves forward to align with the middle of the empty spot beside the earlier detected spot (see Figure 2).

State 2: Align to Car: The car moves back and aligns the front wheel of the car at a ~66 degree angle to the first line of the stall (i.e, left line of the stall) (see Figure 3).

State 3: Enter the spot: Once in the proper position, the car turns left into the parking stall until it detects the wheel stop (see Figure 4).

State 4: Stop and straight: Once in the stall and aligned, the car comes to a halt and straightens (see Figure 5).

6.2 Key Assumptions and Explanations (Maybe discuss constraints here as well?)

Given the hardware limitations, we had to make 4 key assumptions, which are stated (with reason) as follows:

1. A car is parked to the left of the desired stall. This assumption was made for 2 main reasons:

    a. When entering the lane, the camera is unable to detect lines accurately. As a result, the camera misses the spot altogether, which results in the car being unable to locate the empty stall

    b. When turning into the spot, the camera can no longer see the stall lines due to the height at which it is installed. As a result, as soon as the car begins entering the parking stall, all visuals are lost, and the car has to drive 'blind'

Due to these limitations, the camera cannot be relied upon for detecting or navigating into the parking stall. Instead, we depend on LIDAR for stall detection and alignment. For LIDAR-based detection, a reference point is necessary, which is why the presence of a car in the adjacent stall (to the left) is critical for successful parking.

Justification: Our main goal was to make it easier for drivers to park in crowded parking lots. Therefore, it is fair to assume that the adjacent spot is filled and the driver has to forward park directly beside it, or between 2 parked cars (see tolerance for what scenarios we can handle)

2. The parked car in the adjacent stall is fairly centered and straight (within the tolerances defined in Section 7.5)

Reason: This assumption ensures the presence of a reliable reference point for alignment.

Justification: Since our algorithm uses LIDAR to park the car, a clear reference point is necessary. Without a reasonably positioned adjacent car, the system cannot reliably identify or align to the parking spot. A fairly centered and straight vehicle provides the spatial context needed for accurate alignment.

3. The car enters the parking lane fairly straight (tolerance stated in section 7.5). This assumption was made to get an estimate of the initial position of the car before starting the parking algorithm.

Justification: When a vehicle turns left or right into a lane, it is typically aligned with the receiving lane, ideally close to 0° relative to the center line, with slight deviations (deviation tolerance stated in section 7.5). Hence, it is safe to assume that the car enters the lane aligned to the center line with only minor deviations.

Note: We only assume the heading of the car relative to the center line, and are making no assumptions about the distance from the solid yellow line.

4. Wall at the end of the parking stall (imitates a wheel stop): This assumption was made to ensure that the car knows how far it needs to go in the stall. The reason we used a box to imitate a wheel stop was the placement of the LIDAR sensor. Since the lidar can only detect at a height of 8 inches, the wheel stop (i.e, the wall) needs to be at least 8 inches in height.

Justification: Wheel stops in parking spots are very common and are usually found in crowded parking lots. According to the 2010 Americans with Disabilities Act (ADA) Standards for Accessible Design [7], wheel stops are recommended for use to keep drivers from pulling into pedestrian aisles

6.3 Data Collection

In order to precisely assess the performance and functionality of our autonomous forward parking system, we had a systematic data collection procedure. This included having our own parking map laid out, using the data from the camera and LIDAR sensors, and having recordings. In the real world, many variables and different situations can arise, and if the car hasn't been tested for those scenarios, it may have a higher likelihood of failure. The goal for this car project was to collect as much meaningful data in order to identify if our algorithm (which was effectively a state machine) was transitioning to the different states properly, if it was parking properly in the first place (within the lines), and also to see where it fails.

6.3.1 Physical Setup

For the parking map layout, we essentially had the following. We had two parking spots, one empty spot,t and one spot with a cardboard box which mimicked a parked car. The spots were marked with black electrical tape. In addition, for the empty spot, there was a wall indicating the end of the spot. This was the test grounds we had for all of our mock trials.

6.3.2 Data Logging

In order to determine if our parking was working adequately, we had several logging metrics we logged down for each run, and the results of our key metrics and findings will be discussed in 7.5. The first of which is state transitions. We noted down when/if the state transition occurred, and if it occurred at the right time during the parking algorithm. For instance, when the car backed up, would the car identify the cardboard box as a parked vehicle and transitioned into the next state. Furthermore, for each parking run, we noted measurements of how far the parked car was from the parking line. In addition, through runs we had print statements such as the current steering angle and velocity at each second to ensure it aligns with appropriate values at a given instance of the algorithm.

6.3.3 Video Recording

The video recordings we did using the onboard camera were another part of the collection process. The camera was tilted completely to the left. This provided us with a complete view of the parking spot and was used to see how the car was aligning with the parking spot. Additionally, the video was used as a means of testing other potential algorithms, like edge detection using the camera, to identify the empty parking spot.

6.4 Evaluation Against Metrics

1. **Time to Park:** As mentioned in section 5, in most driving license exams, a time limit of 60 seconds is allocated to park the car. We achieved an average parking time of just 13 seconds, well within this limit.
2. **Collision Avoidance:** The parking process must be free of collisions. Our algorithm ensures the vehicle stops before reaching the wheel stop and avoids collisions with adjacent vehicles, provided the assumptions outlined (within defined tolerances) are satisfied.
3. **Alignment in Spot:** During parking, the vehicle should remain within the designated spot. While the wheels touching the parking lines is not considered a complete failure, we treat it as a *partial success*, since a 'good' parking outcome maintains a reasonable distance from both side lines.

Our algorithm achieves successful parking fully within the spot without touching the lines in 80% of the cases. In the remaining 20%, the wheels touch the lines, which we classify as partial successes.

6.5 Tolerance Evaluation

In real-world scenarios, vehicles are often not perfectly centred within their stalls or parked at a straight angle. To evaluate the robustness and reliability of the forward parking system, we conducted a series of experiments around three key parameters involving the position of the neighbouring parked vehicles. These tests aimed to determine how much variation the system can tolerate while still successfully parking in the stall.

6.5.1 Lateral Distance from the Target Stall
6.5.1.1 Description

To determine how the lateral proximity of a neighbouring vehicle parked to the left of the target stall affects the system's ability to recognize and complete a forward park in the stall.

6.5.1.2 Methodology

A neighbouring car was placed straight and positioned to the left of the target stall at varying lateral distances, as indicated by label ① in the figure below. The autonomous system then attempted to perform a forward parking maneuver under each condition, while keeping all other variables, such as the vehicle's initial starting position and orientation, constant.

6.5.1.4 Conclusion

When the neighbouring parked car is positioned straight, the system is able to successfully maneuver into the target stall as long as the neighbouring vehicle remains within a certain lateral distance from the left boundary of the target stall.

In Trial 5, the neighbouring car was positioned as far from the target stall as possible while still remaining within its own parking space. While the system successfully completed the parking maneuver, the vehicle's left wheels crossed over the boundary line of the stall. Since the neighbouring car was parked at the opposite edge of its own stall, this boundary crossing did not present any risk of collision. However, if the width of parking stalls were larger in a real-world scenario, such behaviour could result in the vehicle being positioned on the boundary line between two stalls.

6.5.2 Longitudinal Offset from the Top of the Stall
6.5.2.1 Description

This experiment tests how changes in the longitudinal position of a neighbouring parked car, specifically its offset from the front boundary of the stall, impact the autonomous system's ability to detect both the neighbouring vehicle and the target stall, and to perform a forward parking successfully. This setup simulates real-world parking conditions, where vehicles may either be parked deep into the stall or extend beyond the stall's front boundary.

6.5.2.2 Methodology

A neighbouring vehicle was placed in the adjacent stall to the left of the target stall, with varying longitudinal offsets relative to the top/front boundary of the parking stall, as shown by label ② in the figure below. The autonomous system then attempted a forward parking under each condition, while keeping its initial pose and lateral position consistent throughout the trials.

6.5.2.4 Conclusion

The autonomous parking system handled moderate longitudinal variations in neighbouring vehicle positions without issue. For offsets up to 8 inches, the stall was consistently identified, and the forward parking maneuver was completed. However, when the offset was over 5 inches, the system slightly overshoots during the final alignment. As a result, the vehicle crossed the left boundary slightly, and the vehicle's front right wheel ended up on the right boundary line of the stall. While the maneuver remained collision-free, this outcome suggests that excessive longitudinal offset by a neighbouring vehicle can affect final alignment accuracy.

6.5.3 Angular Deviation of the Neighbouring Car
6.5.3.1 Description

This experiment evaluates how the angular misalignment of neighbouring parked vehicles affects the system's ability to detect the target stall and perform a forward maneuver. In real-world scenarios, vehicles are often not perfectly aligned within the stall. This test simulates those conditions to assess how tolerant the system is to such variations.

6.5.3.2 Methodology

A diagram below shows the setup where the target stall is defined, and a neighbouring car is parked with an angular deviation, which is labelled ③. The vehicle is misaligned within the stall, simulating a real-world scenario where parked cars are not perfectly aligned.

6.5.3.4 Conclusion

The results show that when neighbouring vehicles, specifically one parked to the left of the target stall, are at an angle, the system's ability to detect and align with the target stall is negatively impacted. In both trials, the system was able to park within the target stall; however, the vehicle crossed boundaries, and one of the wheels was on the boundary line. These outcomes suggest that the system is more sensitive to the angular position of adjacent vehicles than to their longitudinal and lateral distance.

7.0 Conclusion

Our project demonstrated a LIDAR-based forward parking implementation that uses a finite-state-based approach. This implementation was tested heavily to test the robustness and correctness of our algorithm.

The results clearly show that our approach was able to complete successful forward parking attempts a majority of the time. We tested this under an appropriate range of different values for lateral, longitudinal offset, and angular deviation of the neighboring vehicle that was used as the reference point. For most of the test cases, we performed a successful park, and in the extreme case, our parking led to some boundary crossing. However, the system never caused any crashes and performed the maneuvers for adequate parking despite crossing the line in a few cases. This displays how effective and robust our algorithm was in different scenarios.

For the future, we would like to expand our current implementation to handle uncontrolled conditions. Currently, we have a lot of fixed variables in play, so being able to detect a parking spot (by the lines) and see if it's empty would be a step in the right direction. Furthermore, based on that information, it should map out a path for the car to the spot and be able to have real-time path changing happen in case of any dynamic obstacles.

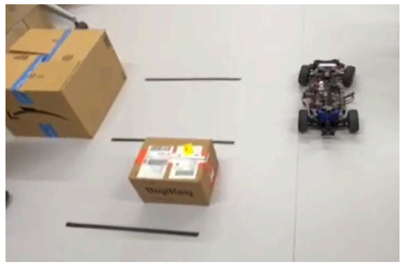
Our project serves as a start to forward parking in autonomous vehicles. Hopefully, our study can serve to help the development of forward parking algorithms in real-life vehicles.

8.0 References

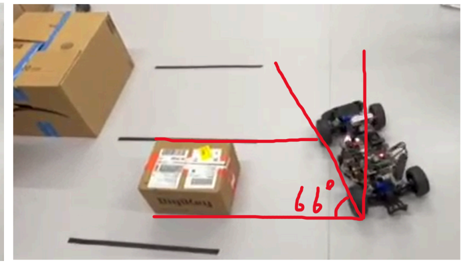
[1] Optimization-Based Trajectory Planning for Autonomous Parking
[2] A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles
[3] Smart Parking Lot Based on Edge Cluster Computing for Full Self-Driving Vehicles
[4] Ford Active Park Assist
[5] Tesla AutoPark
[6] Aptiv AI Parking System
[7] Parking Lots and Tripping Hazards

9.0 Appendix

(Figure 1)

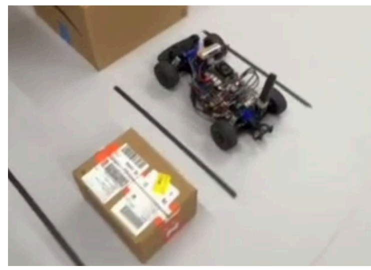


(Figure 2)



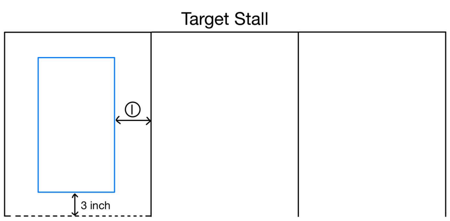
(Figure 3)

(Figure 4)

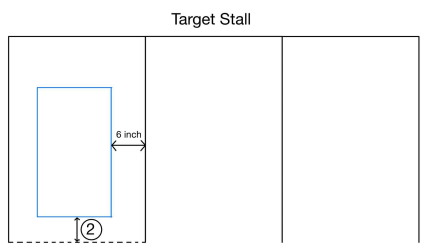


(Figure 5)

6.5.1.2



6.5.2.2



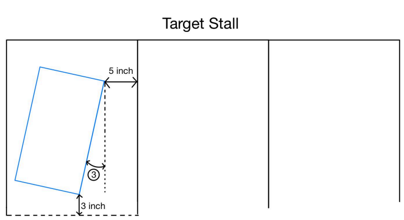
6.5.1.3 Results

Trail	Lateral Distance	Parked?	Notes
1	6 inch	Yes	Positioned in the center and daily straight
2	7 inch	Yes	Parked closer to the left boundary line and angled
3	8 inch	Yes	Centred but at an angle
4	9 inch	Yes	More angled than trail three and closer to the left boundary
5	10 inch	Partial	The left rear tire is on the left boundary line

6.5.2.3 Results

Trail	Longitudinal Distance	Parked ?	Notes
1	3 inch	Yes	Parked fairly straight
2	4 inch	Yes	Parked at an angle and closer to the right boundary line
3	5 inch	Yes	Crossed the left boundary while maneuvering
4	6 inch	Yes	Crossed the left boundary while maneuvering
5	7 inch	Partial	The front right wheel touches the right boundary
6	8 inch	Partial	The front right wheel touches the right boundary

6.5.3.2



6.5.3.3 Results

Trial	Angle	Parked?	Note
1	17.58°	Partial	The front wheel touches the right boundary when parked
2	13.73°	Partial	Cross the left boundary while maneuvering

10.0 Other Approaches and Experiments

Several alternative parking strategies were explored prior to the adoption of the LIDAR-based FSM approach. These experiments provided insight into performance trade-offs and guided the selection and adjustment of the final method. Other approaches and experiments include the Reverse Parallel-Parking Routine, Camera-Only Spot Detection, and Sampling-Based Path Planning.

10.1 Reverse Parallel-Parking Routine

A conventional reverse parallel-parking algorithm was implemented as a baseline. This procedure applied a state-machine framework, which goes through the steps of spot detection, angled reverse entry, and straightening. This routine utilizes LIDAR data, a side-facing camera for stall identification, a back-facing camera for parking adjusting and stopping, and a steering controller (installed in the car's software). This approach sounds achievable in theory; however, when it was tested in practice, the camera used for detecting and aligning with the spot produced inconsistent results under the consistently varying lighting and angles, leading to unreliable transitions between the states. Only one camera (at a not-so-flexible angle) could be provided to the car, thus making it even more difficult to achieve the full capability of the design. These equipment challenges proved the design to be impractical for the scope of this course.

10.2 Reverse Parking Routine

Similarly, the normal reverse parking routine uses the same algorithm as reverse-parallel parking, with the only difference being that it has a different turning angle and a constant stall detection camera when backing in. The main issue with this algorithm lies in the limitations of our equipment — the camera is unable to consistently detect parking stall lines while moving, and the equipped angle of the camera is also limited.

10.3 Camera-Only Spot Detection

A vision-based algorithm was designed to identify parking spots using edge detection and perspective correction. While effective under consistent lighting conditions, this method also exhibited unreliable performance with the camera's data extraction process when the perspective changed. Frequent false positives from floor tape further reduced reliability. Ultimately, the approach was abandoned due to an insufficient camera mounting angle and equipment inconsistency.

10.4 Sampling-Based Path Planning

A sampling-based planner was tested on a 2D occupancy grid generated from LIDAR scans. Although this method accommodated varied starting poses and obstacles, the extra-long planning time was incompatible with the need for real-time executions on the controller, and the LIDAR’s scan angle was also limited. The complexity and uncontrollability outweighed the benefits.

10.5 Conclusion

These experiments addressed the balance between the complexity of the algorithm, the test environment, and the performance of the sensor equipment. The LIDAR-FSM approach emerged as the most effective compromise for its reliability and efficient stall detection. Future enhancements may extend system flexibility by including improved and accelerated planning phases.