Proposal: Predictive Path Planning for Multi-Agent Racing

TAREQ ALANSARI, UBC, Canada
RATIQ NARWAL, UBC, Canada
TONY KULENOVIC, UBC, Canada
ARAV VERMA, UBC, Canada

ACM Reference Format:
Tareq Alansari, Ratiq Narwal, Tony Kulenovic, and Arav Verma. 2024. Proposal: Predictive Path Planning for Multi-Agent Racing. *J. ACM* 1, 1, Article 1 (November 2024), 7 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Autonomous multi-agent racing is an intricate challenge in the F1TENTH community, featuring the intersection of a wide variety of fields. This project successfully developed a robust and efficient framework for predictive path planning and strategic decision-making, tailored for competitive racing environments.

The framework was implemented with ROS2 on a modified RC car equipped with a 2D LiDAR sensor, an NVIDIA Jetson computing unit. It leverages advanced racing methodologies such as occupancy grid-based planning, motion prediction, and heuristic algorithms like A*. Through a combination of reactive behavior and predictive capabilities, our system achieved safe and optimal performance in stochastic multi-agent scenarios.

The results demonstrate the feasibility of applying predictive path-planning techniques in high-speed, dynamic environments, pushing the boundaries of autonomous racing systems. This work contributes not only to the F1TENTH community but also to broader research in autonomous systems, offering insights into the challenges and solutions for operating in real-world multi-agent environments.

1.1 Problem Statement

The challenge lies in designing a predictive path-planning system for a multi-agent autonomous racing environment. Unlike static obstacles, moving agents must be identified and distinguished from walls or other static objects, requiring advanced perception capabilities. This may require us to depend on the camera and computer vision libraries such as OpenCV or PyTorch. However, due to our technology stack, the simulator has no support for vision based algorithms. As a result, we limited our algorithms to work with LiDAR, Odometry, and race map data.

It should be noted that although our algorithm works in theory and in the simulator, this might not be true for real-world testing. The algorithm is susceptible to faults in odometry data, which may be unreliable in real life.

Furthermore, the system must operate effectively at high speeds, necessitating a balance between reactive components for immediate decision-making and planning components for strategic maneuvers such as overtaking. The goal is to develop a system that can dynamically adapt to the

Authors' Contact Information: Tareq Alansari, UBC, Vancouver, Canada, tareq.ayz@gmail.com; Ratiq Narwal, UBC, Vancouver, Canada, tareq.ayz@gmail.com; Tony Kulenovic, UBC, Vancouver, Canada, tareq.ayz@gmail.com; Arav Verma, UBC, Vancouver, Canada, tareq.ayz@gmail.com.

ACM 1557-735X/2024/11-ART1
https://doi.org/XXXXXXX.XXXXXXX

behavior of other agents, ensuring safe as well as efficient navigation while maintaining competitive performance on the track.

1.2 Importance

Within the F1TENTH community, this project represents the pinnacle of autonomous racing challenges. This project is significant because it provides a clear goal to strive toward: building a competition-standard autonomous racing car and algorithm. It requires the integration of knowledge from algorithms, data structures, mathematics, and programming, while encouraging research into advanced methods for improving the car's performance.

Beyond the academic context, this project offers valuable real-world insights into how autonomous vehicles must operate in stochastic environments with multiple unpredictable agents. It highlights the challenges of balancing safety and performance, allowing us to appreciate the complexity of software systems behind safe autonomous driving vehicles.

1.3 Anticipated Challenges

One of the most significant challenges of this project is translating theoretical knowledge into a practical and feasible implementation. While the concepts of predictive path planning and multi-agent systems are well-documented, applying these ideas to the real-world will prove to be a challenge.

The project is particularly hard because there is no definitive end goal; solutions can always be further optimized. This may involve developing new strategies to reduce runtime overhead, minimize memory usage, or achieve better performance results. Here are some challenges that will need to be faced:

- **Simulation Setup:** Configuring the ROS2 simulator for multi-agent racing scenarios is a challenge in itself. This includes implementing dynamic agents with realistic racing behavior.
- **Predictive Path Planning:** Designing and implementing a predictive path-planning system that accounts for the behavior of other agents. This involves combining reactive components to create a dynamic path planning with race strategies.
- **Perception Systems:** Identifying moving agents as distinct from walls or static obstacles is non-trivial. LiDAR-based perception and may not be enough, in which case computer vision might be an asset.
- **High-Speed Operation:** Ensuring the system operates efficiently and accurately at high speeds. This involves making sure we implement optimized algorithms with a reasonable runtime.

2 Proposed Work

This section outlines the objectives and a feasible timeline for implementing a predictive path-planning system in a multi-agent racing environment.

2.1 Objectives

The primary objectives of this project are:

- **Develop a Multi-Agent Simulator:** Configure a ROS2 simulation environment in RViz to replicate multi-agent racing scenarios with dynamic agents.
- **Design Predictive Path-Planning Algorithms:** Create a framework capable of distinguishing moving agents from static obstacles and optimizing race trajectories for overtaking while avoiding collisions.

- **Integrate Reactive and Strategic Components:** Combine real-time reactive elements with race path planning advanced decision-making.
- **Demonstrate and Evaluate Performance:** Validate the system in both simulation and real-world, with a focus on safety and speed.

3 Evaluation Metrics

The performance of the predictive path-planning system will be evaluated based on both quantitative and qualitative metrics defined below.

3.1 Success Criteria

The success of the project will be determined by the following criteria:

- The ability of the system to distinguish moving agents from static obstacles in both simulation and real-world tests.
- Successful execution of overtaking maneuvers without collisions in the simulator and in the real-world.
- Competitive lap times that demonstrate efficient path planning and optimized race trajectories (with and without opponents).
- Robust performance under varying scenarios, including different speeds and opponent behaviors.

4 Path Planning Algorithms

In this section, we discuss the algorithms employed to implement a multi-agent racing system. The design leverages a hierarchical planning approach, where a global planner generates an initial, pre-defined trajectory based on map data, and local planners refine this trajectory in real-time to account for dynamic obstacles, static obstacles, and general deviatians from the global path.

4.1 Global Planner

The global planner provides a high-level trajectory by discretizing the track into a series of waypoints, which act as reference points for the car to follow. All of this occurs We utilized the Pure Pursuit algorithm as the core method for global path planning, due to its simplicity and effectiveness in generating smooth trajectories.

Here's an overview of how the algorithm works:

(1) **Map Preprocessing**: Map data was obtained using a SLAM pipeline, which generated a 2D occupancy grid of the environment. In this project, we used the *slam_toolbox* package to create accurate and detailed maps of the racing track. These maps served as the foundation for the subsequent steps in the planning process.
(2) **Waypoint Generation**: To create a raceline for the Pure Pursuit algorithm, we first extracted key features from the map, such as track boundaries and curves. A spline was then fitted to represent the optimal racing path. This spline was discretized into evenly spaced waypoints, providing a high-resolution reference for the algorithm. Each waypoint was annotated with positional and curvature information, which aids in generating smooth trajectories.
(3) **Pure Pursuit Algorithm**: Using the generated waypoints, the Pure Pursuit algorithm calculated a curvature-based trajectory by steering toward a dynamically selected look-ahead point along the path. This method ensured smooth and adaptive navigation while balancing simplicity and real-time efficiency, making it particularly well-suited for high-speed racing scenarios.

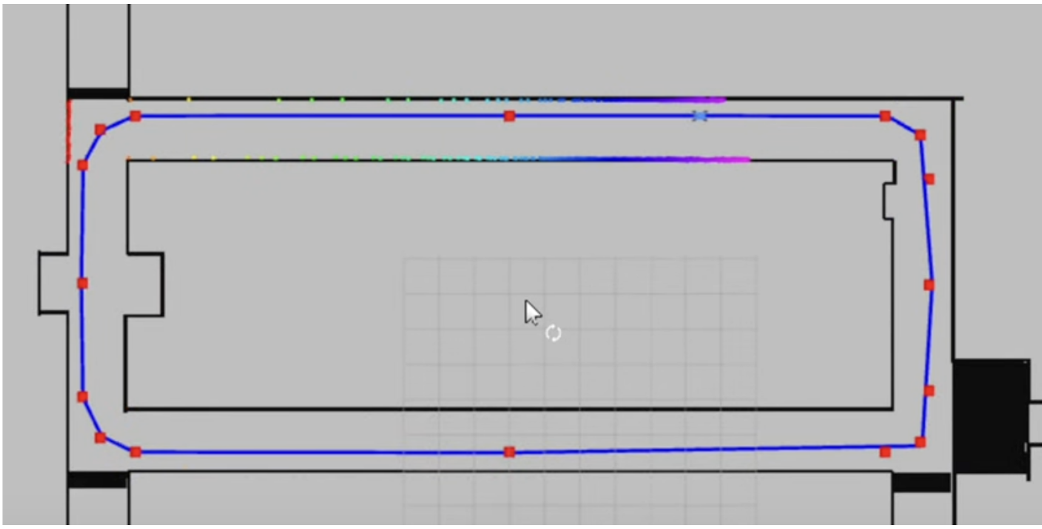

Fig. 1. Pure Pursuit Algorithm

This planner performs exceptionally well in both simulation and real-world scenarios, provided that the map data is accurate and corresponds directly to the real-world track. However, the algorithm's limitations become evident when unpredicted obstacles are introduced on the track. To address this, a local planner is required to dynamically adjust the trajectory and ensure safe navigation in real-time.

4.2 Local Planners

While the global planner provides an optimal baseline trajectory, it does not account for dynamic obstacles or changes in the environment. This is where the local planner comes into play. In this section, we cover a range of strategies and methodologies for local planners.

4.2.1 Occupancy Grid Methodologies. In this section, we cover local planners that leverage an occupancy grid to model the environment, integrating sensor data (e.g., LiDAR) to detect obstacles and other agents. This grid provides a probabilistic representation of the space, identifying free, occupied, and unknown regions.

In our occupancy grid representation, we used -1 to represent free space and a float value in the range $[0, 1]$ to represent the probabilistic occupancy measure for a specific grid coordinate. We used Brehansam's line algorithm to populate the grid from lidar data. Then, several algorithms can be used to produce a local path from the car to the lookahead waypoint.

Several algorithms were explored for local path planning on the occupancy grid:

(1) **A* Algorithm**
A* was chosen for its reliability and computational efficiency in grid-based pathfinding. It uses a heuristic to estimate the cost of reaching the goal, enabling the generation of least-cost collision-free paths.The cost function was designed to prioritize paths that adhered closely to the global trajectory while avoiding obstacles.

(2) **RRT and RRT***
These sampling-based planners excel in navigating complex environments. RRT generates feasible paths by exploring the space rapidly, while RRT* refines the path for optimality.

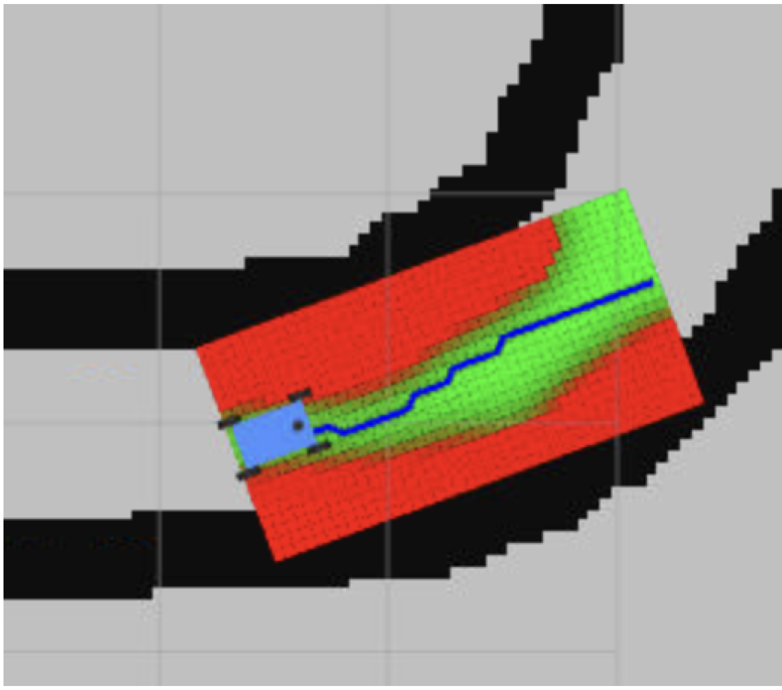

Fig. 2. Occupancy Grid Local Planning, *A* Algorithm*

While promising, their higher computational cost limited their applicability in real-time racing scenarios.

(3) **D* Algorithm** The D* (Dynamic A*) algorithm is an extension of the A* algorithm, specifically designed to handle dynamic environments where the map can change over time. Unlike A*, which computes a static path from the start to the goal, D* continuously updates the path as new information becomes available. This makes it particularly suitable for scenarios with moving obstacles or dynamic changes in the occupancy grid.

Limitations of Occupancy Grid Methods. Although occupancy grid methods are simple and effective, they are unable to differentiate between dynamic obstacles—i.e., other agents—and static obstacles, like walls. This limitation hinders their ability to implement advanced racing strategies.

For instance, in real-world multi-agent racing scenarios, RRT-based planners often struggle with overtaking opponents. The local path may **oscillate** around the opponent's position, potentially causing erratic behavior or, in severe cases, collisions with the opponent. This makes these methods suboptimal for real-world racing scenarios, where differentiation between static and dynamic elements is crucial for advanced strategy development.

Additionally, iterating over LiDAR and populating the occupancy grid is a costly operation. Ways to optimize this would be finding strategies to minimize read and write operations to the occupancy grid, like populating the grid on a need-to-know basis or saving occupancy grid information for next iterations.

Future Considerations. To overcome these limitations, integrating dynamic obstacle recognition techniques (e.g., incorporating velocity data or motion prediction models) could significantly enhance the effectiveness of local planners. Additionally, exploring hybrid approaches that combine occupancy grids with advanced methodologies like Model Predictive Control or Reinforcement Learning could provide more adaptive solutions.

5 Results

The success of our project was measured through a combination of quantitative and qualitative evaluations, focusing on the car's ability to navigate multi-agent racing scenarios effectively. In this section, we outline the key metrics, observations, and areas for further exploration.

5.1 Quantitative Measurements

To evaluate the performance of our system, we collected quantitative measurements based on the car's ability to avoid collisions with other agents in a dynamic racing environment. The key observations include:

- The car successfully avoided agents in the majority of test cases, demonstrating the effectiveness of the local planner in identifying and reacting to dynamic obstacles.
- Path planning algorithms, such as A* on the occupancy grid, consistently produced safe trajectories, ensuring that the car maintained a collision-free path even in challenging multi-agent scenarios.
- Visual inspection of the car's movements showed that the algorithm was functioning as intended, adapting to the presence of other agents and recalculating paths in real time.

These results confirm that the primary objective of the project—designing a system capable of multi-agent racing with dynamic obstacle avoidance—was successfully achieved.

5.2 Qualitative Observations

Although the primary focus was on quantitative metrics, some qualitative observations were made during the project:

- Visual feedback from simulation runs indicated smooth trajectory adjustments and reliable agent avoidance behavior.
- The car demonstrated consistent adherence to the global planner's path while incorporating local adjustments for agent avoidance.

5.3 Limitations and Future Work

While the results indicate success, there are areas where further development and measurements could enhance the system's performance:

- **Additional Quantitative Metrics:** Given more time, we would have measured lap times to evaluate the racing strategy's overall efficiency. Additionally, we could use these race times to compare different strategy implementations. This would have provided deeper insight into the effectiveness of different strategies in the context of multi-agent racing.
- **Optimization of Racing Strategies:** Future work could focus on fine-tuning the balance between speed and safety, developing advanced racing strategies for overtaking and time optimization.
- **Graph-Based Visual Results:** More extensive qualitative evaluations, such as assessing the car's behavior in edge cases or more complex scenarios, would provide a holistic understanding of system performance.

6 Conclusion

The primary aim of this project was to design a racing strategy capable of navigating multi-agent racing settings with a high degree of success while maintaining a competitive lap time. Based on our results, we successfully achieved this goal by integrating the Pure Pursuit algorithm with occupancy grid-based local planners, particularly through our implementation of the A* algorithm. Our system demonstrated the ability to safely detect and avoid both dynamic and static obstacles, all while maintaining competitive racing speeds.

Looking ahead, future work will focus on building upon these results to achieve higher performance and incorporate more advanced algorithms. Specifically, one area of improvement involves identifying and distinguishing opponents to implement sophisticated racing strategies, such as overtaking maneuvers, which were beyond the scope of the current implementation. Despite these limitations, we are confident that the current system provides a robust foundation for future advancements, enabling the development of increasingly competitive and intelligent multi-agent racing strategies.