

YZV406E Robotics - Assignment #1

TurtleBot Explorer Implementation Report

Muhammed Rugen Birben
150220755

birben20@itu.edu.tr
Faculty of Computer and Informatics Engineering
Istanbul Technical University

December 8, 2024

1 World Description and Robot's Role

I used Gazebo to build a random virtual environment for this assignment. The world is made up of four connected areas with well placed obstacles to make exploration difficult. The major goal of the TurtleBot is to map and explore this environment on its own while adhering to safe navigation practices.

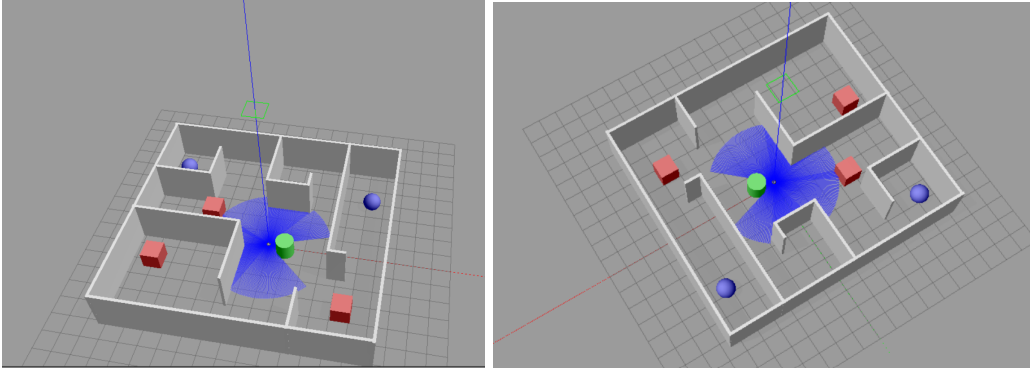


Figure 1: World Images Side by Side the object that is emitting the laser is the TurtleBot

2 Exploration Strategy

The exploration strategy implements a hybrid approach combining frontier-based exploration with artificial potential fields, enhanced by a memory system for efficient area coverage.

2.1 Frontier Detection and Evaluation

The robot identifies frontiers by analyzing laser scan data, specifically looking for transitions between measurable distances and unmeasurable ranges (infinity/NaN values). These transitions indicate the boundary between known and unknown space. For each scan point i , the system calculates frontier points (x, y) using:

$$x = x_{robot} + d \cos(\theta + \theta_{yaw}) \quad (1)$$

$$y = y_{robot} + d \sin(\theta + \theta_{yaw}) \quad (2)$$

where d is the valid distance measurement plus a 0.5m offset, and θ is the scan angle.

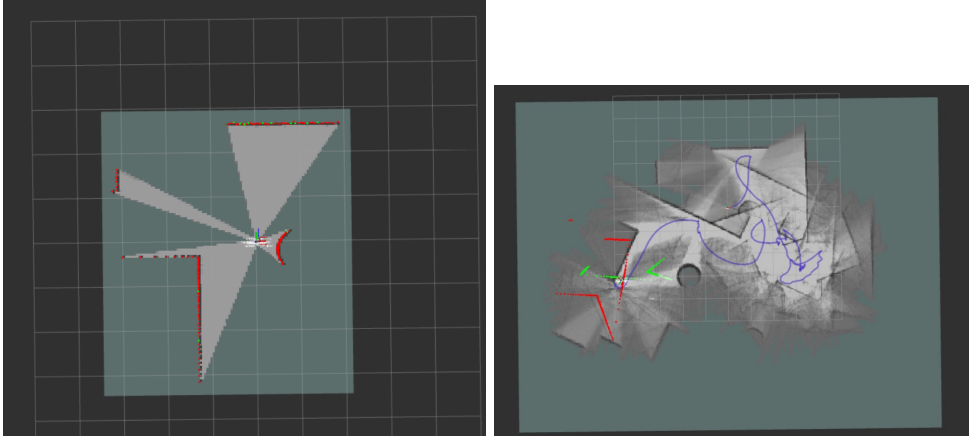


Figure 2: Sensor Images Side by Side

2.2 Frontier Scoring System

Each frontier point is evaluated using a comprehensive scoring function:

$$score = -((d - 2.0)^2) - 2.0|\Delta\theta| + p_{age} \quad (3)$$

where:

- $(d - 2.0)^2$ penalizes points too close or too far from the optimal 2-meter distance
- $|\Delta\theta|$ is the absolute angle difference from current heading
- p_{age} is the age penalty: $-0.5 \cdot \frac{age}{timeout}$ for previously visited frontiers

2.3 Potential Field Navigation

The navigation system employs a dynamic potential field with carefully tuned parameters:

- Wall repulsion strength: -3000.0 units
- Unexplored area attraction: 120.0 units
- Visited area penalty: -0.01 units
- Decay factor: 0.95 for smooth force transitions

The potential at any point (i, j) from a source at (x, y) is calculated as:

$$P(i, j) = strength \cdot (0.95^{distance}) \quad (4)$$

where $distance = \sqrt{(x - i)^2 + (y - j)^2}$

2.4 Movement Control Parameters

The robot's movement is governed by several key parameters optimized for efficient exploration:

- Normal speed: 1.5 m/s for efficient coverage
- Turning speed: 0.9 rad/s for responsive rotation
- Slow speed: 0.3 m/s for precise movements

- Rotation threshold: 0.5 rad for movement transitions
- Safety distance: 0.5m minimum front clearance

The angular velocity (ω) is calculated based on the angle difference ($\Delta\theta$):

$$\omega = \begin{cases} \pm 0.9 & \text{if } |\Delta\theta| > 0.5 \\ 0.7\Delta\theta & \text{otherwise} \end{cases} \quad (5)$$

2.5 Memory and Timeout System

The exploration system maintains a memory of:

- Last 10 frontier points
- 30-second timeout for frontier validity
- Visited position tracking for coverage optimization

This memory system prevents the robot from repeatedly exploring the same areas and ensures efficient coverage of the environment. The frontier memory is continuously updated and cleaned based on the timeout condition:

$$valid_frontier = (current_time - frontier_time) < 30.0s \quad (6)$$

This comprehensive system allows the robot to efficiently explore unknown environments while maintaining safe navigation and preventing repetitive exploration patterns.

3 Putting it all together: Navigation System

The navigation system combines the exploration strategy and safety mechanisms into a practical movement system. At each update (approximately 10 times per second), the robot follows this decision-making process:

3.1 1. Finding Where to Go

The robot first analyzes its laser scan data to find frontier points - the edges between known and unknown space. It looks for discontinuities in the scan data, where measurements transition from valid distances to infinity. These points, offset by 0.5m for safety, become potential exploration targets.

3.2 2. Choosing the Best Direction

From the identified frontier points, the robot selects its next direction by:

- Preferring frontiers about 2 meters away (optimal exploration distance)
- Favoring frontiers that require less turning
- Avoiding recently visited frontiers (using the 30-second memory system)

3.3 3. Moving Safely

The robot then adjusts its movement based on three simple rules:

- If the path ahead is clear ($>0.5\text{m}$), move at full speed (1.5 m/s)
- If a large turn is needed ($>0.5\text{ rad}$), slow down (0.3 m/s)
- If obstacles are too close ($<0.5\text{m}$), stop and rotate

This straightforward approach allows the robot to efficiently explore while maintaining safety. When stuck or uncertain, it naturally rotates to find new frontiers, ensuring continuous exploration of the environment.

4 Exploration Results

The result can be seen in the explored map in the following figure.



Figure 3: Final Exploration Map

as can be seen, the robot explored the environment and mapped it to the best of its ability.

The used exploration algorithm is a frontier-based exploration algorithm. Though it is not a perfect algorithm, it is a good start for the robot to explore the environment. The algorithm is not perfect because it may make the robot get stuck in local minima. This can be solved by using a more advanced exploration algorithm. Maybe implementing a more complex memory instead of potential field or using a more advanced frontier detection algorithm.

5 Challenges and Solutions

During the implementation of this project, I faced several interesting challenges:

- **Local Minima:** My initial implementation often got the robot stuck in corners. I solved this by adding the memory system and age penalties to encourage exploration of new areas.
- **Parameter Tuning:** Finding the right balance for the potential field parameters was tricky. Too strong repulsion made the robot overly cautious, while weak repulsion led to collisions. After extensive testing, I found the current values (-3000 for walls, 120 for attraction) provided the best balance.
- **Computational Efficiency:** Initially, my frontier detection algorithm was scanning every possible point, making the system sluggish. I optimized it by only processing significant changes in laser readings, which greatly improved performance.
- **Navigation Smoothness:** The robot's movement was initially jerky due to rapid changes in decision-making. I addressed this by implementing gradual speed transitions and the decay factor in the potential field calculations.

Although these difficulties occasionally annoyed me, overcoming them improved my knowledge of the theoretical and practical aspects of robotic exploration.

6 Conclusion

In this project, I successfully created an environment with obstacles and implemented an autonomous exploration system for the TurtleBot that combines frontier-based exploration with potential fields. While my implementation isn't perfect, it effectively demonstrates the core concepts of autonomous robot exploration. The robot successfully mapped the test environment while avoiding obstacles and maintaining safe navigation parameters. It had two minutes, so it could not explore the whole environment.

Through this implementation, I gained valuable hands-on experience with robotic navigation concepts, sensor data processing, and real-time decision-making systems. The challenges I encountered, particularly with local minima and parameter tuning, taught me the importance of balancing theoretical approaches with practical considerations.

Looking ahead, there's room for improvement in areas such as implementing a more sophisticated memory system or developing a more robust frontier detection algorithm. However, this project has provided me with a solid foundation in autonomous robotics and practical experience that will be valuable for future work in this field.

Note: I'm writing this note to our assistants and to our professor. First I would like to thank you for your effort in teaching us the robotics course, it is a fascinating field and I'm glad to be learning it. I noticed that while doing this assignment, it was both really enjoyable and extremely challenging at the same time. The reason is that I had to use the knowledge I gained from the lectures and apply it to the implementation. The thing is, this is the first time I'm doing something like this and I had to learn a lot of new things. So the feedback is: We should have much more time to do applied labs to familiarize ourselves with the material. It is both REALLY enjoyable to actually implement the things we learn and a WONDERFUL experience to learn by doing! I think robotic is really a fitting field to learn by doing, and I think I speak for everyone when I say that I learn the most when I'm doing something and failing at it. I'm not saying that I failed at it :) But I sure learned a lot from it. It would be blessing if you could consider this feedback of mine to turn the course into a more applied one. Have a great day!