Battery Energy Optimization for Single-Agent Frontier Based Exploration and Mapping

EECE5550 Mobile Robotics

1st Dumpala, Kodanda Rama Naidu 2nd Blanchard, Nicholas 3rd Sureshkumar, Hariharan

***Abstract*—Autonomous exploration of unknown environments is a profound concentration in the field of robotics and frontier-based methods are well-established in this area. This report focuses on expanding the frontier-based approach of exploration by incorporating a battery constraint into a Rapidly Exploring Random Trees (RRT) path planning algorithm. The addition of this constraint ensures that the robot not only explores the environment efficiently but also considers energy limitations. These energy limitations create an additional sense of reality as robots in the real world must consider their energy levels. This energy constraint will force the robot to make decisions on whether to prioritize exploration or prioritize the route to recharge stations to continue exploration without human intervention.**

I. Introduction

Autonomous robotic exploration is a crucial research area in robotics. The goal of exploration is to efficiently create the most accurate map of an environment without human intervention. Frontier-based exploration strategies have been a fundamental and widely used approach since the concept was first conceived by Brian Yamauchi in 1997 [1]. A frontier is defined as the boundary between known and unexplored space. The robot acknowledges these frontier points and plans a path towards the most desirable frontier point. The planning of this path is performed by the Rapidly Exploring Random Tree (RRT) algorithm [2].

The basis of this algorithm includes building a tree of nodes by sampling random points in space and connecting the closest pre-existing node from the tree to this random point in space. Once a path or branch of the tree extends to a node in an unknown space, this point is identified as a frontier point. The robot evaluates each of these frontier points and follows the optimal path created by the RRT algorithm to the optimal frontier point. Although RRT-based frontier exploration has proven to be effective, traditional implementations do not account for the energy limitations of robots.

In real-world applications, field robots must operate within energy constraints while ensuring mission success. If the robot has enough energy to complete its tasks, this limitation does not need to be taken into account. However, when exploring large environments, recharging is crucial. Ignoring these constraints can result in incomplete mapping of the environment and subsequently mission failure. Additionally, the robot will not have enough battery to return to a set location if needed, which may make it difficult to retrieve if left in a dangerous environment.

This paper addresses this constraint by introducing a battery threshold, which is incorporated directly into the RRT framework. In the scenario presented in the paper, a robot is tasked with exploring and mapping an unknown environment while monitoring its battery level. When the robot's battery dips below this threshold it shifts its objective from exploration to navigating to the closest charging station. Including a battery constraint into RRT-based frontier exploration enhances the utility of the algorithm in real-world scenarios while balancing efficient exploration.

II. Problem Statement

The objective is to design a robotic exploration strategy that maximizes coverage of an unknown environment while adhering to strict battery constraints. The robot must dynamically balance exploration and energy management, ensuring it navigates to fixed charging stations before battery depletion. This involves addressing challenges such as static charging station locations, requiring efficient path planning; real-time energy computation to continuously evaluate energy consumption and remaining battery life; and dynamic decision-making to adapt its objectives, switching from exploration to recharging as battery levels fall below a predefined threshold. The solution must optimize exploration coverage while ensuring safe and reliable operation in diverse and unknown environments.

III. Proposed Solution

A. Base RRT Algorithm and its role in exploration

The Rapidly Exploring Random Tree (RRT) algorithm is a popular motion planning tool for effectively searching high-dimensional areas. It gradually constructs a tree by picking random points in space and connecting them to the nearest existing node in the tree. This approach provides quick coverage of the space while avoiding obstructions, making it ideal for navigating uncharted environments. In the context of frontier-based exploration, RRT assists the robot in systematically moving toward undiscovered locations by recognizing and linking paths to frontier points, which mark the boundary between known and unknown areas.

B. Battery Constraint Implementation

Our updated framework adds a battery-aware method to the basic RRT algorithm. A threshold mechanism regularly checks the robot's battery level. When the battery level falls below a certain threshold (for example, 30% of total capacity), the robot suspends exploration duties and navigates to the nearest designated charging station. Once recharged, the robot starts its investigation at the last recorded state. This prevents power depletion and allows the robot to run longer missions without interruption.

C. Integration with Frontier-Based Exploration

The battery-aware system complements frontier-based exploration by balancing energy management and exploration demands. When the battery is full, the robot selects frontier spots based on proximity and information acquired, indicating the possibility to explore new places. If the battery level falls below a predetermined threshold, the system bypasses the frontier prioritization and leads the robot to the nearest charging station. After recharging, the robot resumes exploration from its last explored place, causing minimum disturbance to the operation while maintaining adequate map coverage.

D. Battery Constraint Implementation

Our updated framework adds a battery-aware method to the basic RRT algorithm. A threshold mechanism regularly checks the robot's battery level. When the battery level falls below a certain threshold (for example, 30% of total capacity), the robot suspends exploration duties and navigates to the nearest designated charging station. Once recharged, the robot starts its investigation at the last recorded state. This prevents power depletion and allows the robot to run longer missions without interruption.

E. Proposed Framework

The block diagram depicts the high-level architecture of the battery-constrained frontier exploration system. It focuses on the key components and their relationships.

1) Frontier Detection: Identifies unexplored regions (frontiers) in the environment based on the robot's mapping progress.

2) RRT Planner: Identifies unexplored regions (frontiers) in the environment based on the robot's mapping progress.

3) Battery Monitoring: Identifies unexplored regions (frontiers) in the environment based on the robot's mapping progress.

4) Charging Station Locator: Identifies unexplored regions (frontiers) in the environment based on the robot's mapping progress.

5) Charge and Restart Mechanism: Identifies unexplored regions (frontiers) in the environment based on the robot's mapping progress.

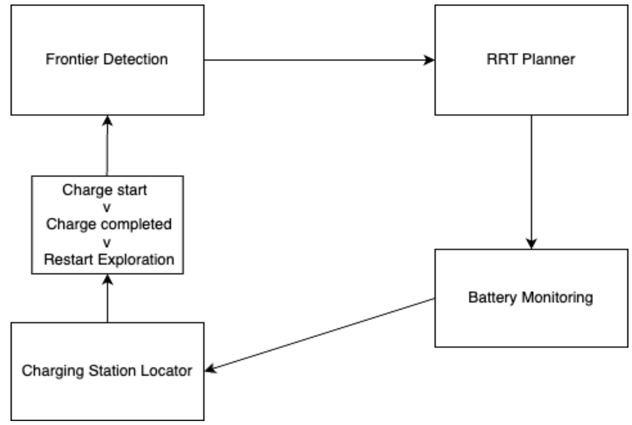

Fig. 1. Proposed framework.

F. Energy Cost Model

1) Battery Drainage Mechanics: The battery level decreases over time, with a fixed rate proportional to the distance traveled and exploration duration.

2) Recharging Behavior: Recharging is modeled as a fixed-duration event (e.g., 5 seconds) during which the battery level is restored to full capacity.

3) Threshold Mechanism: When the battery drops below a critical threshold, the robot calculates the shortest path to the nearest charging station and transitions from exploration to recharging mode.

```
while exploration_active:
    if battery_level < threshold:
        halt_exploration()
        nearest_station =
find_nearest_charging_station(robot_position)
        navigate_to(nearest_station)
        recharge()
        resume_exploration(last_exploration_state)
    else:
        frontier = select_optimal_frontier(frontiers,
robot_position, information_gain)
        path = plan_path_with_rrt(robot_position, frontier)
        navigate_to(path)
```

Fig. 2. Pseudocode.

G. Map Representation

Each cell in the two-dimensional grid representation of the environment represents a tiny area of the real world. Each cell's condition, which may be divided into three categories, offers crucial information about the robot's environment:

1) Free Space: Navigable and verified obstacle-free cells are designated as "free space." The robot can roam freely in these spaces without worrying about collisions. (White area)

2) Unknown Areas: The robot hasn't yet investigated these areas. Unknown cells are important because they symbolize the robot's ignorance of its surroundings. Through methodical data collection, the robot's exploratory duty is intended to progressively decrease these uncharted regions. (Grey area)

3) Obstacles: The robot labels cells that it recognizes as having physical boundaries or walls as obstacles. The robot steers clear of certain regions during navigation since they are regarded as impenetrable. (Occupied cells / Black walls)

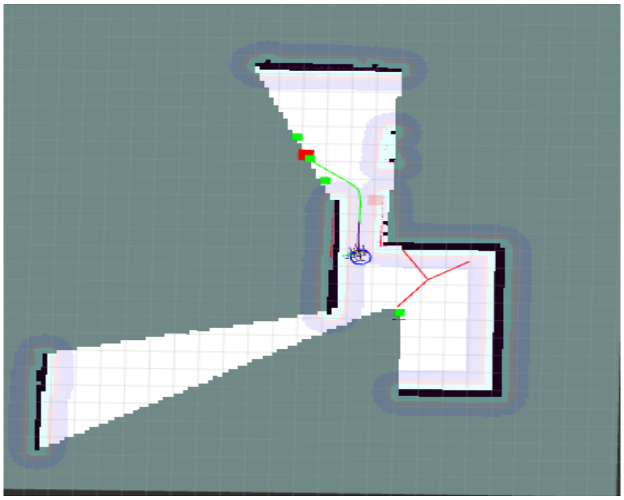

Fig. 3. Simulation Map

IV. Implementation and Testing

A. Simulation Setup

This study's robot is a Kobuki differential drive robot, which is widely employed in robotics research because of its sturdy construction, usability, and agility. By regulating the relative speeds of each wheel, Kobuki robot's have two independently driven wheels allow for precision exploration and navigation in two-dimensional surroundings. Path-planning and exploration algorithms can be implemented and tested in structured and semi-structured environments with this configuration.

In robotics research, the Kobuki robot is a common differential drive platform that is small, sturdy, and adaptable. Its passive caster wheel and two independently driven wheels allow for accurate movement and agility in 2D surroundings. With key sensors like bumper sensors for collision detection, a gyroscope for orientation, and wheel encoders for odometry, the robot offers dependable feedback for mapping and navigation tasks. It is the perfect platform for creating and testing autonomous systems because of its modular design and smooth interface with ROS (Robot Operating System). The Kobuki robot is especially well-suited for putting battery-aware exploration algorithms into practice and verifying them because it also has real-time battery monitoring and automatic docking for recharging. In this arrangement:

The main platform for managing the Kobuki robot and putting the exploratory framework into practice is ROS (Robot Operating System). A modular middleware framework offered by ROS makes it easier to integrate sensors, control robots, and communicate in real time. Because of its adaptability, the robot's operations can easily incorporate path-planning, battery monitoring, and recharging procedures. The Kobuki robot's interaction is replicated in a virtual 2D world using Gazebo Simulation world. Realistic physics simulations are available on Gazebo, including:

Collision detection: Making sure the robot steers safely and stays clear of obstructions.

Wheel Dynamics: Modeling the acceleration, friction, and slippage of Kobuki's wheels as they would appear in the actual world.

Environmental Interactions: Precisely simulating how the robot interacts with its environment, including impediments or dynamic changes in the map.

The robot's environment is represented as a 2D occupancy grid with clearly defined free space, unknown areas, and obstacles. As the robot maps and investigates its surroundings, the grid is updated dynamically. Several pre-established charge stations are positioned thoughtfully throughout the map to assess the robot's capacity to effectively handle battery limitations and find the closest charging station. The robot can prioritize unexplored regions and flexibly modify its exploration course thanks to constant frontier detection and updates. ROS (Robot Operating System), which offers a middleware framework for robot control and communication, is used to simulate the robot in a two-dimensional environment. Real-world physics, like as collision detection, wheel dynamics, and ambient interaction, are replicated in the Gazebo simulation environment. This guarantees that the simulated performance closely resembles behavior in the real world.

B. Test Scenarios

1) Single Charging Station:

- Objective: Evaluate the robot's ability to navigate back to a single fixed charging station when the battery level drops below the threshold.
- Setup: The charging station is placed at a fixed location (e.g., one corner of the map). The robot is tasked with exploring the environment while managing its battery constraints.
- Observation: The robot successfully detected its low battery state and calculated the shortest path to the charging station. After recharging, it resumed exploration from its previous location, demonstrating seamless integration of exploration and energy management.

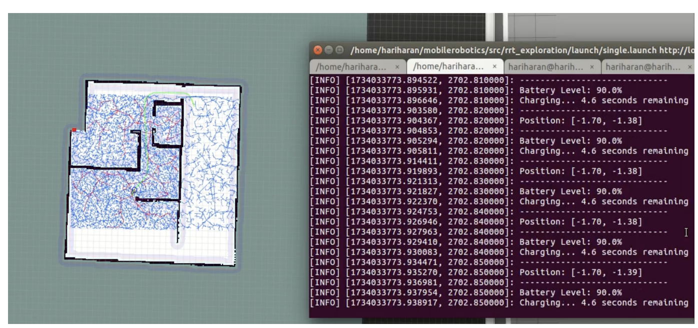

Fig. 4. Single Charging Station situated at point (-1.70, -1.38)

2) Two Charging Stations:

- Objective: Assess the efficiency of dynamically selecting the nearest charging station in a map with two stations.
- Setup: Charging stations are distributed across the environment, with equal or varied spacing. the robot must identify the optimal station based on its current position and battery level.

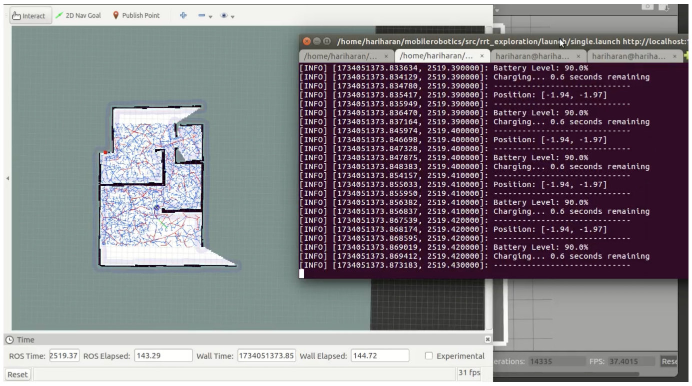

Fig. 5. Two Charging Station exploration First charging station at point (-1.70, -1.38)

- Observation: The robot consistently selected the nearest charging station, minimizing travel distance and time spent recharging. This scenario highlighted the algorithm's ability to adapt to dynamic recharging needs, improving overall efficiency.

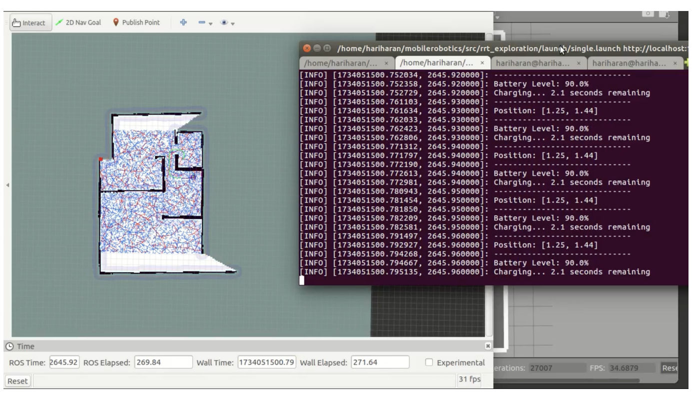

Fig. 6. Two Charging Station exploration Second charging station at point (1.25, 1.44)

V. Results

This is the gazebo house world which the robot will explore and map.

The robot's exploratory duty begins with recognizing environmental frontiers. Using the RRT method, it plots a course toward the best frontier point, selecting unexplored regions based on proximity and possible information acquisition. The robot moves across open space, avoiding impediments and dynamically updating its occupancy grid.

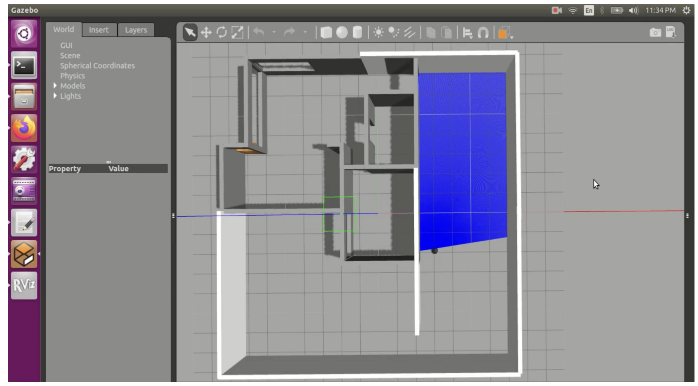

Fig. 7. Gazebo Simulation

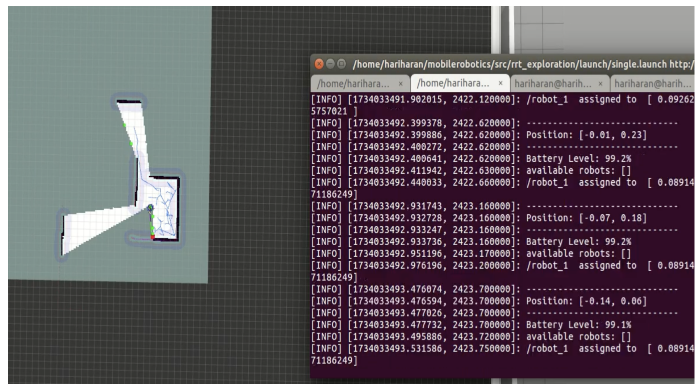

Fig. 8. Robot starts exploration.

When the robot's battery level falls below a certain threshold (e.g. 30%), it stops exploring and calculates the quickest path to the nearest charging station at (-1.72, -1.41). The robot uses real-time sensor data and path-planning algorithms to dynamically avoid obstacles, ensuring safe and efficient navigation to the station.

When the robot reaches the charging station, it docks and begins the recharging procedure. The mechanism ensures that the robot remains immobile until its battery is fully charged. This phase is critical for ensuring continuous exploratory missions by limiting battery depletion.

When the battery is completely charged, the robot easily restarts its investigation from where it left off. The system returns to the previous recorded exploration state, ensuring little disturbance and continuity in the mapping process.

After investigating all frontiers and covering the entire map, the robot successfully completes its exploration mission. The final occupancy grid represents a fully mapped environment, exhibiting the robot's capacity to efficiently manage energy restrictions while providing extensive coverage.

VI. Conclusion and Future Work

This study introduced a battery-aware frontier exploration framework that was combined with the RRT algorithm and validated using simulations on a Kobuki differential drive robot. The robot successfully displayed smooth transitions

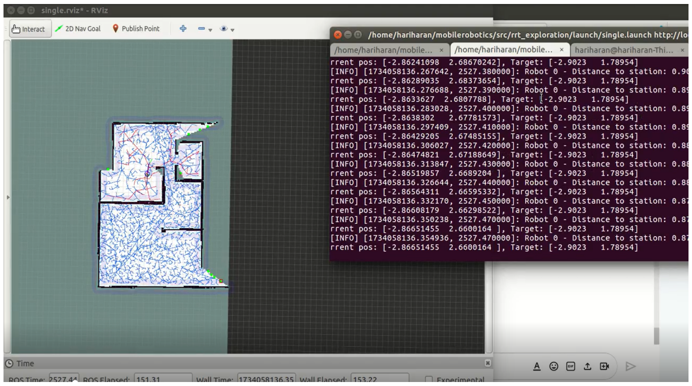

Fig. 9. Robot moving towards charging station.

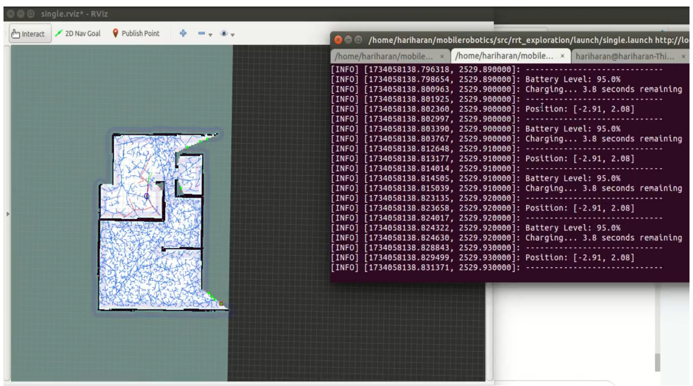

Fig. 10. Robot starts charging.

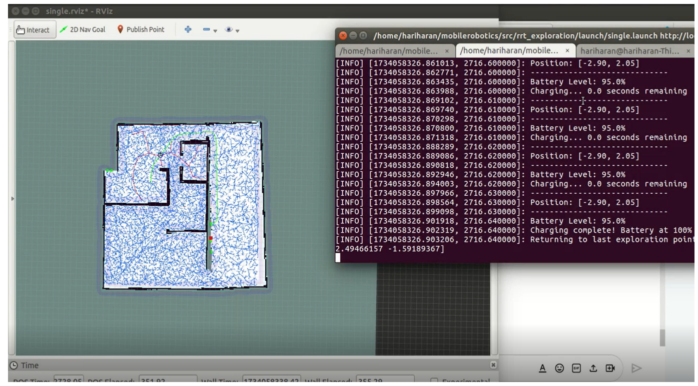

Fig. 11. Robot starts exploration after charging to 100%

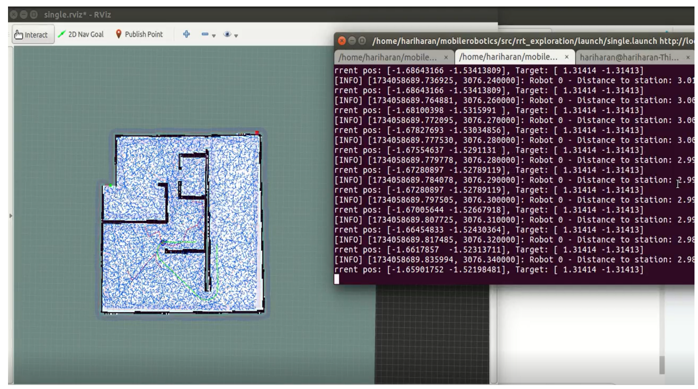

Fig. 12. Robot completes exploration.

between exploring and recharging activities, assuring continuous functioning. The system balanced energy restrictions with exploration objectives by dynamically prioritizing frontiers and efficiently traveling to charging stations when battery levels fell below a certain threshold. Predefined charging stations and dynamic path-planning enabled the robot to adapt to a variety of settings, including single and multiple charging station combinations.

The findings demonstrate the framework's usefulness in assuring consistent exploration while resolving energy constraints. After recharging, the robot resumed exploring on a continuous basis, ensuring mapping continuity and maximum coverage. However, the simulations revealed several trade-offs, such as the increased recharge frequency required for larger exploration radii. These findings underscore the need of striking a balance between exploration speed and energy efficiency in autonomous robotic systems.

This work highlights the potential for real-world applications such as search and rescue missions, environmental monitoring, and warehouse automation, all of which face major energy restrictions. Future studies could solve restrictions like fixed charging station locations by investigating dynamic deployment tactics or collaborative multi-robot systems. Additionally, improving the program to handle 3D environments for drones or underwater robots could increase its application in a variety of settings.

Unless there are six authors or more give all authors' names; do not use "et al.". Papers that have not been published, even if they have been submitted for publication, should be cited as "unpublished" [**?**]. Papers that have been accepted for publication should be cited as "in press" Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation.

References

[1] B. Yamauchi, "A frontier-based approach for autonomous exploration," Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation, Monterey, CA, USA, 1997, pp. 146–151. doi: 10.1109/CIRA.1997.613851.

[2] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Computer Science Department, Iowa State University, Ames, IA, 1998.

[3] H. Umari and S. Mukhopadhyay, "Autonomous robotic exploration based on multiple rapidly-exploring randomized trees," 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 2017, pp. 1396-1402, doi: 10.1109/IROS.2017.8202319. keywords: Detectors;Image edge detection;Robot sensing systems;Path planning;Navigation,