

Comparative Study of Delaunay Path Planner and RRT/RRT* for Path Planning in Autonomous Racing

Rodrigo Carrión Caro

March 17, 2025

Contents

Glossary	1
Abstract	2
Acknowledgements	3
1 Introduction	4
1.1 Background	4
1.2 Aim	4
1.3 Objectives	5
1.4 Product Overview	5
1.4.1 Scope	5
1.4.2 Audience	6
2 Background Review	7
2.1 Related Literature	7
3 Methodology	11
3.1 Research and Software Development Process	11
3.1.1 Application of Agile Methodology	11
3.1.2 Development Phases	11
3.2 Technology	11
3.2.1 Implementation Tools and Resources	11
3.2.2 Simulation and Testing Tools	11
3.3 Version Management	11
3.3.1 Source Code Control with GitHub/GitLab	11
3.3.2 Data and Version Storage in Google Drive	11
3.3.3 Source Code Repository Link	12
4 Results	13
4.1 Results and Testing	13
4.1.1 Simulation Setup	13
4.1.2 Test Scenarios	13
4.2 Performance Metrics	13
4.2.1 Computation Time Evaluation	13
4.2.2 Path Quality Analysis	13
4.2.3 Adaptability to Dynamic Obstacles	13
4.3 Experimental Results	13
4.3.1 Comparison of Metrics Between Optimized Delaunay and RRT* .	13

4.3.2	Visualization of Results Through Graphs	13
5	Professionalism	14
5.1	Project Management	14
5.1.1	Development Activities and Schedule	14
5.1.2	Data Management	14
5.1.3	Project Deliverables	14
5.2	Risk Analysis	14
5.2.1	Identified Risks and Mitigation Strategies	14
5.2.2	Updated Project Plan Based on Risk Evaluation	14
5.3	Legal, Ethical, and Environmental Considerations	14
5.3.1	Compliance with Professional Codes of Conduct	14
5.3.2	Ethical and Environmental Impact of the Project	14
6	Conclusion	15
6.1	Summary of Findings	15
6.1.1	Key Insights from the Algorithm Comparison	15
6.2	Future Work	15
6.2.1	Improvements in RRT* Implementation	15
6.2.2	Real-World Applications	15
7	Bibliography	16
8	Appendices	18
8.1	Supplementary Data	18
8.1.1	Source Code Repository (GitHub/GitLab)	18
8.1.2	ROS2 + Unity Configuration Details	18
8.1.3	Raw Simulation Results	18

Glossary

- **RRT** - Rapidly-Exploring Random Tree
- **ROS2** - Robot Operating System 2
- **Unity** - Motor de simulación
- **Path Planning** - Planificación de rutas

Abstract

0.1 Contexto

Escribe aquí el contexto del problema.

0.2 Objetivo

Escribe aquí la descripción clara del propósito del proyecto.

0.3 Metodología

Describe el enfoque de desarrollo y herramientas utilizadas.

0.4 Resultados principales

Explica los principales hallazgos y comparación entre métodos.

0.5 Conclusión

Menciona el impacto del estudio y posibles mejoras.

Acknowledgements

Escribe aquí los agradecimientos al supervisor, equipo OBRA, etc.

Chapter 1

Introduction

1.1 Background

In autonomous vehicles, path planning is essential, as it allows for efficient and safe route planning in real time. Specifically, the RRT algorithm “RRT Rapidly exploring Random Tree” can explore large spaces efficiently. This makes it ideal for dynamic environments, such as an autonomous car race. These characteristics make it widely used in the field of robotics. The RRT (Rapidly exploring Random Tree) algorithm is a search method used to efficiently find paths in large spaces [1]. It is particularly useful in dynamic environments where viable paths need to be found quickly in real-time. Currently, the Oxford Brookes Racing Autonomous (OBRA) team uses a neural network path planner. A Delaunay path planner is being developed but has not yet been implemented. These approaches present some limitations in real-time situations. The development and implementation of an RRT [2] will allow for greater flexibility and adaptability. Ideally, this will improve the team’s path planning capabilities. In addition to developing the RRT, this project will compare its performance with the current Delaunay path planner. By evaluating their efficiency and adaptability in dynamic scenarios, the goal is to determine which algorithm provides better results for autonomous racing.

1.2 Aim

The aim of this project is to develop and optimize two advanced path planning algorithms for autonomous racing: an improved Delaunay-based planner and an optimized RRT* algorithm. These planners will be designed to enhance adaptability, computational efficiency, and trajectory optimization in high-speed, dynamic environments.

This project also aims to conduct a detailed comparative analysis of both algorithms, evaluating their performance across key metrics such as computation time, path efficiency, adaptability to dynamic obstacles, and robustness under racing conditions.

By implementing and testing these algorithms within a ROS2-based simulation environment using Unity, this research seeks to identify the most effective path planning solution for the OBRA team’s autonomous racing vehicle. The insights gained from this study will contribute to both the OBRA competition strategy and the broader field of autonomous vehicle navigation.

1.3 Objectives

- To conduct a comprehensive study of path planning techniques used in autonomous vehicle navigation, analyzing their advantages, limitations, and applications in high-speed racing scenarios. [3]
- To develop and implement an improved Delaunay-based path planner that enhances adaptability, computational efficiency, and trajectory smoothness in dynamic environments.
- To develop and implement an RRT algorithm from scratch for real-time path planning in autonomous vehicles, ensuring compatibility with the OBRA car's ROS2 framework.
- To optimize the RRT algorithm by integrating RRT* [4], improving its ability to generate efficient and dynamically adaptable routes.
- To validate both the optimized Delaunay and RRT* algorithms in a controlled simulation environment using Unity, testing their performance under varying racing conditions.
- To conduct a detailed comparative analysis of the optimized Delaunay and RRT* algorithms, evaluating key performance metrics such as computation time, path efficiency, adaptability to dynamic obstacles, and robustness in high-speed scenarios.
- To identify and address potential limitations of each algorithm, proposing refinements or hybrid approaches that could further enhance their performance.
- To integrate the most effective path planner into the OBRA team's autonomous racing pipeline, ensuring real-world applicability and alignment with competition requirements.

1.4 Product Overview

1.4.1 Scope

The objective of this project is to develop and optimize two new path planners for autonomous racing:

- **Optimized Delaunay Path Planner** – A modified version of the traditional Delaunay-based planner, improving its efficiency and adaptability for dynamic racing environments.
- **RRT*** – An enhanced Rapidly-exploring Random Tree algorithm that generates smoother and more efficient paths by reducing randomness and refining route selection.

Both planners will be developed in Python, ensuring seamless integration with the ROS2 framework used in the OBRA autonomous car. The testing and validation process will be conducted in simulation environments using Unity, allowing for extensive evaluation before potential real-world implementation. Once developed, these two new planners

will be compared to determine which provides better performance in terms of adaptability, computational efficiency, and trajectory smoothness under high-speed, dynamic conditions.

1.4.2 Audience

The primary audience for this study is the Oxford Brookes Racing Autonomous (OBRA) team, as the improved path planning algorithms will directly contribute to their autonomous racing performance. Additionally, this research is relevant to the academic community, particularly in robotics, AI, and autonomous vehicle navigation, by providing insights into optimization strategies for real-time path planning. From an industry perspective, this study holds significance for autonomous systems professionals, particularly those developing path planning solutions for high-speed and dynamic environments, such as self-driving vehicles, robotics, and UAV navigation.

Chapter 2

Background Review

2.1 Related Literature

Reference	Zhao, H., Wu, Z., Li, Y., & Wang, J. (2021) ‘Improved Bidirectional RRT* Path Planning Method for Smart Vehicle’, Mathematical Problems in Engineering, pp. 1-14. doi: 10.1155/2021/6669728.
Title	Improved Bidirectional RRT* Path Planning Method for Smart Vehicle
Summary	The study proposes an improvement to the bidirectional RRT* algorithm to optimize route planning in intelligent vehicles, aiming for shorter and more efficient routes.
Evaluation	The approach is useful for static environments, but it does not sufficiently address the challenges in dynamic environments, which may limit its applicability in competitive vehicles.
Reflection	The proposed improvements could be applied to optimize the RRT* algorithm in my project, particularly in reducing computation time and optimizing routes.
Main Themes	RRT* optimization, route planning, autonomous vehicles.

Table 2.1: Summary of Zhao et al. (2021)

Reference	Gasparetto, A., Boscariol, P., Lanzutti, A., & Vidoni, R. (2015) ‘Path Planning and Trajectory Planning Algorithms: A General Overview’, Journal of Intelligent & Robotic Systems, pp. 1-33. doi: 10.1007/978-3-319-14705-5_1.
Title	Path Planning and Trajectory Planning Algorithms: A General Overview
Summary	The article provides an overview of the main trajectory and route planning algorithms in robotics. It analyses methods such as Roadmap, Cell Decomposition, and RRT, as well as their applications in industrial and autonomous environments.
Evaluation	The study offers a comprehensive overview of the algorithms but focuses more on static industrial systems, limiting its applicability to dynamic environments. However, the review of RRT is useful for improving my project.
Reflection	This article will be key to contextualizing my work, as it provides a solid foundation on traditional methods and suggests possible areas for improvement, such as applying them in more dynamic environments.
Main Themes	Route planning, RRT, optimization algorithms, autonomous robotics.

Table 2.2: Summary of Gasparetto et al. (2015)

Reference	Wang, H., Li, G., Hou, J., Chen, L., & Hu, N. (2022) ‘A Path Planning Method for Underground Intelligent Vehicles Based on an Improved RRT* Algorithm,’ Electronics, vol. 11, no. 3, p. 294. doi: 10.3390/electronics11030294.
Title	A Path Planning Method for Underground Intelligent Vehicles Based on an Improved RRT* Algorithm
Summary	The study proposes an improved RRT* method for route planning in underground intelligent vehicles, adjusting the dynamic step size and turn angle constraints.
Evaluation	The approach is innovative for underground environments and offers improvements in efficiency and safety, but it is limited to controlled spaces and does not address navigation in fully dynamic environments.
Reflection	This study is relevant to my project, as the proposed RRT* improvements could be applied to optimize the algorithm in more complex scenarios, such as autonomous racing.
Main Themes	RRT*, underground autonomous vehicles, route planning, optimization.

Table 2.3: Summary of Wang et al. (2022)

Reference	Sánchez-Ibáñez, J.R., Pérez-del-Pulgar, C.J., & García-Cerezo, A. (2021) ‘Path Planning for Autonomous Mobile Robots: A Review’, Sensors 2021, 21, 7898. doi: 10.3390/s21237898.
Title	Path Planning for Autonomous Mobile Robots: A Review.
Summary	The article reviews route planning algorithms for mobile robots, focusing on their classification and applicability in autonomous environments.
Evaluation	It provides a very useful overview of the main approaches, but it focuses on controlled scenarios and may be limited for dynamic environments such as competitions.
Reflection	This article provides a good foundation for comparing different approaches, which will help me justify the choice of RRT in my project.
Main Themes	Route planning, mobile robots, path search algorithms.

Table 2.4: Summary of Sánchez-Ibáñez et al. (2021)

Reference	C. Messer, A. T. Mathew, N. Mladenovic and F. Renda, "CTR DaPP: A Python Application for Design and Path Planning of Variable-strain Concentric Tube Robots," 2022 IEEE 5th International Conference on Soft Robotics (RoboSoft), Edinburgh, United Kingdom, 2022, pp. 14-20, doi: 10.1109/RoboSoft54090.2022.9762088.
Title	CTR DaPP: A Python Application for Design and Path Planning of Variable-strain Concentric Tube Robots.
Summary	The study presents a modular platform in Python that uses the RRT* algorithm for route planning and design optimization in concentric tube robots. It focuses on trajectory planning in environments with torsion and curvature constraints.
Evaluation	The implementation in Python is relevant to my project, as it allows for the flexible use of planning and optimization algorithms.
Reflection	This article supports the use of Python in my project, demonstrating that it is an effective option for developing and testing algorithms like RRT*.
Main Themes	RRT*, route planning, concentric tube robots, design optimization, use of Python.

Table 2.5: Summary of Messer et al. (2022)

Reference	Kolski, S., Ferguson, D., Stachniss, C., & Siegwart, R. (2006) 'Autonomous Driving in Dynamic Environments', Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1-10. doi: 10.3929/ethz-a-010079481.
Title	Autonomous Driving in Dynamic Environments.
Summary	The study presents a hybrid autonomous navigation system that operates in both structured and unstructured environments, handling dynamic obstacles like pedestrians and other vehicles.
Evaluation	Unlike many approaches focused on static environments, this system is highly relevant for dynamic settings, such as autonomous car competitions, where real-time adjustments to moving obstacles are critical.
Reflection	This study is essential for my project as it highlights the importance of dynamic environments and provides useful insights for improving my route planning system.
Main Themes	Autonomous navigation, dynamic environments, route planning, moving obstacles, autonomous vehicles.

Table 2.6: Summary of Kolski et al. (2006)

Chapter 3

Methodology

3.1 Research and Software Development Process

3.1.1 Application of Agile Methodology

Explanation of how Agile is applied in the development process.

3.1.2 Development Phases

- Phase 1: Implementation of the optimized Delaunay path planner.
- Phase 2: Development of the RRT* algorithm from scratch.
- Phase 3: Testing and comparison of both algorithms.

3.2 Technology

3.2.1 Implementation Tools and Resources

- Programming Languages: Python.
- Frameworks: ROS2 for robotic integration.
- Simulation Environment: Unity for path planning validation.

3.2.2 Simulation and Testing Tools

Details on how Unity and ROS2 are used for validation.

3.3 Version Management

3.3.1 Source Code Control with GitHub/GitLab

Version control and repository structure.

3.3.2 Data and Version Storage in Google Drive

How project logs, datasets, and backups are stored.

3.3.3 Source Code Repository Link

URL or reference to the project repository.

Chapter 4

Results

4.1 Results and Testing

4.1.1 Simulation Setup

Explanation of the Unity + ROS2 environment configuration.

4.1.2 Test Scenarios

List of simulated scenarios used for validation.

4.2 Performance Metrics

4.2.1 Computation Time Evaluation

Data on execution times.

4.2.2 Path Quality Analysis

Metrics evaluating the efficiency and smoothness of generated paths.

4.2.3 Adaptability to Dynamic Obstacles

Explanation of testing under dynamic conditions.

4.3 Experimental Results

4.3.1 Comparison of Metrics Between Optimized Delaunay and RRT*

Presentation of comparative data.

4.3.2 Visualization of Results Through Graphs

Graphs and their interpretation.

Chapter 5

Professionalism

5.1 Project Management

5.1.1 Development Activities and Schedule

Project logs, reports, and Gantt charts.

5.1.2 Data Management

Storage and organization of research documents.

5.1.3 Project Deliverables

Summary of key milestones.

5.2 Risk Analysis

5.2.1 Identified Risks and Mitigation Strategies

Discussion of risks encountered and strategies used.

5.2.2 Updated Project Plan Based on Risk Evaluation

Adjustments made due to identified risks.

5.3 Legal, Ethical, and Environmental Considerations

5.3.1 Compliance with Professional Codes of Conduct

References to BCS, ACM, and industry standards.

5.3.2 Ethical and Environmental Impact of the Project

Analysis of social and environmental implications.

Chapter 6

Conclusion

6.1 Summary of Findings

6.1.1 Key Insights from the Algorithm Comparison

Main takeaways from testing and evaluation.

6.2 Future Work

6.2.1 Improvements in RRT* Implementation

Potential refinements to enhance algorithm performance.

6.2.2 Real-World Applications

Application of findings to actual autonomous vehicle scenarios.

Chapter 7

Bibliography

Bibliography

- [1] LaValle, S. (2006) 'Rapidly exploring Random Trees: Overview', Available at: <https://lavalle.pl/rrt> (Accessed: 10 October 2024).
- [2] Bécsi, T. (2024) 'RRT-guided experience generation for reinforcement learning in autonomous lane keeping', Scientific Reports, 14, Article number: 24059. Available at: <https://www.nature.com/articles/s41598-024-73881-z> (Accessed: 16 October 2024).
- [3] Muhsen, D.K., Raheem, F.A., and Sadiq, A.T. (2024) 'A Systematic Review of Rapidly Exploring Random Tree RRT Algorithm for Single and Multiple Robots', Cybernetics and Information Technologies, 24(3), pp. 78-101. Available at: <https://doi.org/10.2478/cait-2024-0026> (Accessed: 19 September 2024).
- [4] Fan, H., Huang, J., Huang, X., Zhu, H., and Su, H. (2024) 'BI-RRT*: An improved path planning algorithm for secure and trustworthy mobile robots systems', Heliyon, 24(e26403). Available at: <https://doi.org/10.1016/j.heliyon.2024.e26403> (Accessed: 10 October 2024).

Chapter 8

Appendices

8.1 Supplementary Data

8.1.1 Source Code Repository (GitHub/GitLab)

Reference link to the source code.

8.1.2 ROS2 + Unity Configuration Details

Technical setup instructions.

8.1.3 Raw Simulation Results

Unprocessed data from testing and evaluation.