F1Tenth Autonomous Racing: Simulation to Real-World Interaction

Aman Waghchoure

***Abstract*— Autonomous racing presents a compelling arena to explore robotics, control algorithms, and integrate AI. In this research, I investigate techniques for autonomous navigation of a 1/10th scale racing car, commonly referred to as F1Tenth, specifically focusing on three algorithms—Wall Following with PID control, Follow-The-Gap for reactive obstacle avoidance, and Pure Pursuit for path tracking—and evaluate their performance in a simulated F1Tenth environment. The goal is to maximize lap speed, minimize collisions, and enable car-like robots to navigate complex racetracks. While I encountered challenges in converting the simulator directly to the physical car, the presented experiments highlight both the promise of autonomous racing techniques and the insight necessary to bridge simulation and the real world.**

***Index Terms*— Autonomous racing, F1Tenth, PID control, simulation-to-reality, path planning, obstacle avoidance.**

I. Introduction

Autonomous driving research has gained significant traction in recent years, with applications that span transportation, exploration, and competitive racing. While full-scale Formula 1 cars are incredibly expensive ranging from 10 million to 20 million dollars [1], scaled-down vehicles such as F1Tenth provide an accessible and interactive way for students and researchers curious about autonomous vehicles to develop robust autonomous systems. There are many resources the F1Tenth clerisy provides as resources to individuals such as an F1Tenth simulator, which can simulate various controls, planning, and vision algorithms before deploying them onto a physical car.

Motivated by the desire to integrate ROS2 development, AI-driven decision-making, and racetrack navigation, this project focuses on the simulation environment for the F1tenth car. The objective is to achieve stable, collision-free navigation along realistic racetracks that are scaled to match the minimized nature of F1Tenth, exemplified by tracks such as the 1/10th replica of the Circuit of the Americas (COTA) [2, 3], maximizing lap speed and minimizing contact with obstacles. Ultimately, this simulation-based study aims to guide future work on transferring learned control policies into the real world.

II. Related Work

Research into scaled autonomous racing platforms, including F1tenth, has demonstrated how the integration of perception, control, and planning algorithms can yield agile and robust navigation [4]. Wall Following approaches, often implemented with PID control [5], have been shown to effectively guide robots through corridors and tracks by maintaining parallel alignment to boundaries. Pure Pursuit algorithms [6] have long been a standard for path planning, using look-ahead points to achieve smooth trajectories. Additionally, reactive methods like Follow-The-Gap [7] have proven effective for real-time obstacle avoidance in cluttered environments.

While prior research has validated these techniques individually, less emphasis has been placed on comparing them head-to-head under similar conditions, or examining their combined use within the F1Tenth simulation environment. This work addresses this gap by implementing and evaluating multiple algorithms in a unified simulator setting that mimics the real F1tenth hardware platform.

III. Problem Formulation

I define an optimization goal that involves maximizing lap speed (L) while minimizing obstacle collisions (O). The cost function is given by:

$$T = L + (O \times 20)$$

Here, L represents the recorded lap time, and O represents the number of collisions detected. The aim is to minimize T, indicating faster lap completion with fewer obstacles hit.

We employ a realistic F1tenth simulation environment that includes racetracks modeled after real-world circuits. This environment publishes sensor data (e.g., LIDAR scans) and enables a standard ROS-based pipeline for perception and actuation. Thus, the solutions developed in simulation can be directly translated to the physical F1tenth hardware once properly tuned and integrated.

IV. Method

Below are three main navigation algorithms evaluated:

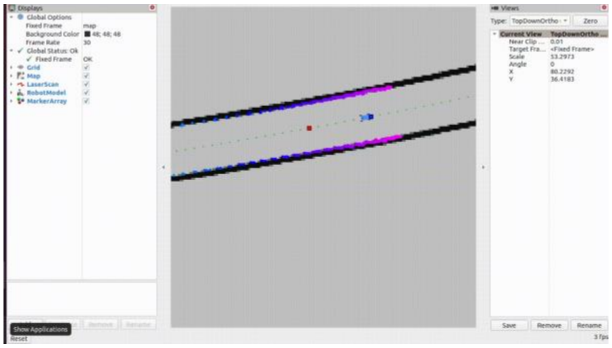
1. **Wall Following (PID Control):**

This approach uses a Proportional-Integral-Derivative controller to maintain a set distance from the wall. The car drives parallel to track boundaries, adjusting steering to correct deviations. This method ensures stability but may suffer from slow speeds in open areas. The error of the previous position compared to the wall is utilized to adapt the steering angle.

Wall Following Algorithm (PID Control) in F1Tenth Simulator

2. **Pure Pursuit Path Planning:**

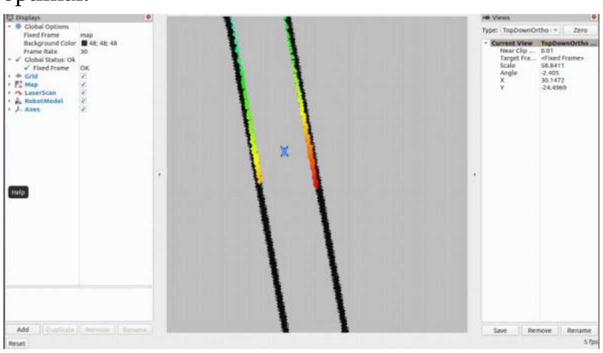
Pure Pursuit uses a geometric-based path tracker that determines the steering angle by projecting a look-ahead point on the desired path. Localization provides the current position, and the algorithm aims for smooth, collision-free trajectories. While Pure Pursuit is known for simplicity and robustness, tuning parameters like look-ahead distance is critical as it can determine if the vehicle will run into walls or navigate around them.



Pure Pursuit Path Planning Algorithm in F1Tenth Simulator

3. **Follow-The-Gap (Reactive Avoidance):**

For dynamic obstacle avoidance, Follow-The-Gap identifies the largest "gap" in the sensor data free of obstacles. The robot then steers toward this gap, ensuring a reactive, locally optimal maneuver. This approach quickly adapts to unexpected obstacles but may not be globally optimal.



Follow-The-Gap Algorithm (Reactive Avoidance) in F1Tenth Simulator

All algorithms were integrated into the same simulation environment and used the same set of published/subscribed ROS topics (/scan, /drive, /map, /ego_racecar/odom). I implemented and tested these approaches on a known racetrack layout, with consistent initial conditions.

V. Results

I evaluated each algorithm on a simulated racetrack with minimal obstacles. The goal was to minimize the cost function which was computed as: $T = L + (O \times 20)$. The results were as follows:

- **Wall Following:**
  The first algorithm tested achieved a performance score of 302, where the cost function was:
  $$302 = 302 + (0 \times 20).$$
  This demonstrates stable, albeit slower navigation, with no collisions. This slow navigation was due to the nature of the algorithm, requiring constant realignment and adjustment based on the previous pose and error.
- **Follow-The-Gap:**
  The second algorithm tested achieved a score of 275, again with no collisions, where the cost function was:
  $$275 = 275 + (0 \times 20).$$
  Although this is an improvement over Wall Following in terms of speed, it still faces limitations in optimizing the global trajectory, similar to greedy algorithms. It failed to quickly navigate through the course as it was oscillating between the walls.
- **Pure Pursuit:**
  Attained a score of 176, where the cost function was:
  $$176 = 176 + (0 \times 20).$$
  This represents the best overall performance. The Pure Pursuit path planning led to faster lap completion times and maintained collision-free driving. It followed an already determined center and followed this path.

These results highlight that while Wall Following and Follow-The-Gap ensure safety and feasibility, Pure Pursuit showed the greatest potential for speed and efficiency. However, we note that the Pure Pursuit algorithm did not achieve an "optimal" path; instead, it followed a centered path. Further parameter tuning and integration with obstacle avoidance strategies could yield even better performance.

VI. Conclusion And Discussion

My investigation into F1tenth autonomous racing compared three popular navigation algorithms within a unified simulation environment. These results indicate that Pure Pursuit offers the best combination of speed and safety, outperforming both Wall Following and Follow-The-Gap in our experiments. These findings suggest that geometric path planning strategies can provide a strong baseline for autonomous racing scenarios, particularly when obstacles are minimal. However, additional challenges emerge within Pure Pursuit when data is not directly obtained from the racetrack. This requires the Pure Pursuit algorithm to complete SLAM for the path creation. This requires an additional algorithm to be implemented for such a task: Particle Filter or Extended Kalman Filter.

Due to F1Tenth's physical parts being delayed, integrating and testing the simulation environment's solutions directly into a physical F1Tenth car was impossible. Future work will focus on improving the Pure Pursuit parameters, path, and incorporating more advanced perception methods (e.g., SLAM-based localization). In conclusion, my long-term goal is to deploy my simulation code to a physical F1Tenth robot which will bring me one step closer to robust and scalable autonomous racing.

References

[1] Jessner, W. (2024, February 28). *How much does a Formula 1 car cost and where can I buy one?* Red Bull. https://www.redbull.com/us-en/theredbulletin/what-does-a-formula-1-car-cost

[2] Wikimedia Foundation. (2024, November 23). *Circuit of the Americas*. Wikipedia. https://en.wikipedia.org/wiki/Circuit_of_the_Americas

[3] Betz, J., Zheng, H., Liniger, A., Rosolia, U., Karle, P., Behl, M., Krovi, V., & Mangharam, R. (2022). Autonomous vehicles on the edge: A survey on autonomous vehicle racing. *IEEE Open Journal of Intelligent Transportation Systems*, *3*, 458–488. https://doi.org/10.1109/ojits.2022.3181510

[4] Zheng, H., Betz, J., O'Kelly, M., Luong, K., Xue, H., Whitley, J., Zheng, R., Silberman, J., & Benson, J. (2023, September 5). *F1TENTH - Build Documentation*. Build. https://roboracer.ai/build.html

[5] Ang, K. H., Chong, G., & Li, Y. (2005). PID control system analysis, design, and Technology. *IEEE Transactions on Control Systems Technology*, *13*(4), 559–576. https://doi.org/10.1109/tcst.2005.847331

[6] Samuel, M., Hussein, M., & Binti, M. (2016). A review of some pure-pursuit based path tracking techniques for control of Autonomous Vehicle. *International Journal of Computer Applications*, *135*(1), 35–38. https://doi.org/10.5120/ijca2016908314

[7] Sezer, V., & Gokasan, M. (2012). A novel obstacle avoidance algorithm: "Follow the gap method." *Robotics and Autonomous Systems*, *60*(9), 1123–1134. https://doi.org/10.1016/j.robot.2012.05.021