

COLLABORATIVE ALGORITHMS FOR LOCALIZATION AND EXPLORATION IN MULTI-ROBOT SYSTEMS

by

EHSAN LATIF

(Under the Direction of Ramvijas Parasuraman)

This dissertation contributes to the collaborative multi-robot systems literature, which is predominantly hindered by reliance on expensive sensors and predefined learning models, scalability issues, high communication costs, and difficulties adapting to environmental changes. The research proposes innovative algorithms and strategies for improving localization and exploration capabilities for multi-robot collaboration in ad hoc networks.

We introduce an online localization algorithm that enhances accuracy and efficiency by combining wireless sensor nodes (WSN) and mobile robots, leveraging Radio Signal Strength Indicator (RSSI). A relative localization technique is proposed to enhance this algorithm, effectively working in scenarios lacking infrastructure information. It combines graph optimization and Gaussian process regression, surpassing traditional model-based methods. A mechanism is designed to provide resilience to robotic systems, making localization solutions failure-tolerant, thus ensuring high positional accuracy in real-world applications.

Additionally, we develop a learning-based framework for simultaneous localization and adaptive exploration by generating virtual maps for efficient exploration in dynamic environments. This methodology is founded on Bayesian inference over the Gaussian probability distribution of wireless signals. A strategy is presented for coordinated map exploration that integrates Q-learning for efficient path planning. We devise a feature-matching map-merger strategy to create a consistent map from sparse maps collected from collaborating robots. An enhanced reinforcement learning technique is deployed for maze exploration, providing superior efficiency in challenging environments by increasing coverage and reducing overlap. We also investigate an integrated framework for simultaneous exploration and localization, which is crucial for robotic operations in dynamic environments.

The research comprehensively addresses the critical issues facing current CMRS by introducing novel algorithms for improved localization and exploration. These strategies demonstrate significant improvements in scalability, communication, resilience to node failure, and adaptability to dynamic environments. The solutions have proven effective in both theoretical models and real-world applications, strengthening their viability. By setting new standards in autonomous robotic collaboration, this dissertation provides a strong foundation for future research in the field.

INDEX WORDS: Multi-Robot Systems, Collaborative Algorithms, Localization, Exploration, Graph Optimization, Gaussian Processes Regression, Bayesian Inference, Robot Exploration, Q-Learning, Map Merging, Reinforcement Learning, Wireless Sensor Networks, Internet of Things, Networked Robots, Ad hoc Networks, Maze Solving

COLLABORATIVE ALGORITHMS FOR LOCALIZATION AND
EXPLORATION IN MULTI-ROBOT SYSTEMS

by

EHSAN LATIF

B.S(CS), National University of Computer and Emerging Sciences, Pakistan,
2018

A Dissertation Submitted to the Graduate Faculty of the
University of Georgia in Partial Fulfillment of the Requirements for the
Degree.

DOCTOR OF PHILOSOPHY

ATHENS, GEORGIA

2023

©2023
Ehsan Latif
All Rights Reserved

COLLABORATIVE ALGORITHMS FOR LOCALIZATION AND
EXPLORATION IN MULTI-ROBOT SYSTEMS

by

EHSAN LATIF

Major Professor: Ramvijas Parasuraman

Committee: Liming Cai
Wenzhan Song

Electronic Version Approved:

Ron Walcott
Dean of the Graduate School
The University of Georgia
August 2023

DEDICATION

I dedicate this dissertation to my supervisor/mentor, Dr. Ramvijas Parasuraman, who has enabled me to perform multi-robotic research and directed me to the right path to a bright future. Furthermore, I also dedicate this dissertation to the faculty members of the School of Computing, who advised me accordingly in every aspect during my doctoral studies. Last but not least, the support of my parents played a vital role in providing positive validation for my doctoral journey.

ACKNOWLEDGMENTS

Words cannot express my gratitude to my professor Dr. Ramvijas Parsuraman and the members of my advisory committee for their invaluable support and feedback. I could not have undertaken this journey without my defense committee, who generously provided knowledge and expertise. Additionally, this endeavor would not have been possible without the generous support from the Higher Education Commission Pakistan, which provided a scholarship for my doctoral studies.

I am also grateful to my friends and cohort members, especially my lab mates, for their editing help, late-night feedback sessions, and moral support. Thanks should also go to the technical support team, undergraduate research assistant Michael Starks, and study participants from the university, who impacted and inspired me.

Lastly, I would be remiss in not mentioning my family, especially my parents. Their belief in me has kept my spirits and motivation high during this process. I would also like to thank Netflix and Youtube for all the entertainment support.

CONTENTS

Acknowledgments	v
List of Figures	ix
List of Tables	xv
1 Introduction	1
1.1 Multi-Robot Systems	1
1.2 Localization in MRS	3
1.3 Exploration in MRS	4
1.4 The Interplay of Exploration and Localization	5
1.5 Existing Approaches and Their Limitations	5
1.6 Proposed Solutions	7
1.7 List of Publications	8
1.8 Contributions	9
1.9 Dissertation Overview	10
1.10 Open Science Contribution	12
2 Background	13
2.1 Multi-Robot System (MRS)	13
2.2 Localization	15
2.3 Exploration	22
2.4 Simultaneous Localization and Mapping	25
3 Instantaneous Wireless Robotic Node Localization Using Collaborative Direction of Arrival	28
3.1 Introduction	28
3.2 Related Work	31
3.3 Problem Statement	34
3.4 Proposed CDOA Approach	35
3.5 Theoretical Analysis	39

3.6	Experimental Validation	48
3.7	Results	51
3.8	Summary	57
4	Multi-Robot Synergistic Localization in Dynamic Environments	60
4.1	Introduction	60
4.2	Related Work	63
4.3	Proposed Approach	65
4.4	Experimentation and Results	71
4.5	Summary	77
5	DGORL: Distributed Graph Optimization based Relative Localization of Multi-Robot Systems	79
5.1	Introduction	79
5.2	Related Work	82
5.3	Problem Formulation and the Proposed DGORL Solution	83
5.4	Theoretical Analysis	89
5.5	Simulation Experiments and Results	91
5.6	Summary	94
6	GPRL: Multi-Robot Relative Localization Using Hierarchical Gaussian Processes Inferencing on RSSI Map of the Wireless Access Point	95
6.1	Introduction	95
6.2	Related Work	97
6.3	Problem Formulation	99
6.4	Source Positioning using Hierarchical Inferencing over GPR	100
6.5	Experimental Setup	106
6.6	Results and Discussion	108
6.7	Summary	113
7	Communication-Efficient Reinforcement Learning in Swarm Robotic Networks for Maze Exploration	115
7.1	Introduction	115
7.2	Related Work	117
7.3	Proposed Approach	119
7.4	Experimental Evaluation	123
7.5	Summary	127

8	CQLite: Communication-Efficient Multi-Robot Exploration Using Coverage-biased Distributed Q-Learning	128
8.1	Introduction	128
8.2	Related Work	130
8.3	Proposed Approach	135
8.4	Experimental Validation	150
8.5	Summary	157
9	SEAL: Simultaneous Exploration and Localization in Multi-Robot Systems	159
9.1	Introduction	159
9.2	Related Work	162
9.3	Problem Formulation	164
9.4	Approach and Algorithms	166
9.5	Experimental Validation	174
9.6	Summary	178
10	Conclusion	180
10.1	Future Work	181
	Appendices	183
	A	183
	Bibliography	188

LIST OF FIGURES

1.1	Interplay of navigation, exploration, and localization overview.	6
1.2	Overview of collaborative algorithms proposed in dissertation	11
2.1	Multi-Robot System: robot, locally connected MRS, group of MRS connected through the network	14
2.2	Robot localizing in a global workspace using wireless sensor nodes.	16
2.3	Particle filters being propagated and updated	18
2.4	Samples expectation and maximization process	19
2.5	Simulated world with explored map and exploration sequence	23
3.1	Overview of the proposed real-time CDOA-based localization of a robotic node following a trajectory with limited IoT/Wireless Nodes.	29
3.2	System architecture of the CDOA-based robotic node localization approach using particle filter and estimation maximization.	40
3.3	Depiction of coverage area for CDOA localization.	44
3.4	The test sample locations in the simulation experiments are shown here. The plots show the robot’s actual trajectory (where the signal sample was taken) in solid lines and the predicted locations in scattered dots.	46
3.5	Locations of the RSSI sample points from the real-world datasets 1 (Left) and 2 (Right) discussed in Sec. 3.6.3.	47
3.6	Hardware experiment setup and samples of the output robot’s trajectories (diagonal, inside, boundary) in the real robot experiments (See Sec. 3.7.3).	48
3.7	Comparative localization performance in the simulation environment.	52

3.8	Localization performance under different RSSI noise levels. The embedded plot represents the RSSI variation at different noise levels to visualize the representativeness of the simulated RSSI with real-world data.	53
3.9	Impact on the localization accuracy of CDOA-PF by the number of particles and inclusion of robot motion model (odometry).	54
3.10	Comparative localization performance on the real-world dataset I.	55
3.11	Comparative localization performance on the real-world dataset 2.	56
3.12	Comparative localization performance on the real robot experiments.	57
4.1	System architecture showing input and output for robot i . External robot position estimates and RSSI are shown in circles out of the gray box as robot j : (x_j, y_j, w_j) and $RSSI_j$ are the best-fit pose estimates, and RSSI of robot j from its particle filter, (x_i, y_i, w_i) indicates the state estimate of robot i based only on received measurements from connected n robots in the MRSI.	61
4.2	Overview of the proposed multi-robot synergistic localization, robots localizing themselves using local information as well as shared information.	65
4.3	The simulation setup for three robots in a 6m x 6m workspace is shown here. It also depicts the two virtual walls in the simulation workspace to create non-line-of-sight conditions. Each wall will attenuate 10 dBm of RSSI signal power.	71
4.4	Localization performance of different algorithms regarding Root Mean Squared Error (RMSE) compared to the ground truth location in the 6m x 6m simulation experiments.	73
4.5	Trajectory of three cooperative robots in the 6m x 6m simulation experiments are shown here. The outputs of localization algorithms are plotted to comparatively observe their performance against the ground truth.	74
4.6	Localization accuracy comparison for varying number of robots	74
4.7	Localization accuracy comparison for non-line-of-site scenario	75
4.8	Localization accuracy comparison for communication challenge: 70% Packet Drops	75

4.9	Localization performance of different algorithms regarding Root Mean Squared Error with three robots in the 60m x 60m simulation workspace.	76
5.1	Overview of configuration space of a multi-robot system, sharing their pose (x_i) and relative range (R_i^j) measurements in our DGORL solution.	80
5.2	The Distributed Graph Optimization for Relative Localization of Multi-Robot Systems (DGORL) system architecture shows input (Received connected robot's motion information and RSSI) and output (relative pose estimation) for robot i . The robot generates and expands graphs based on observability for optimization along with the constraints set up through local and received inertial measurement, which further fed into the optimization function. The optimized graph yields relative pose estimates for all connected robots.	85
5.3	Trajectories and the Optimized Graph. Left: Simulation with the ground truth (green) trajectory along with the colored predicted trajectory of a robot; Right: Robots as colored vertices and colored edges between connected robots with the respective color for optimized graphs.	91
5.4	DGORL performance of five robots in terms of Root Mean Squared Error (RMSE) in the 60m x 60m simulation workspace (Top). Comparison with Collaborative Terrain Relative Navigation (TRN) based on Covariance Intersection (Bottom).	92
6.1	An overview of GPRL with hierarchical inferencing with three levels and relative localization with vector transformation using source position. α, β and γ are resolution parameters for $N \times N$ grid space, ${}_i\mathbf{p}^{AP}$ and ${}_j\mathbf{p}^{AP}$ are position vectors from the robot's position to the predicted AP position, and ${}_i\mathbf{p}^j$ is a relative position vector from robot i to j	96
6.2	A distributed system architecture of GPRL for robot i to perform relative localization using GPR and hierarchical GP inferencing.	99

6.3	Source Prediction performance evaluation comparison plots of SOTA approaches per iteration; MEGPR (M. Xue et al., 2019) (sparse and dense AP subset) and the proposed approach (GP-based source inferencing with hierarchical inferencing and without hierarchical inferencing) along with the effect of hierarchy.	107
6.4	Results of various performance metrics. From left to right, we show the localization accuracy (RMSE) under different simulated RSSI noise levels (the embedded plot represents the respective RSSI variations), the communication payload, the computation overhead, and the run time performance of various relative localization approaches. It can be seen that the GPRL consistently outperforms other approaches in all the metrics.	109
6.5	Gaussian process inferencing for global and local frames along with ground truth and predicted robot trajectories.	110
6.6	Example trial from the multi-robot rendezvous experiment: Initial (top) and final (bottom) state of the robots and their trajectories using GPRL.	112
7.1	Overview of the Communication-Efficient Reinforcement Learning.	116
7.2	Core Simulator (Right) with exploration map (Left).	123
7.3	Time taken by maze coverage strategies for three levels of coverage (one-third, two-third, and full).	123
7.4	Coverage percentage performance metric: Average results over all mazes (Left), after 100 iterations (Center); after 400 iterations (Right).	124
7.5	Map overlap percentage metric: Average results over all mazes (Left), after 100 iterations (Center); after 400 iterations (Right).	124
7.6	Impact of the communication range parameter; number of iterations (Left), average coverage percentage (Center), average overlap percentage (Right).	125
7.7	Impact of the packet loss parameter; number of iterations (Left), average coverage percentage (Center), average overlap percentage (Right).	125
8.1	Overview of the distributed CQLite method for efficient multi-robot exploration, shown with an illustrative simulation.	129

8.2	System architecture of CQLite distributed across several robots. It shows the Robot i 's process showing the mapping, frontier detection, and Q-learning operations along with the communication of local map and updated Q-value information to n connected robots.	135
8.3	A depiction of the outcome in a sample trial. It shows the map generated by three robots in the house world (left column) and three and six robots in the bookstore world (center and right column, respectively) created by the three compared approaches; RRT (top), DRL (center), and CQLite (bottom), with robots moving in a simulated House and Bookstore worlds along with the following trajectories, start and end locations.	150
8.4	Computation (Left), Communication (Center) cost, and Exploration over time (Right) comparison plot of CQLite with RRT and DRL approach in three Gazebo simulated world. Row-wise: Top 3 robots in house world, Middle 3 robots in bookstore world, and Bottom 6 robots in bookstore world.	151
8.5	Exploration map of house world before, during, and after map sharing and merging corresponds to points (A, B, and C from Fig. 8.4) in Computation and Communication plots. Peaks demonstrate the request for map merging in CQLite for Computation and communication plots; RRT runs longer with persistent high communication and computational overhead but explores fewer regions than DRL and CQLite.	152
9.1	Overview of the proposed simultaneous exploration and localization in the multi-robot system, shown with the ROS Gazebo simulation, configuration for relative localization same as DGORL (Latif & Parasuraman, 2022a), and the exploration process with the convex hull	160
9.2	System architecture of SEAL distributed across several robots. It shows Robot i 's process of exploring, convex hull optimization, and relative localization operations using shared information from n connected robots.	165

9.3	Exploration grid maps along trajectories: The top pictures show the Bookstore world and the bottom pictures show the House world. (Top Left) Explored map using RRT (L. Zhang et al., 2020), (Top Center) Explored map using proposed SEAL, (Top Right) Explored maps' confidence (belief map) of SEAL. (Bottom Left) Explored map using DRL (Hu et al., 2020), (Bottom Center) Explored map using proposed SEAL, (Bottom Right) Explored maps' confidence (belief map) of SEAL. <i>Confidence maps are provided only by SEAL.</i>	174
9.4	Exploration progress (Left) and localization error (Right) of the proposed SEAL, RRT (L. Zhang et al., 2020), and DRL (Hu et al., 2020) based exploration and mapping.	175
9.5	(Left) Localization error comparison with RRT (L. Zhang et al., 2020) and DGORL (Latif & Parasuraman, 2022a), (Right) Predicted trajectory of the three robots for different approaches.	176
9.6	Localization performance of SEAL for the individual robot.	177
9.7	Overall localization error comparison with RRT (L. Zhang et al., 2020) and DGORL (Latif & Parasuraman, 2022a) for complete exploration.	178
A.1	A sample output comparison of different localization algorithms in a 40 x 40 m bounded region.	187

LIST OF TABLES

3.1	Notations and their descriptions	36
3.2	Experiment Configurations	48
3.3	Overall performance results with statistics of various experiments from simulations, datasets, and hardware trials.	51
6.1	Resolution (m) of the GPR prediction map in Robotarium simulations. Note the environment size is 3mx2m.	106
7.1	Comparative complexity analysis of maze exploration.	121
8.1	Performance Results of RRT, DRL, and the proposed CQLite. * indicates the best performer.	153
9.1	Performance results of different approaches. * indicates the best performer. L - Localization. E - Exploration.	173

CHAPTER I

INTRODUCTION

The advent of robotics has revolutionized numerous sectors, ranging from healthcare and agriculture to logistics and defense, significantly changing how we operate in these domains (Burgard et al., 2005). As robotics technology continues to mature, systems that include multiple robots, known as Multi-robot systems (MRS), are becoming more prevalent. MRS offers unique advantages over single-robot systems, especially in tasks that require extensive area coverage, task parallelization, and resilience to individual robot failures (Grisetti et al., 2007). Such systems are thus well-suited for complex applications, such as search and rescue operations, environmental monitoring, and surveillance missions (Burgard et al., 2005; Queralta et al., 2020).

I.1 Multi-Robot Systems

In recent decades, there has been a significant surge in the usage of autonomous robotic systems across various applications, from industrial automation and logistics to surveillance, environmental monitoring, search and rescue, and even space exploration (Queralta et al., 2020). These applications often call for the deployment of not just a single robot. Still, a team of robots, or a multi-robot system (MRS), can handle tasks beyond the scope of a single agent, be it due to the size of the environment, time constraints, or the complexity of the task itself. An MRS promises enhanced performance, improved robustness through redundancy, and the ability to handle large-scale or complex tasks through distributed cooperation (Schmuck & Chli, 2019).

1.1.1 Challenges

Despite their numerous advantages, implementing MRS comes with unique challenges. These primarily revolve around communication, coordination, task allocation, and navigation (Olcay et al., 2020). Among the myriad challenges of deploying MRS, two that stand out are exploration and localization. Exploration refers to acquiring knowledge about an unknown environment (Burgard et al., 2005), typically in the form of a map. Localization, on the other hand, involves determining the position and orientation of each robot within the environment, given the map (He et al., 2020). These complex, interdependent tasks constitute the bedrock upon which other higher-level tasks, such as planning and navigation, are built. Among these, this dissertation primarily focuses on the challenges associated with navigation – specifically, the twin tasks of exploration and localization.

1.1.2 Significance

In any autonomous system, understanding and navigating the environment is fundamental. This involves two interconnected components: exploration, which deals with the identification and mapping of unknown environments, and localization, which focuses on the robot’s ability to position itself within that environment. The performance of a multi-robot system in real-world applications largely depends on the efficiency and accuracy of these two tasks (Arshad & Kim, 2021).

Unique Considerations

The complexity of exploration and localization tasks increases when it comes to MRS. It becomes necessary for the robots to coordinate their actions to efficiently explore the environment, avoid collisions, and achieve the overarching task objective. Further, each robot has to maintain an estimate of its position and that of the other robots, given the observed data and the actions performed by the entire team (Q. Li et al., 2020). These considerations introduce unique challenges in decision-making, information sharing, and data fusion, especially when communication bandwidth is limited or the environment is dynamic or hostile (Masaba & Li, 2021).

1.1.3 Objectives

The overall objective of my research is to design infrastructure-less multi-robot algorithms. Specific research objectives are mentioned following:

Localization: Global and relative localization solutions aim to estimate robot positioning information in an environment with negligible outsourcing of computation and communication.

Exploration: Coordination of multiple robot explorations aims to maximize expected information gain (map knowledge) across time. Even though the optimal solution is computationally intractable, I have created a method that is both affordable and effective.

Simultaneous Exploration and Localization (SEAL) The objective of SLAE is to build a framework to fill the gap between localization and adaptive exploration by exploiting both gains.

1.2 Localization in MRS

Meanwhile, localization is equally crucial. For a robot to operate autonomously, it must accurately know its position within the environment at all times. This becomes particularly challenging in unknown or dynamic environments, where traditional positioning systems like GPS might not be reliable or available. This problem is further exacerbated in MRS, where each robot needs to localize itself and keep track of its peers' positions to maintain coordination. Common localization methods rely on sensor measurements or communication with known landmarks, but these can be prone to errors due to sensor noise or environmental factors. Therefore, a growing interest is in developing robust and accurate localization algorithms that can handle the complexities of real-world environments (Schmuck & Chli, 2019).

1.2.1 Relative Localization

Another major contribution of our work lies in the realm of relative localization. Instead of relying on global position information or features in the environment, we propose a relative localization strategy using inter-robot communication data (S. Wang et al., 2023). Our strategies employ graph optimization and Gaussian process modeling frameworks for fusing the data from different robots, thereby facilitating the computation of their relative positions. These methods are robust to sensor noise and uncertainties, making them well-suited for real-world applications where GPS data may be unavailable or unreliable (C. Wang, Luo, et al., 2021).

1.3 Exploration in MRS

In the context of exploration, each robot in the MRS needs to traverse the unknown environment and gather as much information as possible to construct a detailed map. The challenge lies in coordinating the robots' movements to ensure efficient coverage of the entire environment. The robots should ideally spread out to cover different areas but also need to share information about their discoveries to build a coherent global map. Traditional exploration methods, which rely on deterministic rules or random movements, fall short when dealing with complex and dynamic environments. Therefore, there is a need for intelligent exploration strategies that allow robots to learn from their experiences and adapt their behaviors accordingly. This has led to the application of machine learning techniques, particularly reinforcement learning, in robot exploration (Shrestha et al., 2019).

1.3.1 The Need for Efficient Communication and Computation

In multi-robot systems, maintaining consistent estimates of the environment and the state of the robots is a key challenge. Given that each robot has a limited perception of the environment, it becomes necessary for the robots to share information to maintain a global perspective. This calls for efficient communication strategies that can handle the exchange of large volumes of data in real-time in the face of limited communication bandwidth and potential losses or delays in transmission (Gielis et al., 2022). Further, fusing the data collected by the robots into a consistent global estimate is computation-intensive, requiring efficient algorithms that can handle the uncertainty and noise inherent in sensor data (Ghazal, 2022).

1.3.2 Distributed Exploration

Autonomous robots must use advanced decision-making mechanisms to navigate and explore unknown environments efficiently. Our work introduces a novel strategy that leverages the power of distributed linearized convex hull optimization to guide robots in choosing their next-best points for exploration. This strategy capitalizes on the ability of the convex hull approach to ensure the maximum coverage of the unknown region by considering all possible movements of the robots (Pham et al., 2009).

Unlike traditional methods that minimize the map's uncertainty, our approach also considers the robots' movements to provide a more comprehensive

and efficient exploration strategy. Doing so enables faster and more complete exploration of the environment, thereby contributing to enhanced system performance and resource efficiency.

1.4 The Interplay of Exploration and Localization

The issues of exploration and localization are inherently intertwined in the context of autonomous robotic systems, especially multi-robot systems. While exploring an environment, a robot makes decisions based on its current understanding of the surroundings and its location within it. Inaccuracies in localization, thus, lead to suboptimal exploration strategies. Conversely, a robust exploration strategy provides more information about the environment, subsequently improving the performance of localization algorithms (Qin et al., 2019). As such, the tasks of exploration and localization are cyclical, with improvements in one aspect potentially leading to enhancements in the other.

Moreover, the complexity of these tasks escalates in the context of multi-robot systems. The existence of multiple, independently operating robots introduces the need for efficient communication and data-sharing protocols to coordinate exploration activities and to maintain a consistent understanding of each robot's location. In the absence of global localization information, such as GPS signals, the challenge is to develop strategies for relative localization, where robots rely on inter-robot interactions and sensor data to infer their positions (S. Wang et al., 2023). Fig. 1.1 has shown the interplay of localization and mapping for multi-robotic integrated exploration.

The unique challenges and opportunities presented by the intertwined nature of exploration and localization in multi-robot systems have sparked considerable research interest. As we will discuss next, various approaches, ranging from deterministic rules to sophisticated machine learning techniques, have been proposed to tackle these issues.

1.5 Existing Approaches and Their Limitations

Several exploration strategies have been proposed for multi-robot systems, each with its unique strengths and weaknesses. Information-based methods, such as those based on Shannon entropy, aim to minimize the uncertainty of the environmental map by guiding the robots to regions with the highest expected information gain (Botteghi et al., 2020). Other techniques, like frontier-based

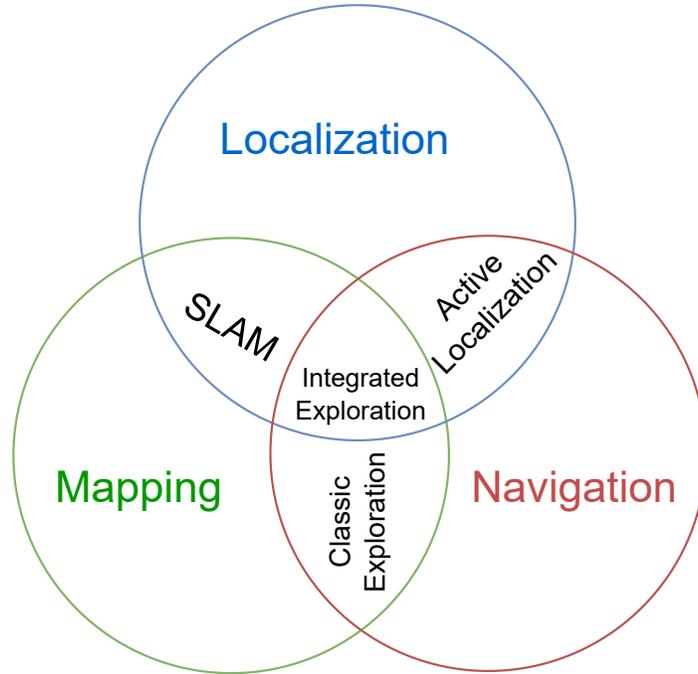


Figure 1.1: Interplay of navigation, exploration, and localization overview.

methods, direct the robots to the boundaries between known and unknown map regions (Dai et al., 2020).

Despite their merits, these traditional methods often fall short in dynamic and complex environments. They do not account for the evolving nature of the environment and cannot adapt their strategies in response to new information. To overcome these limitations, learning-based exploration methods, particularly those based on reinforcement learning, have gained popularity (Z. Zhang, Wang, et al., 2022). By incorporating learning mechanisms, these methods enable the robots to continuously improve their exploration strategies based on the outcomes of their past actions.

On the localization front, a popular approach is a Simultaneous Localization and Mapping (SLAM), where each robot constructs a map of the environment while concurrently estimating its position within it (J. Liu et al., 2022). However, SLAM methods require significant computational resources and often struggle in environments with limited features or without global position information (Zhao et al., 2021). As such, there is a pressing need for more efficient and robust localization strategies that can operate under various constraints.

The literature has presented numerous exploration techniques, including reinforcement learning (RL) approaches like Q-learning. Our research delves

into this area, offering a novel distributed Q-learning technique, CQLite, to minimize data communication overhead between robots while ensuring rapid convergence and thorough coverage in multi-robot exploration (Shrestha et al., 2019). CQLite uses ad hoc map merging, and selectively shares updated Q-values at recently identified frontiers, substantially reducing communication costs. A detailed discussion of this method forms a central part of this dissertation.

As we have considered in our studies, frontier-based exploration methods involve robots deciding their following path by searching frontier points on the border of accessible and unknown areas. These methods often produce approximate solutions due to optimization constraints but are effective in various exploration scenarios (Dai et al., 2020). We combine these approaches with RL paradigms, enabling robots to improve their competence continuously and adapt to their natural surroundings' dynamics (Z. Zhang, Wang, et al., 2022).

1.6 Proposed Solutions

This dissertation proposes a series of solutions to overcome the limitations of existing solutions for localization and exploration and integration framework for efficient exploration without global localization in a communication-restricted environment. The dissertation starts with a single robot localization solution that exploits wireless sensor network collaboration and particle filtering to accurately estimate the robot's position in the sensing range of the wireless sensor network. Using the Bayesian information fusion technique, we extend the solution to multi-robot synergistic localization. Further, we aim to overcome the lack of infrastructure or global referencing challenge and propose a relative localization based on graph optimization. While considering the computational complexity of graph optimization, we move our strategy toward Gaussian process regression-based relative localization. Until now, we have a robust solution for relative localization and moved forward to the exploration domain as it is a critical operation MRS. We begin our exploration by designing a communication-efficient solution for SWARM navigation and then extend it to a coverage-biased Q-learning-based solution for efficient exploration in a communication-restricted environment. Lastly, we integrate relative localization with exploration and devise a solution for efficient exploration without global localization. Building upon the challenges and opportunities in exploration and localization in MRS, we propose the simultaneous exploration and localization (SEAL) approach. SEAL integrates exploration and localization tasks using Gaussian Processes (GP)-based information fusion for

maximum exploration while performing communication graph optimization for relative localization (Y. Xu et al., 2021). The method leverages the Rao-Blackwellization technique to handle cross-dependent objectives to achieve optimal performance.

1.7 List of Publications

The following are the publications that led to this thesis:

- **Paper 1:** Published my work on individual robot localization with the title of “*Instantaneous Localization of Resource-Constrained Robots Using Wi-Fi Direction Estimation Through Collaborative Wireless Sensor Nodes*” (Latif & Parasuraman, 2023b) in the IEEE Internet of Things Journal (IoTJ, 2023).
- **Paper 2:** Published my work on multi-robotic synergistic localization with the title of “*Multi-Robot Synergistic Localization in Dynamic Environments*” (Latif & Parasuraman, 2022b) in the International Symposium of Robotics (ISR Europe, 2022).
- **Paper 3:** Published my work on relative localization with the title of “*DGORL: Distributed Graph Optimization based Relative Localization of Multi-Robot Systems*” (Latif & Parasuraman, 2022a) at the International Symposium on Distributed Autonomous Robotic Systems (DARS, 2022).
- **Paper 4:** Submitted my work with the title of “*GPRL: Multi-Robot Relative Localization Using Hierarchical Gaussian Processes Inferencing on RSSI Map of the Wireless Access Point*” to a journal.
- **Paper 5:** Published my work on maze exploration with the title of “*Communication - Efficient Reinforcement Learning in Swarm Robotic Networks for Maze Exploration*” (Latif et al., 2023) at the 16th International Workshop on Wireless Sensing and Actuating Robotic Networks (IEEE INFOCOM Workshops, 2023).
- **Paper 6:** Submitted my work with the title of “*CQLite: Coverage-biased Q-Learning Lite for Efficient Multi-Robot Exploration*” to a journal. A preprint of this work is available at (Latif & Parasuraman, 2023a).
- **Paper 7:** Published my work on simultaneous exploration and localization with the title of “*SEAL: Simultaneous Exploration and Localization for Multi-Robot Systems*” (Latif & Parasuraman, 2023c) at The 2023

IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS, 2023).

In addition, I also published the below papers related to Multi-Robot Systems that are not included in this thesis.

Paper 8: Published a work titled “Energy-aware multi-robot task allocation in persistent tasks” in the 6th International Symposium on Swarm Behavior and Bio-Inspired Robotics (SWARMS) conference 2022 (Latif et al., 2022).

Paper 9: Published a work titled “Message Expiration-Based Distributed Multi-Robot Task Management” in the 6th International Symposium on Swarm Behavior and Bio-Inspired Robotics (SWARMS) conference 2022 (Gui et al., 2022).

1.8 Contributions

The significant contributions of this dissertation are summarized as follows:

- In Chapter 1, we propose a novel Collaborative Direction of Arrival (CDOA) estimation algorithm using IoT or wireless nodes integrated with Bayesian frameworks for high-accuracy localization of mobile robotic nodes. The algorithm stands out by utilizing a CDOA metric obtained through cooperative communication instead of relying directly on the RSSI metric. The CDOA metric is then integrated with Expectation Maximization (EM) and Particle Filter (PF) Bayesian frameworks for robot node localization. This proposed localization solution was published as **Paper 1**.
- Chapter 2 presents a new Multi-Robot Synergistic Localization (MRSL) algorithm for decentralized synergistic localization in a dense and dynamic environment. The MRSL utilizes Bayesian rule-based integration for information fusion among robots, improving localization accuracy and computational efficiency. This multi-robot localization solution was published as **Paper 2**.
- In Chapter 3, we introduce a distributed method to solve the Multi-Robot Localization (MRL) problem as a graph optimization problem. The DGORL approach utilizes an open-source graph-based framework, g^2o , to efficiently achieve high localization accuracy. This graph optimization solution was published as **Paper 3**.

- Chapter 4 presents a unique learning method, GP-Loc, for accurate relative localization utilizing Received Signal Strength Indication (RSSI) data. This method combines Gaussian Process Regression (GPR) and hierarchical inferencing to accurately predict the location of the Wi-Fi source while being computationally efficient. This GP-based relative localization solution was submitted as a **Paper 4**.
- In Chapter 5, we explore the application of reinforcement learning (RL) for efficient maze exploration by swarm robotics. Our RL algorithm improves coverage efficiency and minimization of overlapping regions compared to existing approaches. This maze exploration solution was published as a **Paper 5**.
- Chapter 6 designs a combined learning and cooperation solution, CQLite, for efficient map exploration. CQLite leverages a distributed Q-learning methodology with a coverage-biased reward function, thus achieving fast convergence, high coverage performance, and reduced communication and update costs. This exploration solution was submitted as **Paper 6**.
- Finally, Chapter 7 introduces SEAL, a novel approach for efficient and accurate robotic localization and exploration designed for multi-robot systems. SEAL combines information-based exploration techniques, linearized convex hull optimization, and Gaussian process modeling to improve the efficiency and accuracy of robotic localization and exploration. The final integration solution was published as **Paper 7**.

1.9 Dissertation Overview

The remaining parts of this dissertation are organized as follows. Chapter 2 provides the background for our strategies in the dissertation. Chapter 3 and 4 provides a single and multi-robot localization solution, respectively. Chapters 5 and 6 discuss relative location with different strategies. Chapter 7 delves into the details of the exploration strategy, beginning with SWARM exploration, Chapter 8 provides details about the communication-efficient solution for MRS exploration, and Chapter 9 illustrates the integration strategy for simultaneous exploration and localization by providing a comprehensive explanation of the linearized convex hull optimization method and its implementation. The final chapter, Chapter 10, summarizes the findings, highlights the contributions, and discusses future work.

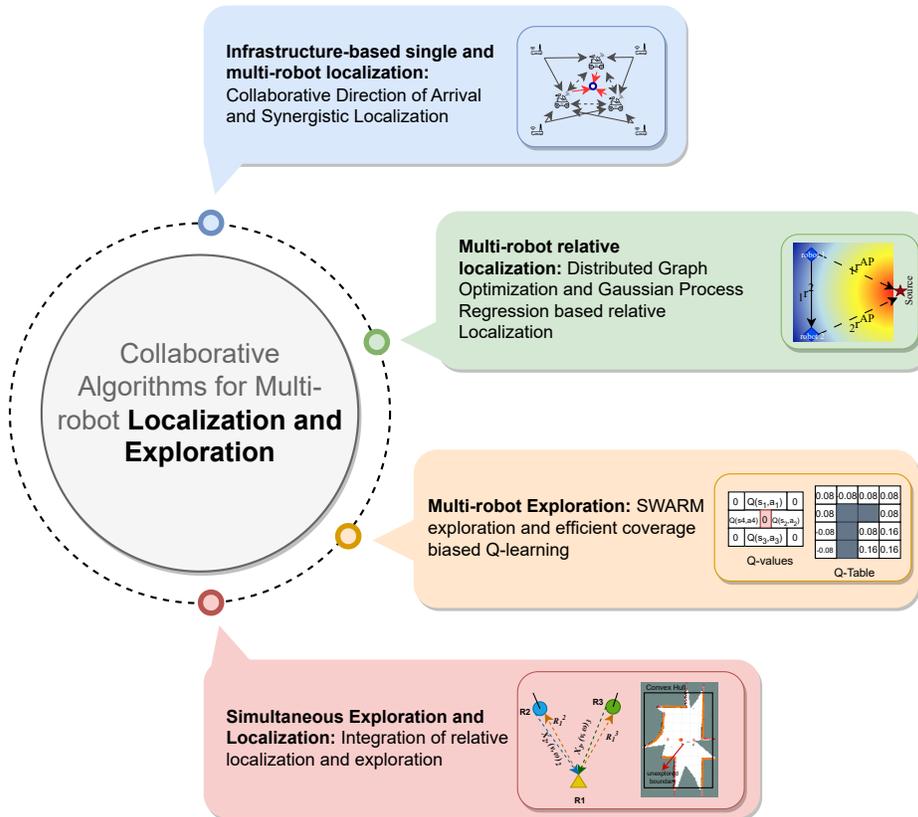


Figure 1.2: Overview of collaborative algorithms proposed in dissertation

As seen in Figure 1.2, this dissertation aims to create a cohesive solution of collaborative algorithms for localization and exploration in multi-robot networks. By pushing the boundaries of simultaneous exploration and localization, we hope to contribute to the body of knowledge in the field and open up new possibilities for multi-robot system applications.

This introduction has presented the motivation and challenges behind our work, outlined the key concepts, and set the stage for in-depth discussions in the ensuing chapters. Our journey begins here, at the intersection of exploration and localization in multi-robot systems, and we invite you to join us on this exciting voyage of discovery.

1.10 Open Science Contribution

The dissertation also contributed to the open science community with the following open-source GitHub repositories associated with the paper mentioned in section 1.7:

1. The source code associate with **paper 1** <https://github.com/herolab-uga/cdoa-localization.git>.
2. The source code associate with **paper 3** <https://github.com/herolab-uga/DGORL.git>.
3. The source code associate with **paper 4** <https://github.com/herolab-uga/gp-multi-robot-localization>.
4. The source code associate with **paper 5** <https://github.com/herolab-uga/MazeCommRL.git>.
5. The source code associate with **paper 6** <https://github.com/herolab-uga/cqlite.git>.
6. The source code associate with **paper 7** <https://github.com/herolab-uga/ROS-SEAL.git>.

CHAPTER 2

BACKGROUND

This chapter will elaborate on the terms and technologies discussed in the dissertation. We will begin with Multi-robotic systems, then localization and all the basic concepts in our proposed solution. The later part will cover the exploration, simultaneous localization and mapping, and relevant concepts.

2.1 Multi-Robot System (MRS)

An MRS refers to a group of robots that work collaboratively to achieve common goals or tasks. It involves multiple robots, each with its own set of capabilities and sensors, operating in a coordinated manner.

Mathematically, a multi-robotic system can be represented as follows:

Robot representation: Let N denote the total number of robots in the system. Each robot i can be represented by its state vector \mathbf{s}_i , which includes information about its position, orientation, and other relevant attributes. In a 2D environment, the state vector of robot i can be defined as $\mathbf{s}_i = [x_i, y_i, \theta_i]$.

Communication: Multi-robot systems often require communication between robots to exchange information and coordinate their actions. This communication can be represented by a communication graph, where each robot is represented as a node, and communication links between robots are represented as edges. The communication graph can be denoted as $G = (V, E)$, where V represents the set of robot nodes and E represents the set of communication links.

Coordination: Multi-robot systems require coordination mechanisms to synchronize the actions of individual robots and achieve desired collective behaviors. This coordination can be achieved through various approaches, such as centralized or decentralized control algorithms, consensus algorithms, or behavior-based approaches. The coordination mechanism can be represented

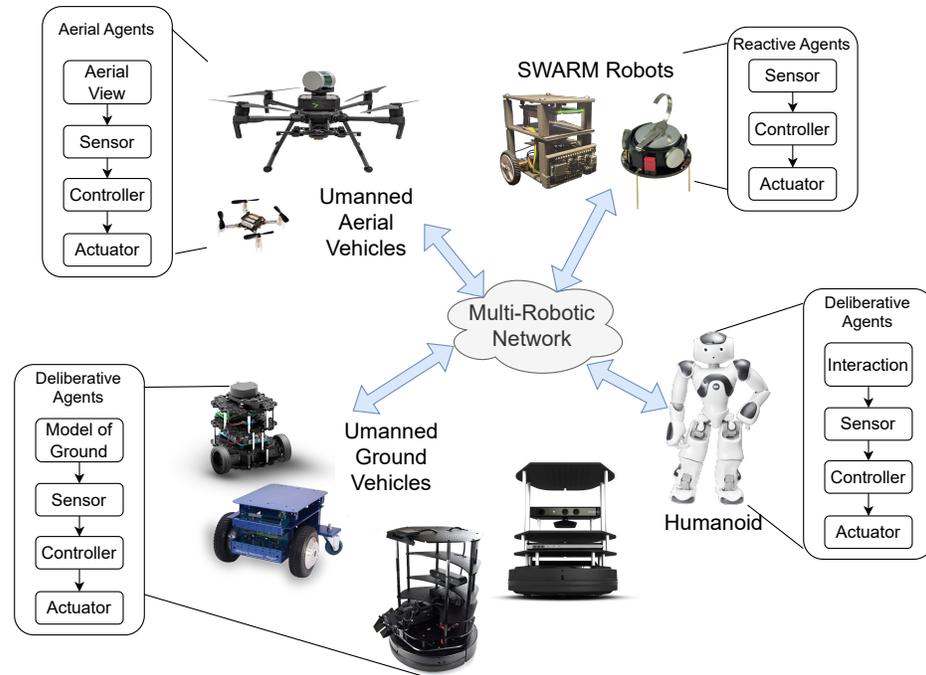


Figure 2.1: Multi-Robot System: robot, locally connected MRS, group of MRS connected through the network

by a control law or coordination function that considers the current states of the robots and the desired collective behavior.

Localization and mapping: Localization and mapping are crucial for multi-robot systems to maintain an accurate knowledge of the environment and avoid collisions or conflicts between robots. Each robot needs to estimate its pose and build a map of the environment. The localization and mapping process can be represented by localization algorithms similar to the ones described in the previous explanation.

Task allocation: In multi-robotic systems, tasks or subtasks need to be assigned to individual robots for efficient execution. This task allocation process can be represented by a function $f : V \rightarrow T$, where V is the set of robot nodes and T is the set of tasks. The function f assigns each robot to a specific task based on various criteria, such as robot capabilities, task requirements, or optimization objectives. Fig. 2.1 delineates the pictorial view of multi-robotic systems where robots are connected through a common network.

By considering the representation of individual robots, communication between them, task allocation, coordination mechanisms, and localization and

mapping, multi-robotic systems can achieve efficient collaboration and perform complex tasks that may be challenging for a single robot.

2.2 Localization

Localization in robotics refers to the process of determining the position and orientation (pose) of a robot in its environment. It involves estimating the robot's coordinates (x, y) and its orientation θ relative to a known reference frame.

Mathematically, localization can be represented as follows:

Environment representation: The robot's environment can be represented as a coordinate system, typically a 2D or 3D Cartesian coordinate system. The robot's position in this coordinate system is denoted by (x, y) , where x represents the robot's horizontal position and y represents the vertical position.

Pose estimation: The robot's orientation, or heading, is represented by θ . This is often expressed as an angle relative to a fixed reference, such as the positive x -axis. Common representations include radians or degrees.

State estimation: The robot's state can be represented as a vector, often denoted by \mathbf{s} , which combines both position and orientation information. In a 2D environment, the state vector can be defined as $\mathbf{s} = [x, y, \theta]$.

Sensor measurements: Localization algorithms utilize sensor data, such as range finders, GPS, or cameras, to gather information about the robot's surroundings. These measurements are denoted by \mathbf{z} .

Motion model: The robot's motion model describes how its state evolves. It incorporates the robot's control inputs, such as wheel velocities or motor commands. The motion model is often represented as a function that predicts the robot's next state based on its current state and control inputs. It can be denoted as $\mathbf{s}' = f(\mathbf{s}, \mathbf{u})$, where \mathbf{s} is the current state, \mathbf{u} is the control input, and \mathbf{s}' represents the predicted next state.

Localization algorithm: Localization algorithms utilize the sensor measurements and the motion model to estimate the robot's pose. Popular algorithms include the Kalman filter, particle filter, or graph-based methods.

Uncertainty estimation: Localization algorithms also estimate the uncertainty associated with the robot's pose estimation. This is often represented by a covariance matrix, denoted by \mathbf{P} , which describes the uncertainty in each element of the state vector. The diagonal elements of \mathbf{P} represent the variances of the respective state variables, while the off-diagonal elements represent the covariances between different state variables.

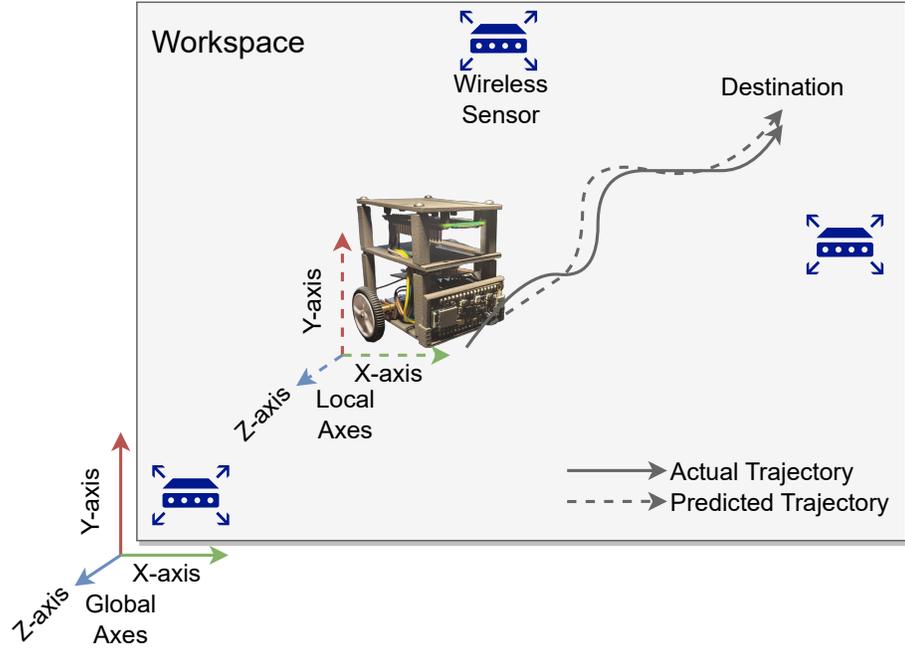


Figure 2.2: Robot localizing in a global workspace using wireless sensor nodes.

By combining sensor measurements with a motion model and applying localization algorithms, a robot can determine its position and orientation relative to a known reference frame, enabling it to navigate and interact with its environment effectively. Fig. 2.2 has shown the robot navigating in a global workspace and connected to a wireless sensor network to attempt to localize and predict the trajectory using localization algorithms.

2.2.1 Wireless Sensor Network

Wireless Sensor Networks (WSNs) are a collection of spatially distributed sensor nodes \mathbf{N} that collaborate to monitor and collect data from the environment wirelessly. Each sensor node \mathbf{i} can be represented by its state vector \mathbf{s}_i , consisting of attributes such as its location $\mathbf{p}_i = (x_i, y_i)$ and sensor measurements \mathbf{m}_i . These sensor nodes communicate with each other through wireless connections, forming a communication graph $G = (V, E)$, where V represents the set of sensor nodes and E represents the set of communication links.

In WSNs, sensor nodes are equipped with sensors to gather data from the environment. The collected data \mathbf{m}_i from multiple nodes can be aggregated to form a global view. To transmit this data efficiently, routing protocols are

employed. A routing function or algorithm $f : V \rightarrow V$ determines the next hop or forwarding node for each sensor node, ensuring the data reaches a designated sink or base station for further processing. Fig. 2.2 has shown the wireless sensor network to facilitate the robot for localization.

By leveraging network representation, communication graphs, sensing and data collection, routing and data dissemination, and energy management techniques, wireless sensor networks enable distributed monitoring and data collection from the environment. They find applications in diverse domains such as environmental monitoring, surveillance, and industrial automation.

2.2.2 Direction of Arrival

The direction of Arrival (DoA) in robotic localization refers to determining the direction from which a signal or sound arrives at a robot's sensor array. Mathematically, the DoA can be estimated by analyzing the signal's gradient at different sensors in the array. Let $\mathbf{S} = [s_1, s_2, \dots, s_N]$ represent the vector of signal strengths at N sensors. The DoA can be estimated using mathematical equations by exploiting the RSSI gradient formulation. For example, in an RSSI array, the DoA estimation can be based on the slope of the gradient of the signals approach. This can be represented by $DOA = \arctan(\frac{g_y}{g_x})$, where g_y denotes the gradient in the y-axis and g_x represents the gradient in the x-axis. The estimated DoA provides valuable information about the angle or bearing of the source concerning a reference direction, enabling robots to localize and track sound sources accurately.

2.2.3 Particle Filter

The particle filter approach is a widely used method for localization in robotics, particularly in scenarios where the environment is uncertain and the robot's motion and sensor models are non-linear. Mathematically, the particle filter represents the robot's belief about its state using a set of particles. Each particle $\mathbf{x}_t^{[i]}$ represents a possible state hypothesis at time step t , and is associated with a weight $w_t^{[i]}$ indicating the likelihood of that particle being the true state. The particles are propagated through time by incorporating the robot's control inputs \mathbf{u}_t and the motion model $p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t)$, while considering the uncertainties associated with the motion.

The particle filter's localization process involves two main steps: prediction and update. In the prediction step, each particle is updated using the motion model to generate a new hypothesis for the robot's state. This can be represented by $\mathbf{x}_t^{[i]} \sim p(\mathbf{x}_t | \mathbf{x}_{t-1}^{[i]}, \mathbf{u}_t)$, where \sim indicates sampling from the mo-

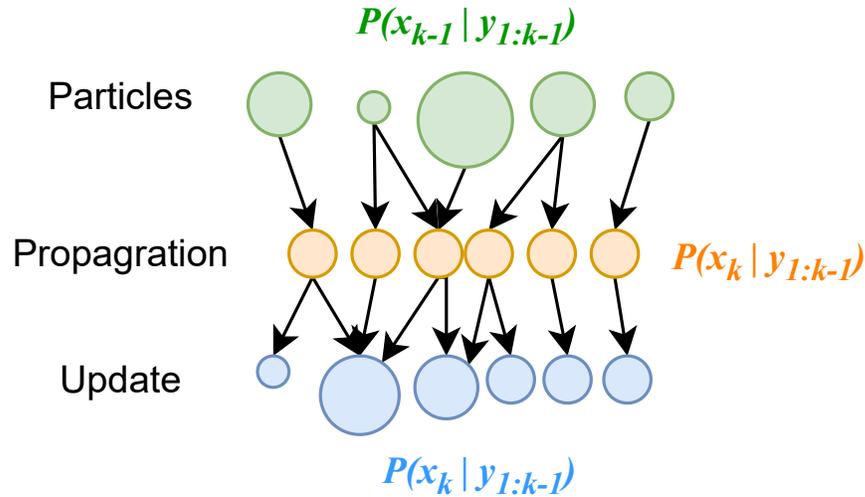


Figure 2.3: Particle filters being propagated and updated

tion model distribution. In the update step, the particles are re-weighted based on the sensor measurements \mathbf{z}_t and the observation model $p(\mathbf{z}_t | \mathbf{x}t^{[i]})$. The particle weights are computed using Bayes' rule as $w_t^{[i]} \propto w_{t-1}^{[i]} \cdot p(\mathbf{z}_t | \mathbf{x}t^{[i]})$, where $w_{t-1}^{[i]}$ is the previous weight. Finally, the particles are resampled according to their weights to create a new set of particles that better represents the posterior distribution of the robot's state. This process allows the particle filter to approximate the robot's state estimate and provide an effective localization solution even in complex and non-linear environments. Fig. 2.3 delineated the process of particle filters.

2.2.4 Expectation Maximization

The expectation-maximization (EM) approach is a widely used method for localization in robotics, particularly in scenarios where the robot has access to partial or noisy measurements of its environment. Mathematically, the EM algorithm aims to estimate the robot's state by iteratively maximizing the likelihood of the observed data. It consists of two steps: the expectation (E) step and the maximization (M) step.

In the E step, the algorithm computes the posterior distribution of the robot's state given the measurements and the current estimate of the state. Let \mathbf{x} represent the robot's state, \mathbf{z} denote the measurements, and θ represent the parameters of the model. The posterior distribution is calculated using Bayes' rule

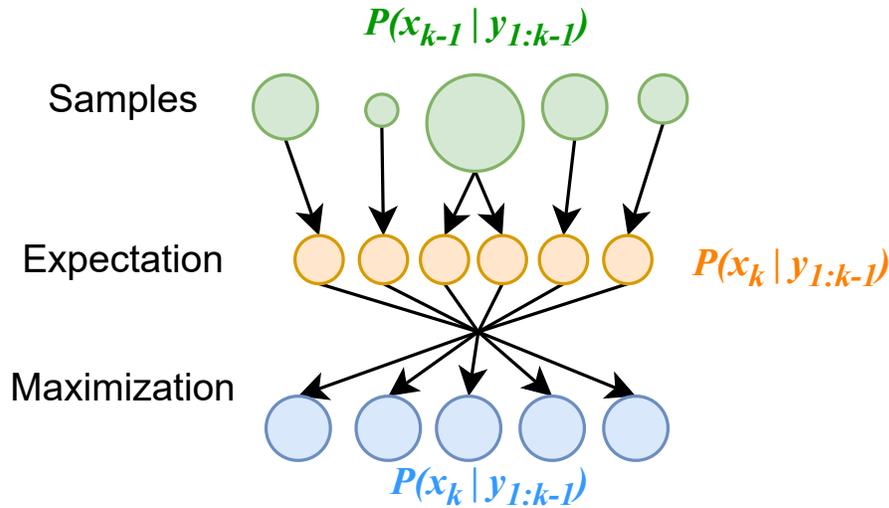


Figure 2.4: Samples expectation and maximization process

as $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}^{(k)})$, where $\boldsymbol{\theta}^{(k)}$ denotes the current estimate of the parameters at iteration k . This step involves computing the expected value of the log-likelihood function with respect to the posterior distribution.

In the M step, the algorithm updates the estimate of the state by maximizing the expected log-likelihood obtained in the E step. This involves finding the new parameter estimate $\boldsymbol{\theta}^{(k+1)}$ that maximizes the expected log-likelihood. The maximization step can be formulated as:

$$\boldsymbol{\theta}^{(k+1)} = \arg \max_{\boldsymbol{\theta}} \mathbf{E} \left[\log p(\mathbf{z}, \mathbf{x}|\boldsymbol{\theta}) | \mathbf{z}, \boldsymbol{\theta}^{(k)} \right] \quad (2.1)$$

This process is iterated until convergence, refining the estimate of the robot's state and the model's parameters. Fig. 2.4 delineated the process of particle filters.

The EM approach is powerful for localization problems as it handles uncertainty and noise effectively, allowing the robot to estimate its state accurately even in challenging environments. It is widely applied in various localization techniques, such as simultaneous localization and mapping (SLAM), where it aids in estimating the robot's pose and building a map of the environment.

2.2.5 Bayesian Information Fusion

The Bayesian probability model is a fundamental framework for information fusion in robotics, enabling the integration of multiple sources of information to make informed decisions. Mathematically, it utilizes Bayes' theorem to update the probability distribution over a hypothesis given new evidence. Let H denote the hypothesis and \mathbf{E} represent the evidence obtained from different sources. The Bayesian probability model calculates the posterior probability $P(H|\mathbf{E})$ using the prior probability $P(H)$ and the likelihood $P(\mathbf{E}|H)$, as given by Bayes' theorem:

$$P(H | E) = \frac{P(H)P(E | H)}{P(E)} \quad (2.2)$$

Here, $P(H)$ represents the prior probability, which represents the belief about the hypothesis before considering the evidence. The likelihood $P(\mathbf{E}|H)$ quantifies the probability of obtaining the evidence given the hypothesis. The denominator $P(\mathbf{E})$ is the normalizing constant, ensuring that the posterior probabilities sum up to one. The Bayesian probability model allows for the incorporation of prior knowledge, updating beliefs based on new evidence, and iterative refinement of hypotheses as additional information becomes available.

By applying the Bayesian probability model, robotic systems can effectively fuse information from different sensors, algorithms, or data sources to make more accurate and reliable decisions. The model provides a principled framework for information fusion, allowing robots to reason under uncertainty and incorporate diverse sources of information into a coherent and consistent representation of the environment.

2.2.6 Graph Optimization

Graph optimization is a powerful technique used in robotics for solving problems such as localization, mapping, and trajectory optimization. It involves formulating the problem as a graph, where nodes represent variables and edges represent constraints or relationships between variables. Mathematically, a graph optimization problem can be represented as minimizing or maximizing an objective function $\mathcal{F}(\mathbf{x})$ subject to constraints. Let \mathbf{x} represent the vector of variables associated with the nodes in the graph. The objective function quantifies the optimization goal, which can be expressed as $\mathcal{F}(\mathbf{x}) = \sum_i f_i(\mathbf{x}_i)$, where $f_i(\mathbf{x}_i)$ represents a cost or objective associated with node i . Constraints are typically modeled as penalty terms or equality/inequality constraints that enforce relationships between variables, such as $\mathbf{g}(\mathbf{x}) = \mathbf{0}$ or $\mathbf{h}(\mathbf{x}) \leq \mathbf{0}$.

Graph optimization problems are solved using optimization algorithms like the Gauss-Newton method or the Levenberg-Marquardt algorithm. These algorithms iteratively update the variables \mathbf{x} to minimize or maximize the objective function while satisfying the constraints. The solution to the graph optimization problem provides the optimal values of the variables, yielding an optimized solution that satisfies the given constraints and minimizes the objective function.

By leveraging the graph optimization framework, robotic systems can solve complex problems by formulating them as graphs and optimizing the variables based on objective functions and constraints. This approach allows robots to efficiently solve localization problems, perform simultaneous localization and mapping (SLAM), optimize trajectories, or solve other optimization tasks, enabling them to operate effectively in dynamic and uncertain environments.

2.2.7 Gaussian process Model Regression

Gaussian Process Model Regression (GPR) is a powerful probabilistic regression method used in machine learning and robotics to model and predict functions based on observed data. It is based on the assumption that the underlying function follows a Gaussian process, which is a collection of random variables where any finite subset follows a joint Gaussian distribution. Mathematically, GPR models the function as a Gaussian process prior and then conditions it on the observed data to obtain a posterior distribution over functions.

Let \mathbf{X} represent the input data matrix and \mathbf{y} denote the corresponding output or target values. GPR aims to estimate the underlying function $f(\mathbf{X})$ by learning the mean function $\mu(\mathbf{X})$ and the covariance function $k(\mathbf{X}, \mathbf{X}')$. The mean function represents the expected value of the function at each input point, while the covariance function captures the similarity or correlation between different input pairs. The GPR model assumes that the observed data \mathbf{y} is related to the function values through a noise term ϵ , following a Gaussian distribution with zero mean and covariance $\sigma_n^2 \mathbf{I}$.

The GPR posterior distribution can be computed using Bayes' rule, resulting in a predictive distribution that provides a probabilistic estimate of the function values for new input points. The predictive distribution is a Gaussian distribution with a mean vector and a covariance matrix, given by $\mathbf{m}(\mathbf{X}_{\text{new}})$ and $\mathbf{K}(\mathbf{X}_{\text{new}}, \mathbf{X}_{\text{new}})$, respectively. The mean vector represents the predicted mean values for the new inputs, and the covariance matrix captures the uncertainty or confidence in the predictions. The GPR framework allows for efficient and flexible regression, providing both point estimates and uncertainty quantification, which is valuable for decision-making and robotic applications.

By leveraging the Gaussian Process Model Regression, robots can effectively model and predict functions based on observed data, allowing them to make informed decisions and adapt to changing environments. GPR finds applications in various tasks, such as trajectory planning, object tracking, and sensor fusion, where accurate and uncertainty-aware predictions are crucial for robotic systems.

2.3 Exploration

Exploration in multi-robotic systems refers to the coordinated exploration of an unknown environment by a group of robots. It involves the efficient allocation of robots to unexplored regions to gather information and create a comprehensive map of the environment. Mathematically, the exploration problem can be formulated as an optimization task, where the objective is to maximize the coverage of the environment while minimizing the time and energy expended. Let N denote the total number of robots in the system, and \mathbf{s}_i represent the state vector of robot i , which includes its position and orientation. The exploration process aims to determine the next best action for each robot, such as moving to a specific location or exploring a particular area, denoted as \mathbf{a}_i . This action selection is typically based on a utility function, which quantifies the value or information gain associated with different exploration options. The exploration problem can be represented as finding the optimal action assignment $\mathbf{A} = \mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N$ that maximizes the overall utility or coverage.

The exploration in multi-robotic systems requires coordination mechanisms to ensure efficient coverage and avoid redundant exploration. This coordination can be achieved through communication and task allocation strategies. Communication enables robots to share their exploration progress and coordinate their actions. Task allocation algorithms allocate different regions or tasks to individual robots based on their capabilities and exploration objectives. Let T represent the set of tasks or regions to be explored, and $f : V \rightarrow T$ denote the task allocation function that assigns each robot to a specific task. The task allocation process optimizes the assignment based on various criteria, such as minimizing the total exploration time or maximizing the coverage efficiency. Fig. 2.5 has shown the exploration sequence to explore unknown space with multi-robotic cooperation

By leveraging exploration strategies in multi-robotic systems, robots can efficiently and collaboratively explore unknown environments, enabling applications such as search and rescue missions, environmental monitoring, or mapping of hazardous areas. The mathematical formulation and coordination

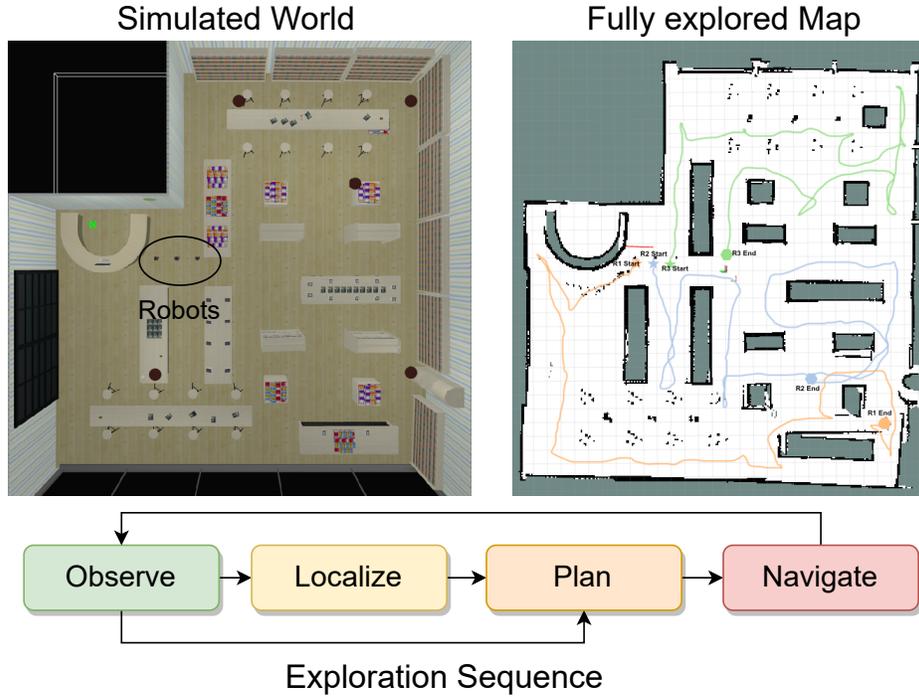


Figure 2.5: Simulated world with explored map and exploration sequence

mechanisms provide a framework for optimizing the exploration process, ensuring effective coverage and utilization of resources in multi-robotic exploration scenarios.

2.3.1 Reinforcement Learning

Reinforcement learning (RL) is a machine learning approach that focuses on learning optimal decision-making policies through interactions with an environment. It is typically modeled as a Markov Decision Process (MDP), defined by a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma \rangle$, where \mathcal{S} represents the set of states, \mathcal{A} denotes the set of actions, \mathcal{R} is the reward function, \mathcal{P} represents the transition probability function, and γ is the discount factor. RL aims to learn an optimal policy $\pi(\mathbf{s})$ that maximizes the expected cumulative reward over time. The policy is a mapping from states to actions, and the goal is to find $\pi^*(\mathbf{s})$ that solves the RL problem.

The RL learning process involves an agent interacting with the environment over discrete time steps. At each time step t , the agent observes the current state \mathbf{s}_t , selects an action \mathbf{a}_t according to its policy $\pi(\mathbf{s}_t)$, and receives a

reward signal r_t and the next state $\mathbf{s}t + 1$. The agent’s objective is to update its policy based on this experience to maximize the expected future cumulative reward. This is typically done using value-based or policy-based RL algorithms. In value-based methods, the agent estimates the value function $V^\pi(\mathbf{s})$ or the action-value function $Q^\pi(\mathbf{s}, \mathbf{a})$ to evaluate the goodness of states or state-action pairs, respectively. In policy-based methods, the agent directly parameterizes the policy $\pi(\mathbf{a}|\mathbf{s}; \theta)$ and updates the policy parameters θ using gradient ascent to maximize the expected reward.

By leveraging reinforcement learning, robots can learn optimal decision-making policies in dynamic and uncertain environments, enabling them to autonomously adapt and improve their behavior over time. The mathematical framework and algorithms of RL provide a principled approach to tackling sequential decision-making problems, making it a valuable tool for various robotic applications, including navigation, manipulation, and task execution.

Q-Learning

Q-learning is a popular reinforcement learning algorithm that enables agents to learn optimal action-selection policies in Markov Decision Processes (MDPs). It is based on the principle of iteratively updating the Q-value function, which estimates the expected cumulative reward for taking a particular action in a given state. Mathematically, the Q-value for a state-action pair (\mathbf{s}, \mathbf{a}) is denoted as $Q(\mathbf{s}, \mathbf{a})$, and it represents the expected cumulative reward when following a particular policy from that stage onwards. Q-learning updates the Q-values using the Bellman equation, given by:

$$Q(\mathbf{s}, \mathbf{a}) \leftarrow Q(\mathbf{s}, \mathbf{a}) + \alpha \left(r + \gamma \max_{\mathbf{a}'} Q(\mathbf{s}', \mathbf{a}') - Q(\mathbf{s}, \mathbf{a}) \right) \quad (2.3)$$

where α is the learning rate, r represents the immediate reward received upon taking action \mathbf{a} in-state \mathbf{s} , γ is the discount factor that balances immediate and future rewards, \mathbf{s}' is the resulting state after taking action \mathbf{a} , and $\max_{\mathbf{a}'} Q(\mathbf{s}', \mathbf{a}')$ represents the maximum Q-value for the next state.

The Q-learning algorithm iteratively updates the Q-values for state-action pairs based on interactions with the environment. It explores the environment by selecting actions based on an exploration-exploitation trade-off, gradually converging into an optimal policy that maximizes the expected cumulative reward. Through exploration and exploitation, Q-learning enables agents to learn effective action-selection strategies in MDPs, allowing them to solve complex decision-making problems. Its simplicity and effectiveness have made Q-learning widely used in various domains, including robotics, where agents can

autonomously learn and adapt their behavior in dynamic and uncertain environments.

2.3.2 Convex Hull

The convex hull is a fundamental concept in computational geometry representing the smallest convex polygon enclosing a given set of points. Mathematically, let \mathcal{P} be a set of N points in a two-dimensional space, represented as $\mathcal{P} = \mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N$. The convex hull, denoted as $\text{conv}(\mathcal{P})$, is the smallest convex polygon that contains all the points in \mathcal{P} . It can be defined as the intersection of all convex sets containing \mathcal{P} or as the set of all convex combinations of points in \mathcal{P} . Mathematically:

$$\text{conv}(\mathcal{P}) = \left\{ \sum_{i=1}^N \lambda_i \mathbf{p}_i \mid \sum_{i=1}^N \lambda_i = 1, \lambda_i \geq 0 \forall i \right\} \quad (2.4)$$

herein λ_i represents the weights or coefficients of the convex combination.

The convex hull has several properties, such as being convex and having its vertices defined by a subset of the original points. Efficient algorithms, such as Graham's scan or Jarvis march, exist to compute the convex hull of a set of points in computational geometry. The convex hull finds applications in various fields, including computer graphics, computational biology, and robotics. For instance, in robotics, the convex hull can be used for collision detection, path planning, or determining the reachable workspace of a robot. By representing the boundary of the set of points, the convex hull provides valuable geometric information that aids in solving a wide range of geometric and computational problems.

2.4 Simultaneous Localization and Mapping

Simultaneous Localization and Mapping (SLAM) in multi-robotic systems refers to the joint problem of building a map of an unknown environment while simultaneously estimating the poses of multiple robots within that environment. It involves integrating sensor measurements and control inputs to iteratively refine the robots' positions and construct a consistent map. Mathematically, SLAM can be represented as a recursive Bayesian filtering problem, where the robots' poses are denoted by \mathbf{x}^t , and the map of the environment is represented by \mathbf{m}^t . The SLAM problem can be expressed as estimating the posterior distribution $p(\mathbf{x}^{1:N}, \mathbf{m}^t \mid \mathbf{z}^1 : t, \mathbf{u}^1 : t)$, given the sensor measurements $\mathbf{z}^1 : t$ and control inputs $\mathbf{u}^1 : t$ up to time step t . This estimation is typically

done using recursive techniques such as Extended Kalman Filter (EKF) or Particle Filter (PF) methods.

The SLAM problem in multi-robotic systems involves coordinating and fusing information from multiple robots to create a consistent and accurate map. Communication and information sharing is crucial in multi-robot SLAM, enabling the robots to exchange their local maps or pose estimates, reducing uncertainty, and improving overall mapping and localization accuracy. The critical challenge in multi-robot SLAM lies in the efficient coordination of robots, data association, and mapping merging. By leveraging multiple robots' collective observations and knowledge, multi-robot SLAM enhances mapping capabilities, accelerates exploration, and enables collaborative localization in complex and large-scale environments.

SLAM's mathematical formulation and algorithms in multi-robotic systems provide a framework for robots to create maps and localize themselves in unknown environments autonomously. This capability finds applications in various fields, including search and rescue operations, environmental monitoring, and cooperative manipulation tasks, where accurate mapping and localization are crucial for effective and coordinated robot behavior.

2.4.1 Extended Kalman Filtering

Extended Kalman Filtering (EKF) is a popular recursive estimation technique used in SLAM to estimate the poses of robots and construct maps of unknown environments. It combines the principles of Kalman filtering with non-linear system models by linearizing them through a first-order Taylor expansion. Mathematically, EKF for SLAM involves maintaining the belief of the robot's state and the map as a joint Gaussian distribution. Let $\mathbf{x}_t^{[i]}$ represent the state vector of robot i at time step t and \mathbf{m}_t denote the map of the environment. The SLAM estimation can be represented by the belief distribution $p(\mathbf{x}_t^{[1:N]}, \mathbf{m}_t | \mathbf{z}_1 : t, \mathbf{u}_1 : t)$. The EKF for SLAM iteratively applies prediction and update steps. The prediction step incorporates the motion model $p(\mathbf{x}_t^{[i]} | \mathbf{x}_{t-1}^{[i]}, \mathbf{u}_t^{[i]})$ to propagate the robot's state estimate. In contrast, the update step integrates the sensor measurements $\mathbf{z}_t^{[i]}$ and the observation model $p(\mathbf{z}_t^{[i]} | \mathbf{x}_t^{[i]}, \mathbf{m}_t)$ to refine the estimates based on new information. The linearization of non-linear models in the EKF enables recursive estimation of the state and map, providing a computationally efficient approach for SLAM in a wide range of robotic applications.

The EKF approach in SLAM enables robots to estimate their poses and construct maps in real-time by effectively combining motion and sensor measurements. Its mathematical formulation and recursive nature make it a widely

adopted technique for SLAM tasks, allowing robots to operate in unknown environments, localize themselves, and create accurate maps of their surroundings.

The above-discussed concepts enable you to understand the terminologies and techniques used in the dissertation.

CHAPTER 3

INSTANTANEOUS WIRELESS ROBOTIC NODE LOCALIZATION USING COLLABORATIVE DIRECTION OF ARRIVAL

3.1 Introduction

In this chapter, we will discuss a single robot localization solution using a collaboration of wireless sensor nodes and direction of arrival. A set of sensors, actuators, and mobile devices are connected to form an Internet of Things (IoT) system. Here, location information is critical for such operations, especially for wireless and mobile robotic nodes. Node localization has been a challenging problem, especially in indoor environments. As such, Indoor localization has emerged as one of the most critical components in robotics, automation, and wireless systems. Here, one fundamental requirement is to provide an accurate and efficient localization system in a real-time (online) manner. Furthermore, GPS-denied or dynamically changing environments pose additional challenges for mobile robot indoor localization (Zafari et al., [2019a](#)).

Sensors such as cameras, LIDAR, inertial measurement units (IMU), and their fusion have been exploited for obtaining accurate indoor localization of mobile devices (Canedo-Rodriguez et al., [2016](#)). However, these technologies are expensive, non-applicable to resource-constrained devices and robots, and also suffer from various limitations, such as the requirement of proper lighting

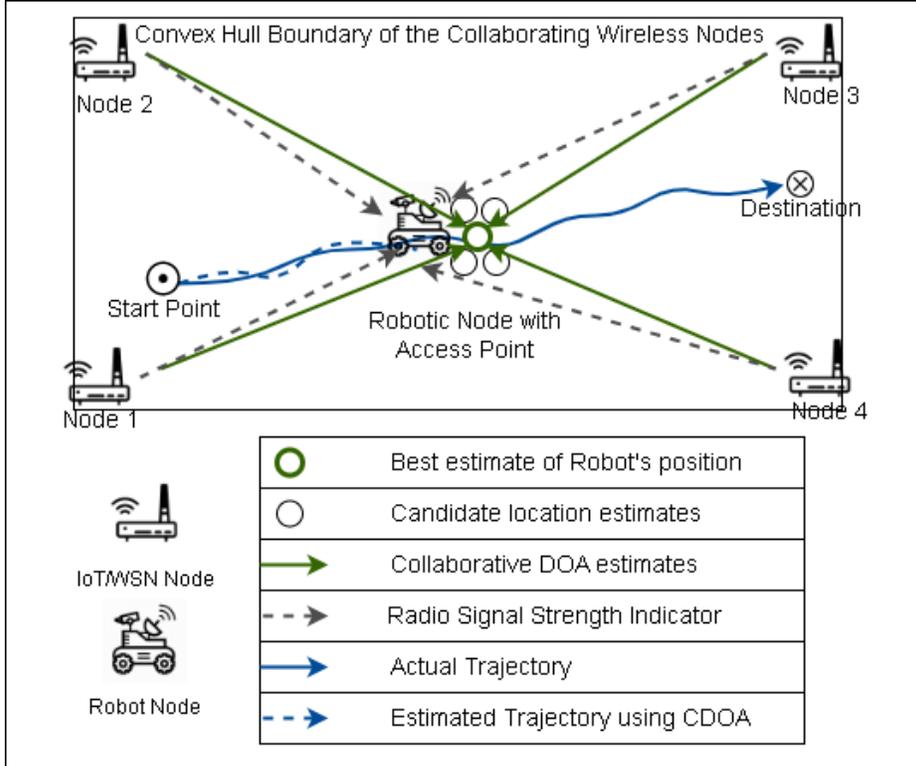


Figure 3.1: Overview of the proposed real-time CDOA-based localization of a robotic node following a trajectory with limited IoT/Wireless Nodes.

conditions in vision-based localization and structured non-dynamic surfaces for LIDAR-based perception.

On the other hand, wireless technologies such as Wi-Fi and Bluetooth are the most extensively utilized for indoor WLANs. The ubiquitous availability of Received Signal Strength Indicator (RSSI) measurements from Access Points (AP) or Wireless Sensor Nodes (WSNs) can be used for various objectives, including localization (Latif & Parasuraman, 2022a; Motroni et al., 2021; Rizzo et al., 2021), multi-robot control (S. Luo et al., 2019; Parasuraman & Min, 2019), and communication optimization (Parasuraman et al., 2018). These advances provide opportunities to exploit the RSSI information from WSNs in aiding mobile robot localization.

Extant RSSI-based indoor positioning systems frameworks require an offline site survey to generate fingerprints and match the current real-time RSSI data to this database for positioning with a supervised machine learning algorithm (e.g., (Sadowski et al., 2020)). However, the fingerprinting approaches

require a dedicated offline phase, in addition to the limitation of generalization, where they can be employed only for the specific environment where they are fingerprinted (Tao & Zhao, 2018). However, for mobile robot deployments, these limitations are not practical.

To address these gaps, we propose a novel algorithm for estimating the Collaborative Direction of Arrival (CDOA) estimated with the RSSI values obtained through collaboration among WSNs. The CDOA estimation is then integrated with two Bayesian framework variants for robust node localization: Expectation Maximization (EM) and Particle Filter (PF). See Fig. 3.1 for an overview of the proposed WSN collaboration-based wireless robotic node localization method.

The main contributions of this chapter are outlined below.

1. We propose a novel collaboration-aided mechanism for a mobile robot to collect RSSI data from the WSNs and estimate Wireless signal CDOA.
2. We integrate the CDOA with a Bayesian framework for robot node localization. We propose two variants (Expectation Maximization and Particle Filter) to exploit the statistical EM method's accuracy and efficiency advantage the sampling-based PF method offers.
3. We theoretically analyze the properties of the proposed CDOA localization method in terms of localization consistency, accuracy, area coverage, scalability, and computational complexity.
4. Through extensive experimental analysis in diverse setups enabled by numerical simulations, publicly available real-world datasets, and in-house robot hardware demonstrations, we evaluate the localization accuracy and efficiency of the proposed variants of CDOA-aided node localization.
5. We validate our approach by comparing with relevant non-fingerprinting methods from the recent literature such as the trilateration (B. Yang et al., 2020a), weighted centroid localization (Z.-M. Wang & Zheng, 2014), differential RSSI (Podevijn et al., 2018), improved RSSI-based localization (W. Xue et al., 2017), and the smooth Particle Filter with Extended Kalman Filter (Zafari et al., 2018) approaches.
6. We open source all our codes and datasets (native Python implementations for the IoT and wireless sensor network community, as well as a ROS (Quigley et al., 2009) package for the robotics community) at <https://github.com/herolab-uga/cdoa-localization>. We believe this will

enable the reproducibility and extension of our approach by the research community.

The core novelty of our approach lies in that we employ a CDOA metric obtained through cooperative communication between the WSNs and the mobile robot instead of relying on the RSSI metric directly (as used in relevant methods in the literature). Further, we integrate and extensively evaluate the CDOA metric with Bayesian frameworks for robot node localization. Our proposed methods achieve superior accuracy, efficiency, and robust localization performance through these novelties while enabling real-time efficiency compared to several state-of-the-art solutions.

3.2 Related Work

According to a recent survey on Wi-Fi-based indoor positioning, (F. Liu et al., 2020a), there are two categories of wireless localization solutions: *Model-based and Survey-based*. Model-based approaches include trilateration using RSSI, triangulation using DOA, and Weighted Centroid using distance. Recent works include variants thereof, such as the filtered trilateration (B. Yang et al., 2020a), differential RSSI-based least squares estimation (Podevijn et al., 2018), and Expectation Maximization (Pajovic et al., 2015b). While the survey-based approaches provide high accuracy based on the precise fingerprints collected from the same environment through a dedicated offline process, they also come with high computation costs of prediction algorithms like the K-Nearest Neighbors (Subedi & Pyun, 2020). Accordingly, we focus on model-based solutions.

In model-based approaches, multilateration and triangulation are the fundamental methods to predict the position of a wireless device (e.g., a mobile robot) using RSSI captured from multiple anchors/APs (Passafiume et al., 2016). However, these methods suffer from co-linearity, ambiguous positioning, non-intersecting circles, etc. A recent survey (Zafari et al., 2019b) on model-based techniques confirms that the balance of accuracy and computing complexity is absent in the literature. Also, different variants of trilateration/triangulation can have different localization accuracy, resulting in inconsistency in its application. The weighted centroid method is less accurate for non-line-of-sight conditions, limiting its applicability. Further, some methods convert the raw RSSI measurements into distance estimates to use in multilateration algorithms, which suffer from the dependency on wireless channel parameters of the environment for RSSI to distance conversion.

A novel trilateration algorithm is put forth by Yang et al. (B. Yang et al., 2020b) for RSSI-based indoor localization. This technique determines the lo-

cation of a target by using the geometric relationships between transmitters and receivers. The precision of localization can be considerably impacted by environmental changes, such as obstructions and multipath propagation, because trilateration is sensitive to these. This approach also needs considerable calibration and exact distance estimation, which might be difficult in dynamic situations. A Gaussian Filtered RSSI-based Indoor Localization method utilizing Bootstrap filtering is presented by Wang et al. (J. Wang et al., 2021). This technique uses particle filtering to determine a target's location within a WLAN. Despite being more resistant to environmental noise, this method still relies on RSSI measurements, which are prone to interference, multipath propagation, and signal attenuation.

Pinto et al. (Pinto et al., 2021) use K-means clustering and Bayesian estimation to create a reliable RSSI-based indoor positioning system. This method seeks to increase localization accuracy by integrating unsupervised learning with probabilistic estimates. It is nonetheless susceptible to the drawbacks of RSSI measurements, such as susceptibility to environmental changes and the requirement for significant calibration. Bayesian filtering method also presented by Mackey et al. (Mackey et al., 2020) for enhancing BLE beacon proximity estimation accuracy. This technique improves the performance of BLE beacons, but it is still vulnerable to interference from other wireless devices and may have decreased accuracy in NLOS scenarios and dynamic surroundings. Another method for estimating the path-loss exponent using Bayesian filtering is presented by Wojcicki et al. (Wojcicki et al., 2021). This method is intended to describe signal propagation in various contexts, increasing the precision of localization. The quality of the RSSI measurements, which the surroundings and signal attenuation can impact, is crucial to the path-loss exponent estimation's accuracy.

Combining data from many sources and utilizing probabilistic techniques, CDOA, particle filtering (PF), and expectation maximization (EM) for wireless sensor networks minimize the drawbacks of the aforementioned methods. While CDOA-EM uses locations as samples to fill a grid and determine robot location using Gaussian probability, CDOA-PF increases robustness in complicated situations by repeatedly updating particles representing potential positions. Both strategies overcome the shortcomings of conventional RSSI-based techniques and offer more precise and dependable localization in the presence of NLOS, multipath propagation, and environmental changes.

As an alternative to the RSSI metric, researchers have proposed the use of the Channel State Information (CSI) metric for robot localization systems (Jadhav et al., 2021; Song et al., 2017; X. Wang et al., 2016). However, most CSI-

based techniques involve extensive offline fingerprinting processes to improve accuracy. Moreover, as with RSSI-based metrics, the CSI metric is limited to very few radios and cannot be exploited for ubiquitous applications. DOA (or Angle of Arrival) based methods achieve higher localization accuracy than RSSI-based solutions. For instance, in research (Arbula & Ljubic, 2020), the authors used a sensor node equipped with Infrared light arrays to estimate the DOA of a mobile robot, which was used to achieve indoor localization with meter-level accuracy. Cooperatively localizing target nodes using multiple reference nodes with known locations has been explored. For instance, the authors in (Hassani et al., 2015) provided a distributed method for cooperatively estimating the DOA of an acoustic sensor network. In contrast, the authors in (J. Xu et al., 2015) used cooperative DOA from Ultra Wideband (UWB) radios to locate target nodes using many reference nodes.

UWB-based indoor localization systems, while highly promising, still face several limitations. The presence of multipath propagation and non-line-of-sight (NLOS) conditions can significantly affect the positioning accuracy (Alarifi et al., 2016). Yang et al. (B. Yang et al., 2022) proposed a UWB-based indoor localization with fewer nodes and utilized deep neural networking to avoid the effect of non-line-of-site; however, this solution requires offline data training and sampling overhead, which makes the system restricted to the trained environment. Additionally, deploying UWB anchors can be challenging in real-world environments due to their need for precise installation (Ridolfi et al., 2021). Further, the power consumption of UWB devices and their susceptibility to interference from other wireless systems can negatively impact their performance and scalability (Zafari et al., 2019b). Integration of UWB-based systems with other sensing modalities can be difficult, as the fusion of data from different sources may be subject to noise and uncertainty (C. Wang, Xu, et al., 2021).

A wireless sensor network with a few WSNs can overcome the limitations of UWB-based indoor localization by leveraging cooperative sensor modalities to provide robust and accurate localization in complex environments. By fusing data from diverse sensing sources, WSNs can mitigate the effects of multipath propagation, NLOS conditions, and interference, enhancing the overall system performance (Celaya-Echarri et al., 2020). Furthermore, commercial UWB-based localization solutions provide accuracy of up to 10cm and connection stability at the cost of high computation power and expensive anchor node solution, which makes it impractical for swarm robots (Starks et al., 2023) with limited computational power and can only possess wireless connectivity. The

proposed solution provides highly scalable, computationally efficient, and high localization accuracy for small to large-scale multi-robot systems.

Estimating a mobile node’s position using only a few reference nodes with high accuracy and efficiency is achievable in wireless sensor networks. Wang et al. (H. Wang et al., 2019) proposed sparse Bayesian learning for robust DOA estimation with only a few base station nodes. But, their implementation assumed multiple antennas at each base station, realizing an EM-based DOA estimation and eventual vehicle localization using the DOA triangulation. Wang’s proposed solution is computationally expensive and requires high-end base stations, which makes it impractical for small robots operating indoor environment. On the contrary, in our work, we assume typical Wi-Fi sensors without having access to multi-antenna data, allowing ubiquitous integration with existing wireless sensor networks/IoT systems and computationally efficient online CDOA-based indoor localization.

Therefore, we propose a CDOA estimation using IoT or wireless nodes and fuse it with Bayesian approaches for high-accuracy localization of mobile robotic nodes. **We depart from the literature in two different ways: 1) we use a collaborative mechanism between the WSNs to obtain the CDOA of wireless signals; 2) we estimate Gaussian probability on the CDOA estimates, adopting EM and PF Bayesian frameworks.** The localization system can be applied independently of the robot’s motion model or combined with the robot’s odometry, if available, to improve accuracy. Moreover, our approach uses only a few reference nodes and works on resource-constrained robotic nodes in real time. Our proposed approach is advantageous by reducing computational complexity without embedding external hardware and using bearing-only information (aided by the cooperative RSSI measurements). It achieves high accuracy even in the presence of signal noise. This way, our method balances the efficiency and accuracy of quick online operation without fingerprinting dependence. While localization of static Access Points has been demonstrated using DOA (Parashar & Parasuraman, 2020), this is the first work that uses CDOA for robotic node localization demonstrated in real-world implementations.

3.3 Problem Statement

We look at the problem of a robot node localizing itself against its surroundings. Here, a limited (smaller) number of WSNs or IoT nodes are distributed in the environment, and the mobile robot is mounted with an AP, which nearby WSNs can sense. The robot can operate within the sensing range of WSNs,

which is assumed to be 40m; the robot is not restricted to be in the boundary of WSNs but confined to be within the range (e.g., it could be outside the convex of boundary).

WSNs are assumed to be static, and their exact position is known to the robot for gradient calculation in the global frame. Furthermore, the robot is restricted to moving within the sensing range of connected WSNs. The WSNs measure the RSSI values coming from the AP and communicate this information to the robot cooperatively (assuming the measurements and shared data are reasonably time-synchronized using NTP-like protocols). Robot will use RSSI values to calculate the gradient and convert it into the CDOA with respect to the position of WSNs. The robot R keeps track of the trajectory along with the CDOA measurements as the tuple: $m_l = \{x_l, y_l, CDOA_l\}$, where (x_l, y_l) is the location of the robot at location l .

The objective is to find the best estimate of the robot's location (x_l^*, y_l^*) , which maximizes the probability of observing the measurement tuples when the robot is at the estimated location $P(x_l, y_l \mid m_l, m_{l-1}, \dots, m_{l-M})$, where m_l is the sample for position l and M is the number of previous samples considered along the completed trajectory so far, given that we employ an arbitrary method to estimate the CDOA. Table 3.1 lists the key symbols and notations used in the chapter.

3.4 Proposed CDOA Approach

Cooperative localization can be accomplished with a network of wireless nodes, where each node can sense the signal strength of the other node in the network. Our approach consists of two units: 1) we propose a CDOA estimation scheme from RSSI measurements with an assumption of the geometric model of the AP/WSN distribution in the environment, which is typical in the literature; 2) our solution deploys EM and PF-based localization of a mobile robot node using the CDOA.

In principle, we need at least three WSNs that form at least two noncollinear segments between them to measure a valid gradient inside the boundary created by the WSNs ((Twigg et al., 2012)). Having a higher number of WSNs will increase the robustness of the solution. It is possible to extend this setup where a mesh network is available with several known and unknown wireless nodes, but we limit our scope and experiments in this chapter with four WSNs deployed at the corners of a robotic node workspace boundary (see Fig. 3.1).

The proposed method has a basis in WSNs collaboration to measure RSSI collaboratively and calculate CDOA. Alg. 1 provides an algorithmic pseudo-

Table 3.1: Notations and their descriptions

Notation	Description
S_i	Measured RSSI at a wireless node N_i
$\vec{g} = [g_x, g_y]$	Gradient of the RSSI's signal strength
(x_c, y_c)	Centroid position of a rectangular workspace
Δ_X, Δ_Y	Distance between the WSNs along the x and y axes
$CDOA_l$	CDOA of the signal at a position along the robot's path l
P_r, R_r	Initial and re-sampled list of particles in PF
ω	Weights associated with each sample
x_t	State at time t
E_r	List of grid positions in the EM algorithm
$w_{E_r}(x_t)$	Gaussian Probability for each grid position in E_r at time t
M	Number of previous samples considered
σ	Standard deviation of error for all previous samples
err_l^k	CDOA error for sample k
w_i	Weight of each particle
$w_i^*(q_i)$	Normalized weight of each particle

code of the CDOA estimation of the mobile robot using the surrounding WSNs. The first part of the Alg. 1 lays out the wireless network collaboration for determining CDOA at the robot. All these computations are performed by the moving AP-mounted robot, which runs a centralized service and receives the RSSI information from all connected nodes that sense wireless signals independently in a synchronized manner.

For collaborative measurement of RSSI on WSNs connected to an access point installed on a robot, time synchronization is essential. These nodes can synchronize their clocks within the *time window* using the Network Time Protocol (NTP), ensuring precise and consistent RSSI readings throughout the whole network (Ranganathan & Nygard, 2010). Using the time window idea, sensor nodes can send and receive RSSI data at predetermined intervals, effectively controlling communication and minimizing the possibility of collisions while enabling the system to coherently process and evaluate the obtained data (W. Fang et al., 2021).

For the algorithm to work, we need at least three spatially-distributed WSNs in the network, and the robot needs to be within the polygonal boundary of the WSNs.

RSSI can be modeled as a vector with two components, and the gradient concerning the center of the robot can be represented as $\vec{g} = [g_x, g_y]$. One of the primary advantages of the central finite difference method is that it provides gradient estimation based on the received signal strength from geometrically oriented wireless nodes. After an appropriate gradient estimation, a receiver node (moving robot) can estimate the direction of arrival of signals based on the reference position for position estimation using EM and PF contrivance.

In our current implementation, the CDOA of the mobile robot within the network is obtained from the geometric rule described in the central finite difference method (Parasuraman, Fabry, et al., 2013). For a rectangular configured networked infrastructure with centroid position (x_c, y_c) , refer to Fig. 3.1 where the RSSI value of Node 1 is S_1 and so on, the RSSI gradient is calculated as:

$$g_x = \frac{S_3 - S_2}{2\Delta_X} + \frac{S_4 - S_1}{2\Delta_X}; g_y = \frac{S_2 - S_1}{2\Delta_Y} + \frac{S_3 - S_4}{2\Delta_Y} \quad (3.1)$$

Here, Δ_X is the distance between the wireless sensor's antennas along the x-axis, and Δ_Y is the distance between the wireless sensor's antennas along the y-axis. We then calculate the CDOA from the gradients calculated using Eq. (3.1).

$$CDOA_l = \arctan\left(\frac{g_y}{g_x}\right) \quad (3.2)$$

The formula provides the CDOA of the wireless signal at a position along the path l using the RSS gradient. We can then suppress the noise of the calculated CDOA by using the exponentially weighted moving average.

We employ a Gaussian probability model on the wireless signal CDOA estimates to calculate the weights of each random particle in the PF. Similar to the work in (X. Li et al., 2015) that uses acoustic signals, this probabilistic model will weigh the quality of signals sensed by each node from N and ultimately produce an accurate robot location estimate through the PF.

The absolute error between Actual CDOA for all wireless sensors at a potential candidate position l of mobile robot with coordinates (x_c, y_c) to the perceived CDOA values for each sensor calculated for each particle. Later, we use the Gaussian probability formula (similar to (Parashar & Parasuraman, 2020)) on this error to calculate the probability of the i_{th} candidate location of the particle $q_i = (x_i, y_i, w_i), i \in (1, ..n)$, where n is the number of samples in the PF spread in the bounded region with the resolution of \mathfrak{R} and w_i is the weight

of each particle calculated over a set of previous path samples as:

$$P_l(q_i) = \prod_{k=0}^{M-1} \left[\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(err_l^k)^2}{2\sigma^2}} \right], \quad (3.3)$$

where $P_l(q_i)$ is the probability density function for the position l of candidate particle i in PF, M is the number of the previous samples considered, σ is the standard deviation of error for all previous samples, and $(err_l^k)^2$ is the CDOA error for sample k . Eq. (3.3) provide the probability for particle q_i considering $M - 1$ previous samples.

There is an intrinsic angular inaccuracy in each CDOA degree that is analyzed. σ represents the fluctuation (deviation) of this error, which is anticipated to be known because we know the correctness of the technique used to assess the CDOA. We use the product of the Gauss likelihood of CDOA error over $M - 1$ prior robot positions (imitating geographically scattered samples) so that the sifted CDOA from earlier path locations can be used similarly as readings from many sensors.

Next, we discuss how the CDOA estimation is integrated with a probabilistic framework to achieve localization using the DOA information.

3.4.1 CDOA-PF

The CDOA probability from Eq. 3.3 is used to calculate the weights of the particles in the PF, which is then employed in the resampling procedure in the next PF step (iteration).

$$w_i \propto P_l(q_i) \quad (3.4)$$

The EM and PF provide initial hypotheses with a uniform sampling of probable robot locations across the environment using a constraint around the present robot location. The Gaussian probability is determined for each particle, the signal source. The particles are subsequently given weights that are proportional to their likelihood, and the weights w_i are normalized as $w_i^*(q_i) = \frac{w_i(q_i)}{\sum_{i=0}^{n-1} w_i(q_i)}$.

This normalized weight determines the likelihood of regenerating a particle in the next iteration. The particle with the highest weight (softmax) best gauges the robot's location. This process is repeated, and the particles eventually converge on the location estimates. It is worth noting that the PF is iterated for each new estimation tuple. Alg. 2 depicts a pseudo-code of particle filtering to estimate the location from the CDOA efficiently. The CDOA-PF algorithm combines a Particle Filter technique with a Cross Difference of Arrival. The algorithm incorporates transition models and CDOA computations and iter-

actively updates particles that reflect the robot’s potential positions. The algorithm calculates the robot’s position by resampling particles according to their associated probabilities and doing so until the end of the robot’s trajectory.

3.4.2 CDOA-EM

Expectation Maximization (EM) is a grid-based localization that uses an explicit, discrete representation for the probability of all positions in the state space. We represent the environment by finite discrete state spaces (Grids). The algorithm updates the probability of each state of the entire space at each iteration. Use a fixed decomposition grid by discretizing each CDOA: (x, y, θ) . For each location $x_i = [x, y, \theta]$ in the configuration space: determine probability $P(x_i)$ of the robot being in that state. Then, it chooses the state with the highest probability. This approach resembles the EM method in (Pajovic et al., 2015a). Alg. 3 depicts the procedure of the EM approach that can be used to estimate the location from the CDOA efficiently. The CDOA-EM algorithm employs Expectation Maximization and Cross Difference of Arrival. It iteratively updates the robot’s position using transition models and CDOA calculations, populating a grid with positions as samples. The program determines the maximum probability grid position by converting CDOA into Gaussian probability and predicting the robot’s location until the end of the trajectory.

3.4.3 Summary of the system architecture

Fig. 3.2 delineates the system architecture of proposed CDOA-based indoor localization in which WSNs collaboration is shown in *CDOA* block where all nodes share RSSI and perform gradient calculation using Eq. (3.1), which was further converted into CDOA using Eq. (3.2) as mentioned in Alg. 1. CDOA with robots’ position estimation is used for PF resampling as discussed in Alg. 2, and similarly, for estimation maximization as mentioned in Alg. 3. These estimates will further be used for state estimation of the robot as the robot moves along a trajectory.

3.5 Theoretical Analysis

Assumption 1 Given the locations of static wireless nodes N , their range R in the wireless sensor network, and the position of the robot node x ; the following observation can be made about the CDOA estimation probability: CDOA is

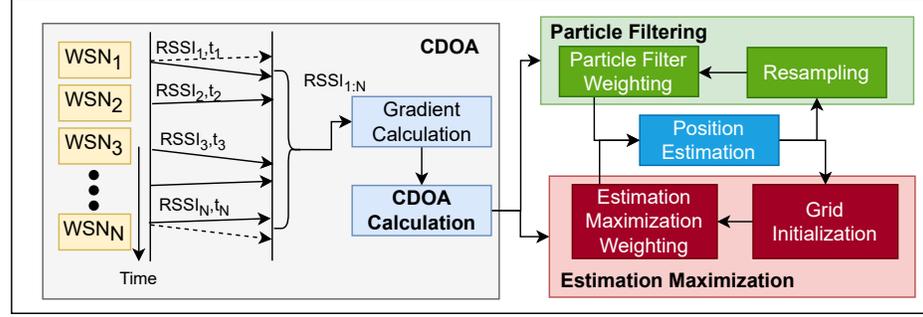


Figure 3.2: System architecture of the CDOA-based robotic node localization approach using particle filter and estimation maximization.

independent of the previous observations, i.e.,

$$P(CDOA_t | CDOA_{t-1}, N, R, x) = P(CDOA_t | N, R, x) \quad (3.5)$$

Assumption 2 Given X as a set of samples in PF and \mathfrak{R} as the resolution of spread. We assume that samples spread randomly in the space with the given resolution spread, greater than the centroid of converged samples in PF. i.e.,

$$\frac{\sum_{i=0}^{n-1} (X_i)}{n} \leq \mathfrak{R} \quad (3.6)$$

Lemma 1. *The error in the location estimation depends upon the cumulative noise percentage $\mathcal{N}\%$ of RSSI from each wireless sensor node in wireless sensor network.*

Proof. First, We prove the relation between the error in CDOA estimation of a robot at a candidate position l with cumulative noise $\mathcal{N}\%$ in RSSI from WSNs.

Let $\eta_x = \eta_{x,1}, \eta_{x,2}, \dots, \eta_{x,j}$, and $\eta_y = \eta_{1,y}, \eta_{2,y}, \dots, \eta_{k,y}$, are noise values of nodes in the horizontal and vertical axis in wireless sensor network respectively. Based on Eq. (3.1), the uncertainty values for g_x and g_y are $\sum_{i=1}^j (\eta_{x,i})$ and, $\sum_{i=1}^k (\eta_{i,y})$ respectively. Hence, calculated CDOA at position l has a cumulative percentage error of g_x and g_y . Next, we note that location estimation is the soft-max of weights of n samples in PF; hence the error in location estimation depends upon the cumulative percentage error of maximum weight in PF, which is further dependent upon the error of CDOA estimation using RSSI in wireless sensor network.

Algorithm 1: CDOA: WSN Collaboration-aided DOA Estimation at the Mobile Robot

```

1 At every wireless sensor node;
2    $S_i \leftarrow []$ , list for all RSSI values for node  $i$ ;
3   sample RSSI  $\rightarrow S_i$ ;
4   Initialize the time window to  $T$ ;
5   while time window is open do
6     |    $S_i \rightarrow$  average all RSSI received;
7   end
8   Publish  $S_i$  to the ROS network;
9 At the robot node;
10 for every wireless node  $i \in N = 1, 2, \dots, 4$  in WSNs do
11   |   Receive RSSI ( $S_i$ ) locally;
12 end
13 Calculate RSSI gradients in Eq. (3.1) using  $S_i \in N$ ;
14 Calculate CDOA at the robot using Eq. (3.2);

```

Moreover, as shown in (Penna & Cabric, 2011), the location estimation error variance estimated using the DOA metric can scale linearly with the DOA estimation variance and quadratically with the target distance (see Lemma 3). \square

Theorem 1. (Convergence) *The CDOA-based localization output will converge to the actual robot position in finite iterations. Let $x_i = \tau x_{i-1} + \varepsilon$, where i is the current iteration, $\tau \in X$ and $\varepsilon \approx \mathfrak{R}$ (resolution). We can claim that the infinite sequence, $x_{i=1}^{\infty}$ has an approximate solution in a finite number of iterations.*

Proof. Let x_1, x_2, \dots, x_n be the converged particles in the PF solution, and let c be the centroid of these particles. The particle filter algorithm iteratively updates the particles according to the likelihood function and the prior distribution. After a finite number of iterations, the particles converge to a stable solution that approximates the true position.

We can represent the particle filter algorithm as an iterative process, where the k -th iteration is represented by the function $g_k(x_{k-1}) = x_k$. We can prove that the sequence of iterates x_k converges to a fixed point x^* of the function $g(x) = \lim_{k \rightarrow \infty} g_k(x)$

Now, let x be any candidate particle with the minimum difference from each of the converged particles x_1, x_2, \dots, x_n . By the definition of the centroid, we know that $\sum_{i=1}^n |x_i - c|$ is minimized. Therefore, $|x - c|$ is also minimized,

Algorithm 2: CDOA-PF: Particle Filter Over CDOA for Mobile Robot Localization

```

1  $P_r = []$  % Initial list of particles in the PF ;
2 while end of trajectory do
3   draw sample  $x_t$  from transition model  $P(X_t|x_{t-1}, z_{t-1})$ ;
4   CDOA calculation using Algorithm 1;
5    $\omega = []$  % weights;
6   for  $x_t$  in  $P_r$  do
7     add  $P(z_t|x_t)$  to  $\omega$ ;
8   end
9    $R_r = []$  % re-sampling;
10  for  $i=1$  to  $n$  (number of particles) do
11    Choose  $p = P_r[i]$  of probability  $w_i(x_i) \in \omega$ ;
12    add  $p$  to  $R_r$ ;
13     $x_t^* = x_i \in w_i(x_i)$  is  $\max(w_i)$ ;
14  end
15   $P_r = R_r$ ;
16 end

```

which means that x is closest to c . Hence, we can say that position estimation can be obtained with uncertainty \mathcal{N} using Lemma 1, and the proposed algorithm converges over time and results in accurate location estimation.

A similar proof can be derived to guarantee the convergence of the CDOA-EM localization output. \square

Lemma 2. *The accuracy of location estimation depends upon the resolution spread \mathfrak{R} , i.e.,*

$$\operatorname{argmin}_{i=0}^{n-1} |x - X_i| \leq \mathfrak{R} \quad (3.7)$$

Proof. According to Theorem 1, we have proven that the PF/EM solution converges to the actual position after finite iterations. Let $X = X_0, X_1, \dots, X_{n-1}$ be a set of n known locations and let x be an unknown location that we wish to estimate. We are given that the location estimation accuracy depends upon the resolution spread $\leq \mathfrak{R}$ based on Assumption 2. To prove this lemma, we will show that for any x , the distance between x and its closest known location X_i is less than or equal to \mathfrak{R} .

We have demonstrated that Theorem 1's statement is true and that the state estimation converges to the actual position after a finite number of iterations.

Algorithm 3: CDOA-EM: Expectation Maximization Over CDOA for Mobile Robot Localization

```

1 Grid dimensions and resolution is given;
2  $E_r = []$  % Populate full-resolution grid positions as samples for EM;
3 while end of trajectory do
4     draw sample  $x_t$  from transition model  $P(X_t|x_{t-1}, z_{t-1})$ ;
5     CDOA at  $x_t$  calculation using Algorithm 1;
6      $\omega = []$  % weights;
7     for each grid position in  $E_R$  do
8         Convert CDOA into  $w_{E_r}(x_t) \in \omega$  Gaussian Probability using
           (Eq. (3.3) for  $E_r[t]$ ;
9          $x_t^* = x_t \in w_{E_r}(x_t)$  is  $\max(w_{E_r})$ ;
10    end
11 end

```

Now, define X as a collection of n known locations, where $X = X_0, X_1, \dots, X_{n-1}$ is the set, and define x as an unknown location that we want to estimate. By Assumption 2, we are informed that the resolution spread $\leq \mathfrak{R}$ determines how accurately we can estimate the position. To demonstrate this lemma, we shall demonstrate that for any x , x 's distance from its nearest known position X_i is less than or equal to Re .

First, we note that for any $i \in 0, 1, \dots, n-1$, the distance between x and X_i is given by $|x - X_i|$. Therefore, the closest known location X_j to x is the one that minimizes this distance, i.e., $\operatorname{argmin}_{i=0}^{n-1} |x - X_i| = X_j$. We want to show that $|x - X_j| \leq \mathfrak{R}$. To do this, we use the definition of $\operatorname{argmini} = 0^{n-1}(x - X_i) \leq \mathfrak{R}$. We know that $\operatorname{argmini} = 0^{n-1}(x - X_i) \leq \mathfrak{R}$ means that for any $i \neq j$, we have $|x - X_j| \leq |x - X_i| \leq \mathfrak{R}$. It follows that $|x - X_j| \leq \mathfrak{R}$ proves the lemma. \square

Lemma 3. (Coverage) *With a minimum of 4 WSNs or IoT nodes in the network available for collaboration with a sensing range of r each, the CDOA localization method's maximum coverage area is $\frac{r^2}{2}$, as long as the robot node to be localized is within the boundary of the collaborating wireless nodes.*

Proof. Assume that the four WSNs or IoT nodes A, B, C , and D are situated at the four corners of a square region (see Fig. 3.1 and 3.3). Assume each side of the square is s in length. The maximum distance between the robot i and any of these nodes must be less than or equal to the diagonal of the square, which is $\sqrt{(2)}s$, because A, B, C , and D are situated at the corners of the square. As a result, the sensing range r must be at least a distance of $\sqrt{(2)}s$ to connect the

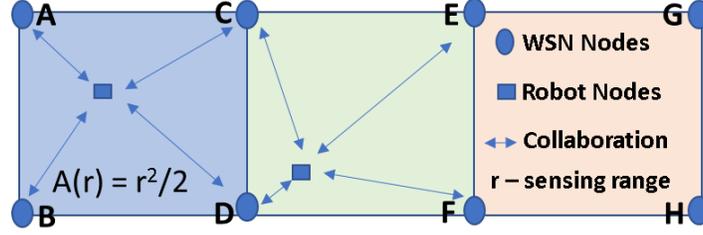


Figure 3.3: Depiction of coverage area for CDOA localization.

robot node with other supporting nodes in the network (i.e., $r \geq \sqrt{2}s$, meaning $s \leq \frac{r}{\sqrt{2}}$). The square region, which forms the outer boundary workspace of the robot node, has an area of s^2 . Therefore, the reliable area of the region that the CDOA approach can cover for localization with nodes having a sensing range of r is $\geq \frac{r^2}{2}$ as long as the robot is inside the boundaries of the four cooperating nodes. Moreover, expressing the coverage area $A(r)$ as a function of r , we can take its derivative as

$$\frac{\partial A(r)}{\partial r} = \frac{\partial (\frac{r^2}{2})}{\partial r} = 1 \quad (3.8)$$

We can see that this derivative is always positive and indicates the linearity of the coverage area concerning the number of nodes and the sensing range. \square

Remark 1 Minimum of three nodes (instead of 4 nodes) can be sufficient as per the forward or backward finite difference equation for node collaboration to estimate DOA (see (Parasuraman, Fabry, et al., 2013)), as long as all the nodes encompass the convex hull of the area boundary.

Remark 2 If more nodes are available to collaborate than the minimum number of nodes, this allows exploiting redundancy in the CDOA estimation, providing robustness and accuracy advantages.

Lemma 4. (Coverage Generalization) *The minimum area coverage mentioned in Lemma 3 can be generalized to a rectangular region with an area of $r^2 \cdot \frac{k}{k^2+1}$, where k is the width factor of the length and width of the new rectangular area, and r is the sensing range. The maximum coverage is achieved when the length is equal to the width (square region).*

Proof. Trivial. Given are the dimensions of a rectangle: length l , width w , and aspect ratio k (the width factor that makes $w = kl$). The sensing range r must be greater or equal to the largest diagonal of the rectangle. That is, $r \geq \sqrt{(l^2 + w^2)} \geq l^2(1 + k^2)$. Therefore, the coverage area is expressed as $l^2 \leq r^2 \cdot \frac{k}{1+k^2}$. Moreover, it is trivial to observe that the width and length must be identical in

order to cover the most area, i.e., when $k = 1$. This relates to a square region. \square

Remark 3 The CDOA estimation and coverage area can be generalized to an arbitrary polygonal shape of the boundary of the localization workspace as long as the convex hull of the boundary nodes can be defined. For instance, the CDOA can be estimated using gradient estimation algorithms as proposed in (D. Han et al., 2009; Verma et al., 2018) when the WSNs are distributed in the workspace without a specific geometric pattern.

Lemma 5. Number of Collaborating Nodes *A minimum of $2n + 2$ number of nodes are required to cover an area of $n\frac{r^2}{2}$, for wireless nodes with sensing range r .*

Proof. Let A be the unit area of the square that can be covered by 4 WSNs (using the result of Lemma 3. To extend this coverage beyond this unit area with a scaling factor of n , we would need $2n$ more nodes, which require replicating the square region n number of times as shown in Fig. 3.3. With $2n + 2$ nodes, the maximum area of coverage then becomes $nA = n\frac{r^2}{2}$. For example, for a unit area A and $n = 1$, we need four nodes (nodes A-D); with double the coverage area, we need a minimum of 6 nodes (nodes A-F). Therefore, the number of nodes required for CDOA scales linearly with the coverage requirement. A single node's coverage region is indicated by r^2 . Let n be the bare minimum of nodes needed to cover area A . We'll now demonstrate that n nodes are required to cover area A completely and that any number of nodes below n falls short of this requirement. The area bounded by n nodes can be expressed as nr^2 . We have $nr^2 \geq A$ because area A must be covered by n nodes. Now, let k be the number of nodes such that $k < n$. We can express the area covered by k nodes as kr^2 . Since $k < n$, we have $kr^2 < A$. Therefore, we have $nr^2 \geq A > kr^2$. This means that n nodes are sufficient to cover area A and that any number of nodes less than n is insufficient to cover area A . Now, we can use this to find an upper bound for n :

$$A = nr^2 \implies n = \frac{A}{r^2} \quad (3.9)$$

We know that n is a positive integer. Therefore, the smallest possible value for n is $\left\lceil \frac{A}{r^2} \right\rceil$. Where $\lceil x \rceil$ is the smallest integer greater than or equal to x . We can now use this to find an upper bound for n :

$$n \leq \left\lceil \frac{A}{r^2} \right\rceil \leq \frac{A}{r^2} + 1 \quad (3.10)$$

We can substitute this into the lemma's statement to get the desired result:

$$n \leq \frac{A}{r^2} + 1 \leq 2n + 2 \quad (3.11)$$

Hence, A minimum of n number of nodes are required to cover an area of $2n + 2$ per r^2 of area to be covered. \square

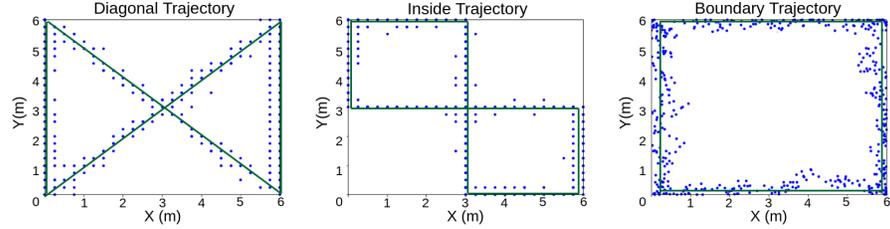


Figure 3.4: The test sample locations in the simulation experiments are shown here. The plots show the robot's actual trajectory (where the signal sample was taken) in solid lines and the predicted locations in scattered dots.

Theorem 2. (CDOA Scalability) *Combining the results of Lemmas 4 and 5, the approximate linear relationship between the number of nodes n and the coverage area A for localization using the CDOA approach can be expressed as $A \approx cn + d$, where c and d are constants.*

Proof. We will prove this theorem using the mathematical induction principle. Considering Eq. (3.1), which uses four wireless nodes ($n = 4$) and proposed approach able to find position estimation in the bounded region, given by algorithm Alg.1, also, in this case, the coverage area is $A_4 = 4c + d$ where c and d both are constants, which satisfies the theorem (see Lemma 4). Next, we will assume that the theorem holds for some arbitrary value of $n = k$ where $k > 4$. In other words, we assume that $A_k \approx ck + d$. Now, we need to prove that the theorem also holds for $n = k + 1$. The coverage area for $n = k + 1$ nodes is given by $A_{k+1} = A_k + \Delta A$, where ΔA is the increase in coverage area due to the addition of one more node. Since the CDOA approach is based on the DOA of RSSI, it can be assumed that the increase in coverage area due to adding one more node is approximately proportional to the existing coverage area. This means that $\Delta A \approx \alpha A_k$, where α is a constant (see Lemma 5). Substituting the value of A_k from the induction hypothesis, we get $\Delta A \approx \alpha(ck + d)$. This means that the total coverage area for $n = k + 1$ nodes is given by:

$$A_{k+1} \approx ck + d + \alpha(ck + d) = (c + \alpha)k + (d + \alpha d). \quad (3.12)$$

$$A_1 = d; A_{k+1} \approx (c + \alpha)k + (d + \alpha d); A_n \approx cn + d \quad (3.13)$$

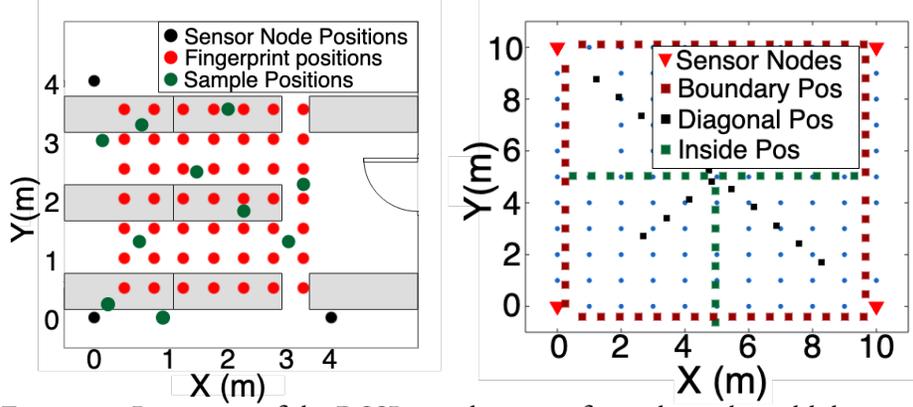


Figure 3.5: Locations of the RSSI sample points from the real-world datasets 1 (Left) and 2 (Right) discussed in Sec. 3.6.3.

Thus, we have proved that the theorem holds for $n = k + 1$. Since we have established the base case and have shown that the theorem holds for all $n = k + 1$ if it holds for some arbitrary value of $n = k$, we can conclude that the theorem holds for all $n \geq 4$. \square

Computational Complexity Let n be the number of samples (or particles) in the EM (or PF), and N be the number of cooperative nodes in the wireless sensor network. At each step, the robot needs to find pose estimation based on our proposed WSN cooperative localization algorithm, which involves the following steps:

- EM and PF initialization with random particles: $O(n)$.
- Weight transfer to new PF for each sample in $O(n)$.
- Robot-WSN collaboration in an open time window of α for sharing and receiving wireless signals from N nodes in $O(\alpha N)$ time.
- Sample weight calculation based on CDOA in $O(n)$.
- Finding the soft max from particle weight distribution as the best pose estimate in $O(n/2)$.

Therefore, one iteration is $O(n \times \alpha(N))$ where α and N are constant values and have a low impact on overall computation complexity; hence, overall time consumed by the proposed algorithm would be $O(n)$.

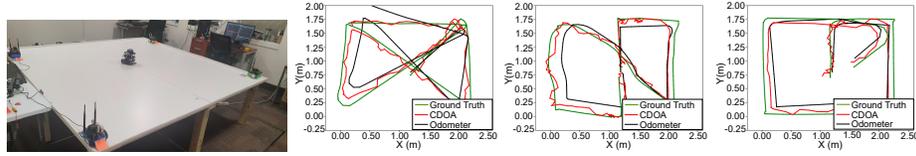


Figure 3.6: Hardware experiment setup and samples of the output robot’s trajectories (diagonal, inside, boundary) in the real robot experiments (See Sec. 3.7.3).

Table 3.2: Experiment Configurations

Simulation Parameters	Experiment Basis			
	Simulations	Dataset1	Dataset2	Hardware
Space Dimensions	6×6	4×4	10×10	2.34×1.75
Resolution of CDOA-EM (ppi)	0.05	0.05	0.1	0.05
Resolution of CDOA-PF (ppi)	0.08	0.1	0.5	0.08

3.6 Experimental Validation

We implemented our approach and compared the performance with relevant recent methods from the literature (see Sec. 3.6.1). We performed extensive experiments through simulations (Sec. 3.6.2), real-world datasets (Sec. 3.6.3), and real robot in-house experiments (Sec. 3.6.4) to verify and validate the performance of the proposed localization in terms of accuracy measured through the Root Mean Squared Error (RMSE) and efficiency measured through the Time Per Iteration (TPI) metrics. In each experiment, we made 100 trials and averaged the localization error over all trials as the distance between the predicted and the actual positions (ground truth). The experiment settings shown in Table 3.2 show diverse settings under which we evaluate the proposed method.

3.6.1 Comparison with the State-of-the-Art (SOTA)

To validate the results of our proposed approach, we implemented the five model-based solutions from the recent literature:

1. **Trilateration** (B. Yang et al., 2020a)
2. **WCL: Weighted centroid localization** (Z.-M. Wang & Zheng, 2014)
3. **D-RSSI: differential RSSI-based localization** (Podevijn et al., 2018)
4. **I-RSSI: Improved RSSI based localization** (W. Xue et al., 2017)

5. **PF-EKF: Particle filter - Extended Kalman Filter** (Zafari et al., 2018)
6. **SBL-DOA: Sparse Bayesian Learning applied over the Direction of Arrival** (H. Wang et al., 2019)

Our CDOA-PF approach applies the Gaussian probability over CDOA for a fixed number of sampled particles, and in our CDOA-EM, the solution is extensively searched through each and every grid of the workspace with a fixed resolution (the EM implementation is similar to the method in (Pajovic et al., 2015a)). More details on each of these methods are included in Appendix.A.

3.6.2 Numerical Analysis with Simulations

We simulated four WSNs distributed on the corners of the simulation workspace, with the robot’s initial position being their center (Fig. 3.1). We simulated three different trajectories for robot motion: Boundary (left), Cross Coverage (center), Diagonal (right), and the scale of the workspace is 6m x 6m, as shown in Fig. 3.4. Through these paths, we cover all potential positions within the bounded region.

We measure the RSS value of each WSN for all positions along the robot’s path. The estimated RSS based on the log-normal radio signal fading model is computed as:

$$RSSI = A - 10 \times \eta \times \log_{10}(d), \quad (3.14)$$

where η denotes the path loss exponent, which varies between 2 (free space) and 6 (complex indoor environment), d denotes the distance from Robot R to the node N , and A denotes received signal strength at a reference distance of one meter. We used this setup to perform experiments and validate the accuracy of localization techniques for different noise conditions on the measurements simulated through a zero-mean Gaussian noise varying from 1 to 4 dBm variance. The path loss exponent η is set to 3 in our simulations to present a reasonable indoor environmental channel in our simulations. The simulation effectively mimics noise in RSSI by incorporating factors like distance, signal frequency, and environmental conditions. Moreover, with varied noise levels, it aptly represents diverse real-world scenarios, substantiating its representativeness in our experiments.

3.6.3 Real-world Datasets

We used two different publicly available real-world RSSI datasets on indoor localization. See Fig. 3.5 for the illustration of the workspace of the datasets

along with the positions of the WSNs and locations where the RSSI samples are measured.

¹ <https://github.com/pspachos/RSSI-Dataset-for-Indoor-Localization-Fingerprinting>

Dataset 1¹ Provides RSSI values for three wireless technologies; BLE, Zigbee, and Wi-Fi in a 4m x 4m room. We have used the data for Scenario 1 of this dataset as it relates to our approach to the geometric positioning of anchor nodes. In this scenario, a room of 6.0 x 5.5 m was used as the experimental testbed. All transmitting devices were removed from the surroundings to establish a transparent testing medium where all devices could communicate without interference. The transmitters were spaced 4 meters apart in the shape of a triangle. The fingerprint and test points were obtained with a 0.5 m distance between the transmitters in the center. The database would be made up of 49 fingerprints due to this. Ten test points were chosen at random for testing. We have arranged the fingerprinting dataset in such a way that it makes a trajectory in the region.

² <https://ieee-dataport.org/documents/multi-channel-ble-rssi-measurements-indoor-localization>

Dataset 2² Provides RSSI values for the three regions of varying ranges in the bounded area: diagonal, boundary, and inside in a 10m x 10m room. Four anchors took RSSI measurements while receiving messages from a single mobile node, delivering advertisement and extended advertisement messages in all BLE channels (both primary and secondary advertisement channels). Four anchors were placed in the corners of a 10 x 10 m office area (no considerable impediments). We have compared the results for different communication channels under different regions in the bounded area.

3.6.4 Real Robot Hardware Experiments

To validate the practicality of the algorithm for real-world scenarios, we performed hardware experiments on a testbed of dimensions $2.34m \times 1.75m$, with the ceiling-mounted camera for visual localization as ground truth; we have mounted a wireless access point of power 20dBm with a 2.4Ghz frequency over the top of a Turtlebot3 mobile robot.

Fig. 3.6 shows the experimentation test-bed and sample trajectories with the Turtlebot3 robot at the center of the WSNs on the four corners. In each trial, we drove the robot remotely and recorded RSSI and ground truth position with an overhead camera-based fiducial marker (AprilTag (J. Wang & Olson, 2016)) tracking). Robot Operating System³ (ROS (Quigley et al., 2009)) has been employed for inter-node communication, as ROS is the de-facto software framework used in the robotics literature. The ROS master runs a service on the experimental robot to receive perceived signal strength from all connected nodes through a synchronization service. The WSN nodes operate at a 10Hz

³ <https://wiki.ros.org/>

rate to calculate the RSSI and publish these values to the ROS topics. Five trials have been conducted for each of the three different trajectories.

3.7 Results

Table 3.3 comprehensively presents the overall DOA-only localization performance, time complexity, and efficiency results of the proposed approach compared to the SOTA approaches in simulation, real-world datasets, and real robot hardware experiments. For time complexity, the variables n , S , and r represent the number of particles in PF, the size of the grid in EM, and the resolution of grids, respectively. Looking at the TPI (efficiency) metric, the model-based approaches such as Trilateration and WCL are the fastest because of the low computational requirement they need for every new sample. However, the CDOA-PF is comparable to the model-based methods in terms of real-time computational tractability, allowing the possibility for instantaneous localization. In general, the CDOA methods performed significantly better than the baselines in terms of higher accuracy and reasonable efficiency. More details on the results are discussed below.

Table 3.3: Overall performance results with statistics of various experiments from simulations, datasets, and hardware trials.

Algorithm	Complexity	Average TPI (ms)	Localization Error RMSE (m)			
			Simulations	Dataset1	Dataset2	Hardware
Trilateration	$O(1)$	82±18	1.22 ± 0.56	2.70 ± 1.78	3.73 ± 2.27	1.63 ± 0.85
WCL	$O(1)$	83 ± 27	2.47 ± 0.84	3.94 ± 0.95	4.93 ± 3.26	2.54 ± 1.21
D-RSSI	$O(1)$	86 ± 18	0.55 ± 0.13	2.18 ± 1.26	2.92 ± 1.52	1.53 ± 0.72
I-RSSI	$O(1)$	94 ± 21	0.42 ± 0.09	1.98 ± 0.86	2.43 ± 1.21	1.11 ± 0.57
PF-EKF	$O(n)$	111 ± 34	0.91 ± 0.18	2.34 ± 1.64	3.12 ± 1.78	1.74 ± 0.91
SBL-DOA	$O(n)$	127 ± 42	0.16 ± 0.06	1.81 ± 0.69	1.72 ± 0.48	0.54 ± 0.21
CDOA-EM (Ours)	$O(S * r)$	270 ± 99	0.13±0.04	1.58 ± 0.53	1.66±0.73	0.34 ± 0.08
CDOA-PF (Ours)	$O(n)$	102 ± 25	0.15 ± 0.05	1.17±0.48	1.67 ± 0.83	0.12±0.16

3.7.1 Simulation Results

Fig. 3.7 summarizes the results of the simulation experiments. The proposed CDOA approaches outperformed all SOTA methods and have been shown to localize a mobile device using an existing WSN or AP infrastructure with up to 8 cm accuracy achieved in our simulation environment of 6x6 m, even in high signal noise of 4dBm. The CDOA-EM method provided the best accuracy among all methods, while the I-RSSI approach had high localization accuracy among the SOTA algorithms. Furthermore, in comparison to efficiency, CDOA-PF

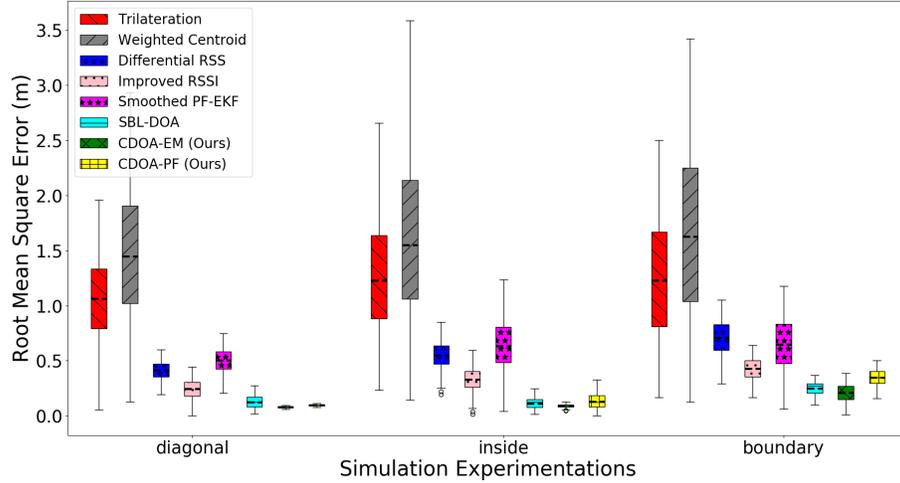


Figure 3.7: Comparative localization performance in the simulation environment.

is 40% more efficient than CDOA-EM in simulations, as expected. SBL-DOA demonstrate comparable localization accuracy than proposed approach but is 10% less efficient than CDOA-PF. It can be seen that the weighted centroid has the least accuracy of 84%, and the proposed approach has the highest accuracy of 92% compared to the ground truth location. Also, as expected, the insider and diagonal trajectories provide better accuracies than the boundary cases for all the compared methods. The RSSI and DOA-based location estimates can be ambiguous on the boundary regions, resulting in higher localization errors.

The PF-EKF approach used the same number of particle filters and applied an EKF to predict and update the robot’s state while using raw RSSI values (which are inconsistent and unreliable (Dong & Dargie, 2012a)) as an observation, resulting in 30% reduced position estimation accuracy than our approach. In addition, the EKF update and prediction step in PF-EKF added overhead in the computation and made it complex, while the proposed CDOA-PF has no such computationally expensive operation; hence time per iteration of CDOA-PF is 60% less than that of PF-EKF. Overall, the proposed CDOA-PF achieved a balance of high localization accuracy and efficiency over all other SOTA algorithms.

Ablation Analysis

In the proposed approach, certain factors, such as the number of particles and noise level in RSSI measurements, can impact localization accuracy.

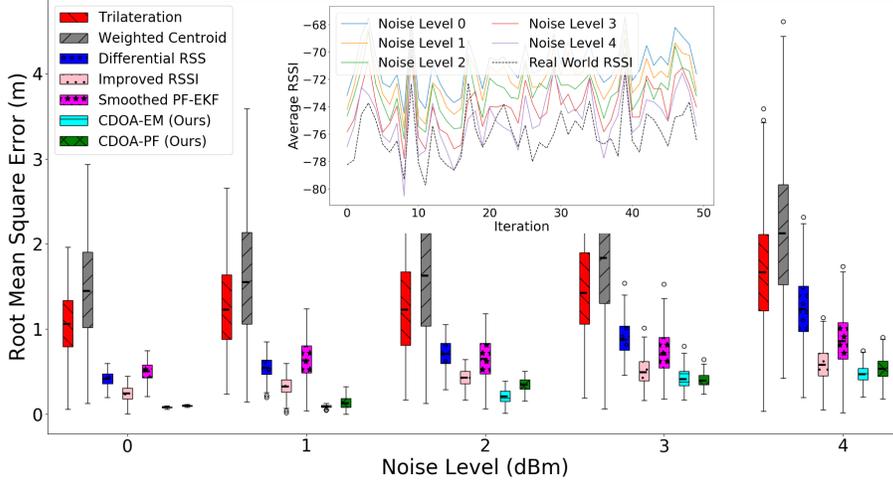


Figure 3.8: Localization performance under different RSSI noise levels. The embedded plot represents the RSSI variation at different noise levels to visualize the representativeness of the simulated RSSI with real-world data.

We analyze the localization accuracy (RMSE) for different simulated noise levels in the measured RSSI values. It can be seen in Fig. 3.8 that the proposed approaches have lower RMSE (high accuracy) among all techniques, even under high noise levels. The accuracy improvement is more pronounced when the noise level is increased. The trilateration approach performed better than the Weighted Centroid method. However, both have 3x lower accuracy than the proposed EM and PF-based methods for most experiments. CDOA-EM performed slightly better than CDOA-PF in terms of accuracy and robustness. However, the CDOA-EM approach is computationally complex as it calculates the Gaussian probability in all the grid areas with a full resolution instead of sparsely and randomly distributed particles in the CDOA-PF.

Next, we present an ablation analysis of the number of particles in the proposed CDOA-PF method since it is a sampling-based method depending heavily on the number of particles. Fig. 3.9 presents the results of the RMSE variations under different numbers of particles in the proposed CDOA-PF approach. The accuracy has improved by increasing the number of particles, but not so much beyond a certain optimum value. Also, more particles increased the computational complexity. Implementing the motion model over PF significantly improved the accuracy at a small expense in the computation time. The results demonstrate the potential of integrating the mobile robot’s motion model (or

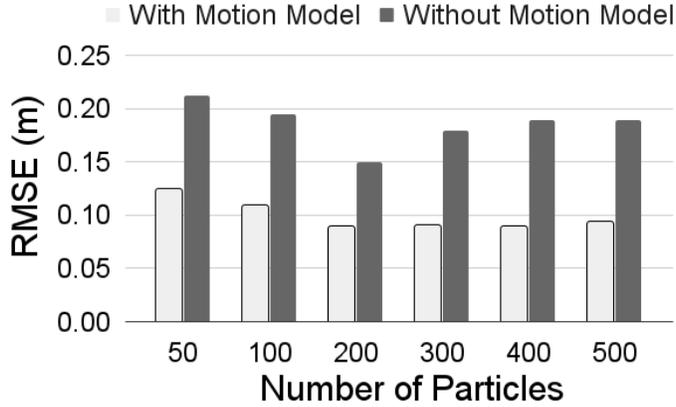


Figure 3.9: Impact on the localization accuracy of CDOA-PF by the number of particles and inclusion of robot motion model (odometry).

IMU sensor), when possible, to improve the CDOA-based localization solution.

3.7.2 Real-world Public Datasets

In general, our method outperformed other methods in both datasets, as can be seen in Table 3.3. Further analysis of individual technologies and channels is provided below.

Dataset 1: Wireless technology comparison Dataset 1 captures RSSI observations from three technologies (Wi-Fi, BLE, and Zigbee). The findings from these experiments (Fig. 3.10) delineate the suitability of using Wi-Fi as a communication channel and the proposed approach for indoor localization, as it has the least RMSE than other technologies, with 1.07m RMSE in a 4m x 4m of bounded region. As expected, there is no significant difference in computational complexity among different technologies because the methods take the same time to run the algorithm, irrespective of where the signals are coming from. However, CDOA-EM requires high time per iteration for position prediction than any other method. The proposed CDOA approach also has 0.4 m better accuracy than the KNN as presented in (Sadowski et al., 2020) scenario 1 of Dataset 1. SBL-DOA has utilized a similar direction of arrival technique and can achieve relatively higher localization accuracy than other SOTA approaches but slightly less than the proposed CDOA-EM/PF. The CDOA-PF provided higher accuracy than the CDOA-EM, contrary to the expectation that EM provides better PF accuracy. We believe this is because the softmax-based

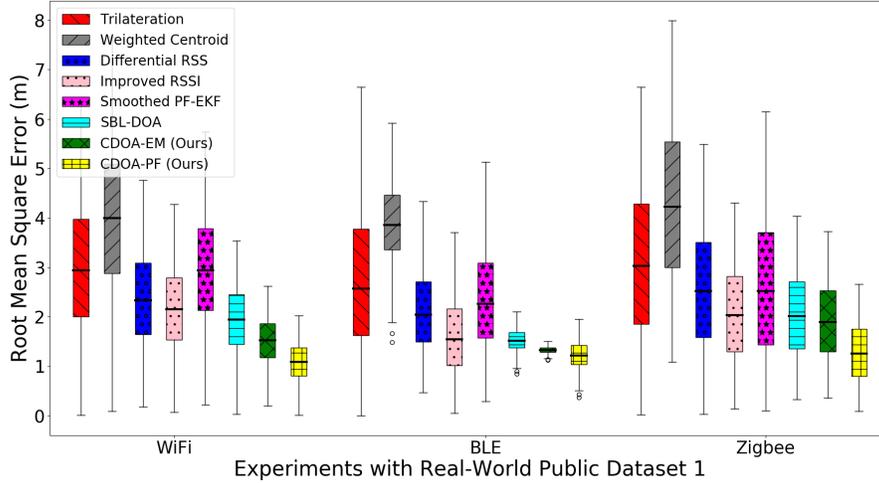


Figure 3.10: Comparative localization performance on the real-world dataset 1.

convergence of a small number of particles in PF after several iterations resulted in the closest position estimate rather than a more significant number of grids converging in the EM. This is especially the case for certain positions where the bounding centroid of grids is very close to the actual location, which yields lower RMSE in CDOA-PF for longer trajectories, as in this dataset.

Dataset 2: Regional and Channel-wise comparison Dataset 2 has RSSI captured in three regions (inside, diagonal, and boundary) in a bounded size 10 x 10 m. All the localization methods work well for the positions inside the AP/WSN perimeter, compared to diagonal or boundary points, similar to the results found in the simulations. However, the robot node would generally be inside the infrastructure boundaries, exploiting the full advantages of the proposed CDOA-based localization methods. Accordingly, the CDOA methods provided the best accuracy compared to other methods.

Dataset 2 also has RSSI observations for 40 different channels under three regions in the bounded dimension 10 x 10 m. We show the results for the combined RMSE for channels 0-39. It can be seen from Table 3.3 that the proposed approach consistently provided the best performance in most of the scenarios with reasonable computational efficiency. We also analyzed the individual channel performance and found that the comparative results did not differ significantly compared to the averaged channel estimates. However, some channels were found to have higher noise and, therefore, poorer performance on all the compared methods. SBL-DOA approach over different channels performed more or less the same as the approach relied on DOA and Bayesian learning to

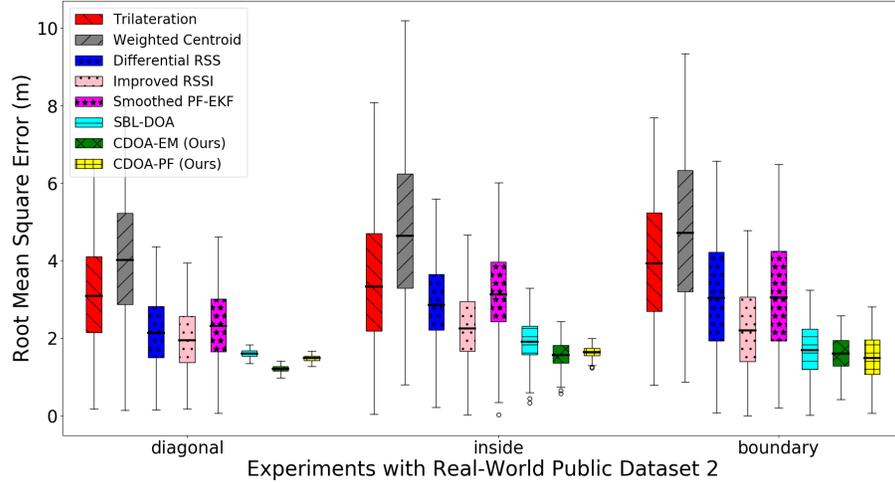


Figure 3.11: Comparative localization performance on the real-world dataset 2.

mitigate the effect of channel variations; hence it achieved close localization accuracy than the proposed method at the cost of high computational cost which makes it less practical in the robotic domain.

3.7.3 Real Robot Hardware Experiments

Fig. 3.12 provides the averaged results of the real robot experiments in different regions. The results delineate high average localization accuracy of 95%, with an absolute RMSE of 0.12 m (in a bounded region of $4m^2$) for CDOA methods in our hardware experiments. A sample of the hardware experiments and demonstration is shown in the attached video⁴. As compared to other SOTA; SBL-DOA performed better and has shown 90% localization accuracy but 5% less accurate than proposed CDOA method.

⁴ Also available at <https://hero.uga.edu/research/localization/>

Table 3.3 also provides comparative numerical values of localization error and time per iteration for other SOTA approaches, which validate the significant performance improvement achieved by the proposed CDOA methods in all trajectories. CDOA-EM has relatively better localization than different approaches but with a high computational cost. The CDOA-PF has more than 93% localization compared to other model-based approaches, confirming the simulation experiments. Interestingly, the CDOA-PF is 35% more accurate than CDOA-EM, similar to the observation in dataset 1.

Furthermore, the hardware experiments also validated the practicality of the proposed CDOA approach by providing evidence of high localization accuracy, similar to simulation and dataset results. Finally, we have made a hardware exper-

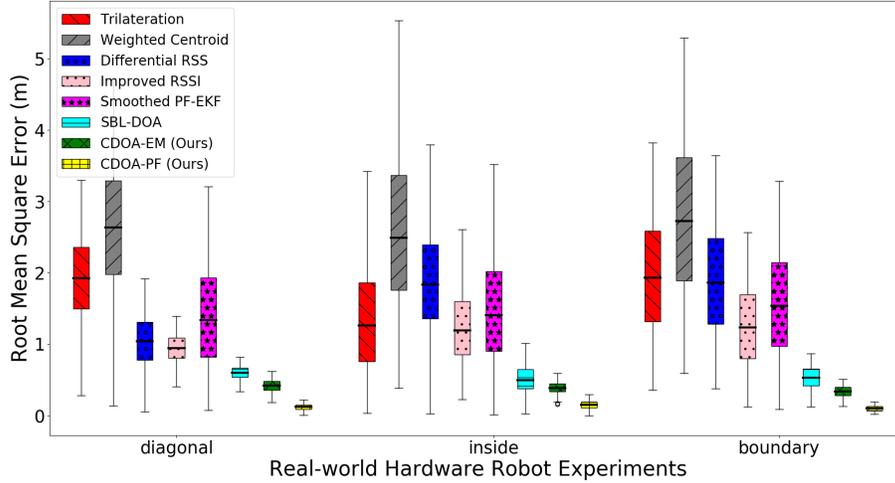


Figure 3.12: Comparative localization performance on the real robot experiments.

iments dataset, and the software source codes of the implementations, available for the research community to reproduce the results and build on the work to improve DOA-based localization approaches further. The proposed CDOA-PF and CDOA-EM approaches, given their impressive localization accuracy in both simulated and real-world environments, should operate effectively even in non-line-of-sight (NLOS) scenarios. These methods, especially the CDOA-PF, combine robustness to variations in signal environment with computational efficiency, making them likely to handle NLOS conditions where signals may be blocked or distorted. Moreover, as these methods have shown superior performance in high signal noise scenarios, they should be robust to the increased noise and multipath effects commonly found in NLOS situations.

Limitations: Similar to any wireless node collaboration-based approach (H. Wang et al., 2019), the proposed approach only works for more than three fixed nodes placed at geometrically-aligned positions of a regular polygon bounded region to obtain accurate CDOA. While it is robust for most scenarios, it depends on the quality of the RSSI and the obstacles or non-line of sight conditions, which need to be studied further.

3.8 Summary

Indoor localization of mobile robotic nodes in GPS-denied environments presents a significant challenge due to the dynamic nature and unstructured layout of

such environments. Conventional technologies, including cameras and LIDAR-based sensors, often fail in these circumstances. Wireless signal-based localization has been an area of considerable focus in the literature, primarily emphasizing fingerprinting and feature-matching paradigms. However, these require dedicated, environment-specific offline data collection, limiting their usability in various scenarios.

This chapter proposes a novel, online robot localization algorithm that leverages collaborative wireless sensor nodes to address these issues. The central innovation of the approach lies in determining the Collaborative Direction of Arrival (CDOA) of wireless signals. The CDOA is obtained by utilizing geometric features and collaboration between wireless nodes. This information is then processed with the Expectation Maximization (EM) and Particle Filter (PF) algorithms to calculate the Gaussian probability of the node's location with high efficiency and accuracy. Notably, the algorithm uses Received Signal Strength Indicator (RSSI) data, making it applicable to resource-constrained devices.

We thoroughly analyze the proposed approach's consistency, accuracy, and computational efficiency. This analysis includes extensive validation through simulations, real-world public datasets, and real robot demonstrations. The proposed method demonstrates impressive real-time computational capability and delivers substantial centimeter-level localization accuracy. The results reveal a significant improvement in accuracy and efficiency compared to state-of-the-art localization approaches.

The primary contributions of the chapter include the proposal of a new collaboration-aided mechanism for collecting RSSI data from wireless sensor nodes and estimating the CDOA, integration of this CDOA with a Bayesian framework for robust node localization, and extensive experimental analysis. We further offer a thorough theoretical analysis of the proposed CDOA localization method, considering factors such as localization consistency, accuracy, area coverage, scalability, and computational complexity. Lastly, all codes and datasets used in the research are made open-source, allowing the wider research community to reproduce and extend the study's findings.

Our method revealed at least a 50% improvement in localization accuracy than other non-sampling-based methods such as trilateration, weighted centroid, etc., using RSSI-only data, and 20% better than sampling-based techniques. Furthermore, despite CDOA being a sample-based technique, due to the less computational cost of CDOA and less frequent resampling, it is more efficient than other sampling and non-sampling-based methods. The experiments proved the practicality of the approach for cooperative robot localization

to achieve high accuracy with low computational costs. In the next chapter, we will discuss a synergistic localization approach to combine the proposed approach in this chapter and the Bayesian information technique to improve localization accuracy.

CHAPTER 4

MULTI-ROBOT SYNERGISTIC LOCALIZATION IN DYNAMIC ENVIRONMENTS

4.1 Introduction

This chapter extends the proposed idea of CDOA localization discussed in the previous chapter to multi-robotic systems. For a robot to navigate in any area, it must be familiar with the aspects of that environment and its position concerning those features. For autonomous mobile robot systems, localization is a critical issue. In static contexts where the map does not change, current solutions to the so-called simultaneous localization and mapping (SLAM) problem (Grisetti et al., 2007) can be applied by creating a map that is then used for the rest of the localization. However, while standard localization techniques work well in static environments, they may not work well in highly dynamic and complicated environments like warehouses, logistics centers, and production halls. The difficulty with these situations is that vast and quick changes infrequently happen, changing the map for a longer length of time. People relocating storage boxes or rearranging shelving are examples of such modifications.

Localization is an estimation of robot pose using the data collected from sensors and previously built maps. The data collected by the sensors is used by the robot to calculate its current position while in motion and to create a map of the environment. The cognition component is crucial in assessing the surroundings and determining the robot's response. A method of navigating in an already mapped or unmapped environment can be defined as the motion control of a robot. Motion control will be used to determine the robot's trajectory in various exploring tactics (Kshirsagar et al., 2018). However, determining

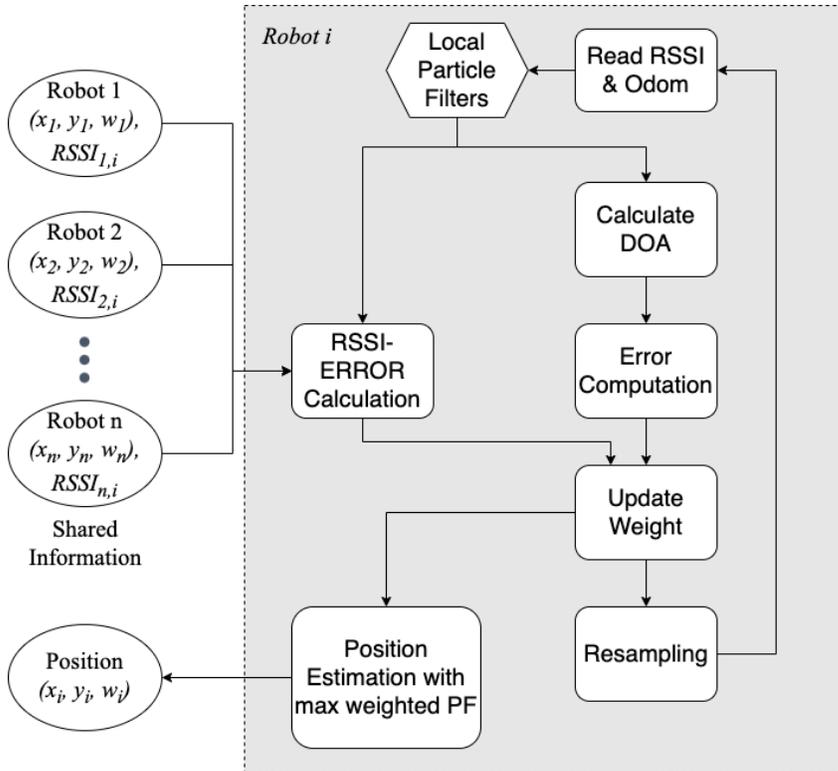


Figure 4.1: System architecture showing input and output for robot i . External robot position estimates and RSSI are shown in circles out of the gray box as robot j : (x_j, y_j, w_j) and $RSSI_j$ are the best-fit pose estimates, and RSSI of robot j from its particle filter, (x_i, y_i, w_i) indicates the state estimate of robot i based only on received measurements from connected n robots in the MRSL.

accurate position estimation is a challenge for a system containing multiple collaborating robots in an indoor environment with dynamic surroundings.

Collaborative Multi-Robot Localization (CMRL) (Fox et al., 1999), which can give a drift-free map-relative position estimate for robotic robots, is one way of solving the GNSS-denied localization difficulty. CMRL compares environmental observations to a known map to determine the robot's most likely position in map coordinates.

CMRL can provide localization precision on par with map resolution in most cases. The CMRL solution's performance, on the other hand, is strongly reliant on the data available in the local topography near the robot. As a result, localization convergence may be poor or unattainable for robots crossing featureless terrain. One possible approach is for robots to work together in groups.

A multi-robot strategy allows robots to share information, allowing all agents to converge even if certain people would not be able to do so otherwise.

The well-known issue of noise correlation amongst robots (Mokhtarzadeh & Gebre-Egziabher, 2014; Prorok & Martinoli, 2011) must be addressed by multi-agent teams sharing information. The underlying assumption of most commonly used estimators, such as Extended Kalman Filters (EKF) (Parasuraman et al., 2018) and particle filters (PF) (Parashar & Parasuraman, 2020), is that noise on new observations is uncorrelated with all past measurements. When this assumption is broken, the filter fails to capture the genuine robot state uncertainty and incorrectly converges to an inaccurate position, resulting in inconsistent estimators.

In some cases, combining data from many robots functioning in concert can reduce some problems associated with GNSS-denied navigation. For example, when faced with flat, altered, or unfamiliar terrain, a team of robots may assign several agents to stay in recognized informative zones to assist others venturing into the featureless ground. When cars navigate without external input and are vulnerable to inertial drift, combining readings from a group of robots effectively increases the total number of sensors available to each agent, reducing mistakes. In some situations, a robot may even be able to stay within GNSS signal range and communicate high-accuracy localization data with the rest of the team.

This paper proposes a decentralized synergistic localization for multi-robotic systems in a dense and dynamic environment. We propose a new Multi-Robot Synergistic Localization (MRSL) algorithm. Every robot can localize itself independently and update its position estimation using pose and sensor data from connected robots. A robot will update its position estimation whenever new information is obtained from its neighbors. When the system senses the presence of other robots in the region, it exchanges position estimates and merges the received data to improve its localization accuracy.

For individual robot localization, we have extended the algorithm for particle filter localization (Parashar & Parasuraman, 2020) of an access point in a certain way that robots localize themselves using RSSI as a source to calculate bearing as the direction of arrival (DOA). While many multi-robot collaboration algorithms from the literature use a Covariance Intersection (CI) based information integration, our approach uses Bayesian rule-based integration, which has shown to be computationally efficient and applicable to asynchronous communication. See Fig. 4.1 for an overview of the MRSL.

The main contributions of this paper are outlined below.

1. We propose a multi-robot synergistic localization algorithm using a particle filter updated based on the DOA and shared pose estimates with Received RSSI sensed between the robots (each robot having an AP).
2. We verify the efficacy of the proposed MRSL algorithm through extensive numerical simulations and analyze the methods from the perspective of impacts sharing RSSI and distance information.
3. We validate the accuracy and efficiency of the MRSL compared to autonomous robot localization (ARL) (Parashar & Parasuraman, 2020), which works without collaboration between robots, and the recent TRN-based collaborative multi-robot localization (Wiktor & Rock, 2020), which used covariance intersection to address the temporal correlation between received signals.

The core novelty of MRSL lies in that we employ a Bayesian rule to fuse shared information among robots in a connected network to achieve high localization accuracy, instead of Co-variance Intersection (CI) for information merging to reduce one source of measurement correlation while appropriately incorporating others have used (as used in relevant methods in the literature). Our method achieves superior localization accuracy through these novelties while enabling real-time efficiency compared to several state-of-the-art solutions.

4.2 Related Work

There has been substantial research on cooperative localization by other researchers. For example, Roumeliotis (Roumeliotis & Bekey, 2002) developed communal localization through a decentralized Kalman filter onboard each robot in a seminal study based on the distributed heterogeneous multi-robot localization (Madhavan et al., 2002) exploited extended Kalman filter. This filter calculates robot poses while simultaneously tracking the cross-correlation terms introduced by inter-robot data sharing, which further expanded to heterogeneous robot teams and outdoor contexts, as well as close range, bearing, and orientation measurements (Martinelli et al., 2005). However, these approaches are based on the Kalman filter and assume that a unimodal Gaussian may represent the posture estimate (Hamer & D’Andrea, 2018; Ullah et al., 2020). These methods also need each robot’s filter to estimate the condition of all other robots, which does not scale well when there are many robots.

Research work (Mokhtarzadeh & Gebre-Egziabher, 2014) presented a method for calculating only the robot’s state utilizing inter-robot measurements to avoid

assessing the status of the other robots. Through the use of Covariance Intersection (CI), the method directly addresses correlation issues. When applied to measures that are, in fact, uncorrelated, CI properly fuses measurements with an unknown amount of correlation, but it is conservative and lowers convergence. According to the authors, signals from a well-localized agent can be transferred to a robot with a poor localization estimate. This approach, however, can only be used with an unimodal Gaussian mean and co-variance-variance representation. The Gaussian distribution utilized in a Kalman or CI filter fails to approximate position estimations in natural terrain because they are inherently multi-modal.

Recent research has expanded on the strategies to take into account cross-correlation. To lessen its influence, (Prorok et al., 2012) changed the particle resampling algorithm. However, this study used a pre-determined sampling proportion to avoid measurement correlation. Because natural terrain introduces such a wide range of correlations, relying on a pre-set heuristic does not guarantee that the filter will remain consistent. In the related subject of target tracking, (Sun et al., 2016a) employs a distributed particle filter and uses channel filters to account for cross-correlation explicitly. This strategy, however, has not been applied to the problem of localization, and channel filters are not well adapted to the intermittent communication of underwater environments since they must be reinitialized every time the network changes.

To the best of our knowledge, no previous work has met all the requirements for a robust collaborative localization technique for robots using CMRL. Therefore, we propose a particle filter-based implementation that can accommodate multi-modal CMRL estimates, explicitly accounting for cross-correlation to avoid over-convergence, with minimal computation and communication requirements which is related to the improved particle filter approach multi-sensor fusion approach (Khan et al., 2011). These advances allow practical implementation of the CMRL on mobile robots with robustness to network changes and communication loss.

For indoor robot localization, existing techniques can achieve high localization accuracy for wireless sensor networks (WSN) and utilizing Radio Signal Strength Indicator (RSSI) as a source to calculate bearing information (F. Liu et al., 2020b; Morales & Kassas, 2018; Owen-Hill et al., 2013). A performance comparison study (Alfurati & Rashid, 2018) also backs the application of wireless sensor for localization. In collaborative multi-robot localization (CMRL), previous research used probabilistic approaches in a centralized or decentralized manner through map merging algorithms. Those techniques work well in static

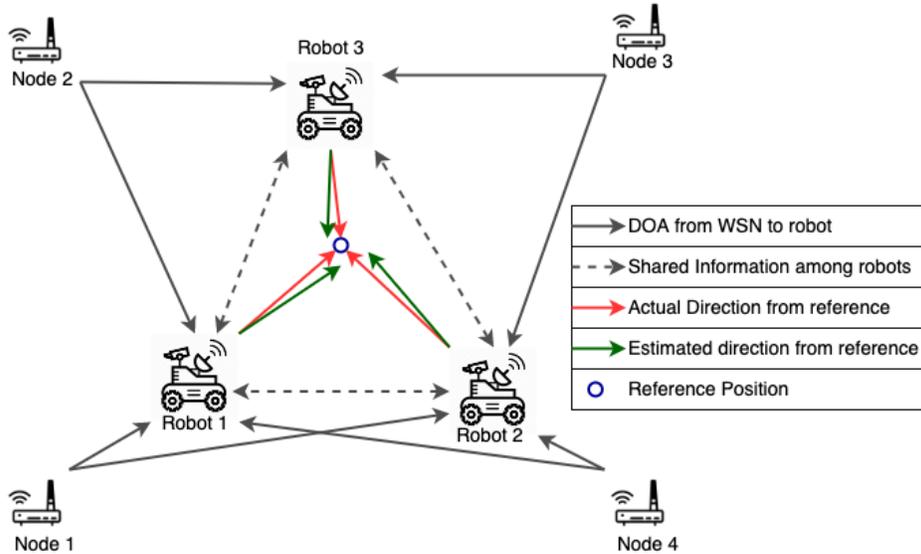


Figure 4.2: Overview of the proposed multi-robot synergistic localization, robots localizing themselves using local information as well as shared information.

environments, but not in highly dynamic and complicated environments like production halls (Sun et al., 2016b).

Approaches to addressing the localization problem in dynamic environments have a high trade-off between performance and efficiency. A recent attempt at CMRL incorporates filter architecture that allows multiple robots to collaboratively localize using Terrain Relative Navigation (TRN) (Wiktor & Rock, 2020). For shared information fusion to refine the localization estimation, an estimator structure that uses CI to reduce one source of measurement correlation while appropriately incorporating others has been used in contrast to current explicit modeling approaches. We aim to localize agents with high accuracy and efficiency in such an environment.

4.3 Proposed Approach

We look at the problem of a robot (or a wireless device) localization (self-localization) against its surroundings. We propose a combined approach for individual and synergistic localization (see Fig. 4.2). In the individual pose estimation, several WSNs are distributed in the environment, and the mobile robot is mounted with an AP, which nearby WSNs can sense. The WSNs measure the RSSI values coming from the AP and communicate this information to the robot (assuming

the measurements and shared data are reasonably time-synchronized). Furthermore, each robot also receives pose estimation and RSSI as shared information among the network from connected robots to improve localization accuracy. We apply the Bayesian rule to incorporate local and shared pose estimation.

Although the robot dynamics and sensors may differ, each robot’s surroundings are the same, and all agents are assumed to have a synchronized clock. It is also assumed that the robots have a sensor capable of making a (potentially noisy) measurement of the position of other agents and that the robots can transmit data to other agents. This work also assumes that the agents can uniquely identify other robots and perform the associated inter-robot measurement, which is demonstrated in (Bahr et al., 2009; S. Luo et al., 2019). The robots can operate asynchronously, and communication links can be established or lost at any time.

4.3.1 Individual Localization

Suppose that we have a wireless AP mounted on a mobile robot (whose location is to be estimated) and that four fixed WSNs $N = \{N_1, N_2, N_3, N_4\}$ are placed at the corners of the bounded region (see Fig. 4.2 for location reference). The WSNs measure the robot’s AP’s radio signal strength (RSS). Using these signal values, we can estimate the DOA of the robot’s vector concerning the WSN’s center. The robot R records its path along with the DOA measurements as the tuple: $m_l = \{x_l, y_l, DOA_l\}$, where (x_l, y_l) is the location of the robot at location l . The problem is to find the best estimate of the robot’s location $(x_R^*, y_R^*) \in \mathbb{R}^2$ which maximizes the probability of observing the measurement tuples when the robot is at the estimated location $P(x_R, y_R \mid m_l, m_{l-1}, \dots, m_{l-M})$, where M is the number of previous samples considered along the completed path trajectory so far, given that we employ an arbitrary method to estimate the DOA.

Our solution uses a particle filter-based localization of a mobile AP for individual localization. There are two main parts of individual localization: 1) cooperative DOA estimation; 2) DOA-based localization.

Cooperative Estimation of DOA

Cooperative localization can be accomplished with a network of three (or more) wireless nodes, where each node can sense the signal strength of the other node in the network. In our current implementation, the DOA of the mobile robot within the network is obtained from the geometric rule described in the central finite difference method (Parasuraman, Fabry, et al., 2013). The Alg. 4 gives an

overview of wireless network collaboration for determining DOA. All these computations are being performed by the moving node, which runs a centralized service and receives the RSSI information from all connected nodes that sense wireless signals independently in a synchronized manner.

RSSI can be modeled as a vector with two components, and the gradient with respect to the center of the robot can be represented as $\vec{g} = [g_x, g_y]$. One of the primary advantages of the central finite difference method is that it provides gradient estimation based on the received signal strength from geometrically oriented wireless nodes. It means that, after an appropriate gradient estimation, a receiver node (moving robot) can estimate the direction of arrival of signals based on the reference position, Which then further be used for position estimation using particle filtering. Using the central finite difference method (Caccamo et al., 2015; Parasuraman, Fabry, et al., 2013), the RSSI gradient can be calculated as follows:

$$g_x = \frac{S_{N_3} - S_{N_2}}{2\Delta_X} + \frac{S_{N_4} - S_{N_1}}{2\Delta_X} \quad (4.1)$$

$$g_y = \frac{S_{N_3} - S_{N_4}}{2\Delta_Y} + \frac{S_{N_2} - S_{N_1}}{2\Delta_Y} \quad (4.2)$$

Here, Δ_X is the distance between the wireless sensor's antennas along the x-axis, Δ_Y is the distance between the wireless sensor's antennas along the y-axis, and $S_{N_1}, S_{N_2}, S_{N_3}$, and S_{N_4} are the RSS values at nodes $N = \{N_1, N_2, \dots, N_4\}$, respectively on the mobile robot, measured at the current path location.

$$DOA_l = \arctan\left(\frac{g_y}{g_x}\right) \quad (4.3)$$

The formula provides the DOA of the wireless signal at a position along the path l using the RSS gradient. We can then suppress the noise of the calculated DOA by using the exponentially weighted moving normally. We consider a fixed window k of the previous positions of the robot along its path. In this way, the sampled \widehat{DOA}_l at a path area l with $K - 1$ past positions included in the window can be composed using moving average.

DOA-based Localization

Similar to (X. Li et al., 2015) for acoustic signals, we employ a Gaussian probability model on the wireless signal DOA estimates to calculate the weights of each random particle in the PF. This probabilistic model will weigh the quality of signals sensed by each node from N and ultimately produce an accurate robot location estimate through the PF.

Algorithm 4: Wireless sensor network (WSN) based cooperation for determination of DOA on a mobile robot.

```

1 for every time node  $N_i \in N$  WSNs do
2   | Each node  $N_i \in N$  measures RSSI ( $S_{N_i}$ ) locally at time  $t$ ;
3   | publish  $S_{N_i}$  and  $t$  to the topic  $r_i$  through a publish-subscribe
   | mechanism by all WSNs;
4 end
5 % At robot node;
6 Initialize time window to  $T$ ;
7 while time window is open do
8   | for every RSSI topic  $r_i \in R$  do
9   |   | extract  $S_{N_i}$  and  $t$  from  $r_i$ ;
10  |   | if  $t$  in time window then
11  |   |   | add  $S_{N_i}$  to  $S_i$ ;
12  |   |   end
13  |   end
14  | Average out all  $S_{N_i}$  in  $S_i$ ;
15 end
16 Calculate RSSI gradients: Eq. (4.1) and (4.2);
17 Calculated DOA of the robot within the WSN: Eq. (4.3).

```

The error between actual DOA (ADOA) for all wireless sensors at a potential candidate position l of mobile robot with coordinates (x_c, y_c) to the perceived \widetilde{DOA} values for each sensor, can be calculated as follows:

$$err_l^{ws} = \sum_{j=1}^N (ADOA_l^j - \widetilde{DOA}_l^j). \quad (4.4)$$

Now, we use the Gaussian probability formula (similar to (X. Li et al., 2015; Parashar & Parasuraman, 2020)) on this error to calculate the probability of the i_{th} candidate location of the particle $q_i = (x_i, y_i, w_i), i \in (1, ..N)$, where N is the number of particles in the PF and w_i is the weight of each particle calculated over a set of previous path samples as:

$$w_i \propto P_l(q_i) = \prod_{k=0}^{M-1} \left[\frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(err_l^{t-k})^2}{2\sigma^2}} \right] \quad (4.5)$$

There is an intrinsic angular inaccuracy in each DOA degree that is analyzed. σ represents this error's fluctuation (deviation), which is anticipated to

be known because we know the correctness of the technique used to assess the DOA. We use the product of the Gauss likelihood of DOA error over $M - 1$ prior robot positions (imitating geographically scattered samples) so that the sifted DOA from earlier path locations can be used in the same way as readings from many sensors.

This component of the DOA probability is used to calculate the weights of the particles in the PF, which is then employed in the resampling procedure in the next PF step (iteration). The particle filter provides initial hypotheses with a uniform sampling of probable robot locations across the environment using constraints around the present robot location. The Gaussian probability is determined for each particle, that is, the signal source.

Algorithm 5: Multi Robot Synergistic Localization of a mobile robot using DOA from WSNs

```

1 Pr = [] % Initial List of Particles in the PF ;
2 while end of trajectory stream do
3   for r in R do
4     sample  $x_{r,t}$  from transition model  $P(X_{r,t+1}|x_{r,t})$ ;
5      $X_{r,t+1} = X_{r,t}$  % motion model for AP embedded robot;
6     W = [] % weights;
7     for  $x_{r,t+1}$  in Pr do
8       add  $P(e_{r,t+1}|x_{r,t+1})$  to W;
9     end
10    Rr = [] % re-sampled particle filter;
11    for  $n=2$  to N do
12      for  $j=1$  to n do
13        Find combined probability of  $P_l(q_i)$  and  $P(r_j)$  as
14          (Eq. (4.8));
15        update  $w_i(x_i)$  as (Eq. (4.9));
16        add P to Rr;
17      end
18    end
19    Pr = Rr;
20     $x_l^* = x_i \in w_i(x_i)$  is  $\max(w_i)$ ;
21 end

```

4.3.2 Synergistic Localization

Suppose we have n number of mobile robots, say $R = R_1, R_2, \dots, R_n$ collaborating in the same environment. All of them are connected to each through the wireless channel as the individual robot is equipped with a wireless transmitter and receiver.

Share information among robots I contains estimated position and probability calculated in the individual robot location process. Here, we extend the probability calculation process using Bayes's probability model by incorporating the shared information described below.

First, each robot calculates error for received and measured RSSI from neighboring robot j :

$$err_j^{RSSI} = RSSI_m - RSSI_r \quad (4.6)$$

where $RSSI_m$ is the measured RSSI and $RSSI_r$ is the received RSSI, we can calculate the $RSSI_m$ using path loss model.

$$RSSI_m = A - 10 \times n \times \log_{10}(distance(R_i, R_j)) \quad (4.7)$$

Here, n denotes the signal propagation exponent, which varies between 2 (free space) and 6 (complex indoor environment), d denotes the distance between Robot R_i to the Robot R_j , and A denotes received signal strength at a reference distance of one meter.

To get the weight for the probability based on the shared information from all neighboring robots, we can use the Gaussian probability formula same as (Eq. (4.5)) with $RSSI$ as an error.

We can further incorporate this probability $P(r_i)$ for robot r_i , to update the weight in Algorithm 5 while calculating the probability as:

$$w_i = P_l(q_i) \times \prod_{j=1}^n \left(P(r_j|x_j) \times P(x_j) \right). \quad (4.8)$$

Using a constraint around the present robot location, the particle filter provides initial hypotheses with a uniform sampling of probable robot locations across the environment. For each unique particle, that is, the signal source, the Gaussian probability is determined. The particles are subsequently given weights that are proportional to their likelihood, and the weights w_i are normalized as:

$$w_i^*(q_i) = \frac{w_i(q_i)}{\sum_{i=0}^{N-1} w_i(q_i)}. \quad (4.9)$$

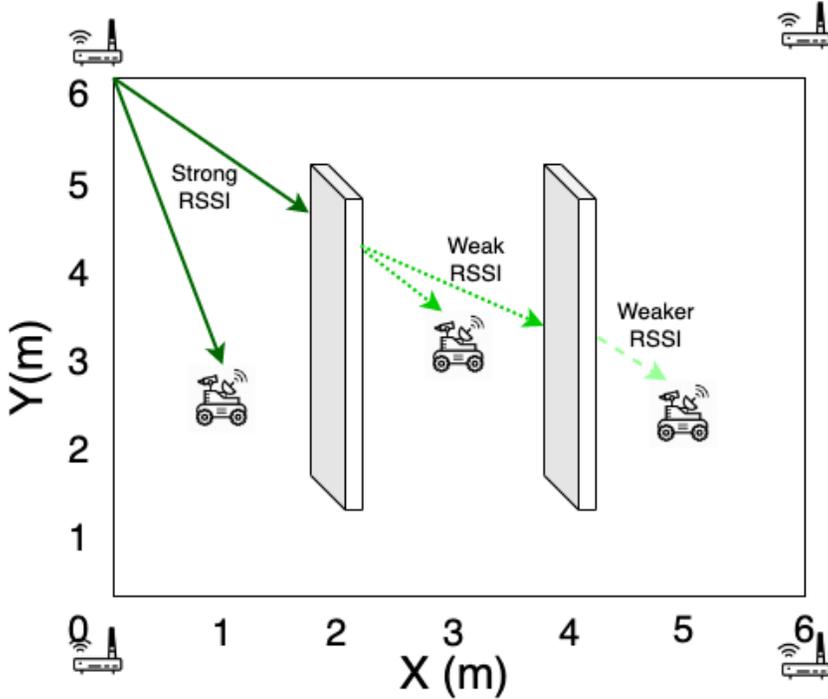


Figure 4.3: The simulation setup for three robots in a 6m x 6m workspace is shown here. It also depicts the two virtual walls in the simulation workspace to create non-line-of-sight conditions. Each wall will attenuate 10 dBm of RSSI signal power.

This normalized weight determines the likelihood of regenerating a set of particles in the next iteration. The particle with the highest weight (softmax) is the mobile robot’s most recent best pose estimate. This process repeats until either one unique particle remains or no modern tests are available. The PF iterates for each new estimation tuple. Algorithm. 5 provides an overview of the MRSL.

The MRSL has an upper bound of complexity as $O(nc)$ for c particle filters and n number of connected robots. Since c is a constant, the algorithm runs linearly and scales linearly with the number of robots.

4.4 Experimentation and Results

We have performed extensive simulation experiments in a 6 x 6 meters bounded region under different scenarios to analyze the algorithm. We assumed that the infrastructure information (position of the WSN nodes) was available to all

robots. The communication between connected nodes is asynchronous, i.e., the robot only updates its position estimation when it receives pose estimates from other robots. The temporal correlation among received information manages through union covariance and Bayesian rule incorporation. In all the experiments, the robots start at a random location in the workspace and move with a random walk trajectory.

Several studies have shown a range of 40 m is optimum to receive a good RSSI signal with high link quality (Dong & Dargie, 2012b; Parasuraman, Kershaw, et al., 2013). Therefore, we estimate that in our wireless sensor networks testbed, the nodes can be spaced with a diagonal range of up to 40 m for optimum performance in ideal conditions. This means, for a four-corner node setup as in our simulation setup in Fig. 4.3, the algorithm can handle an area of 800 m^2 . The nodes can be static or mobile, depending on the application requirements. We can increase the number of WSN nodes accordingly for a larger area. In reality, this range can be further limited due to non-line-of-sight conditions and packet drops.

We have compared our approach with the following benchmarks from the literature: 1) individual non-collaborative autonomous robot localization (ARL) with bearing (DOA) sensors (Parashar & Parasuraman, 2020) that use the WSN nodes (without inter-robot collaboration) 2) Collaborative Terrain Relative Navigation (CTRN) from (Wiktor & Rock, 2020). We implemented the proposed MRSL algorithm with the distance and RSSI sensor variants.

We have designed three sets of experiments to validate the scalability, robustness, and fault-tolerance of MRSL:

1. Number of robots varying from 1 to 100;
2. Non-line-of-site environment with (virtual) walls generated in simulations;
3. Communication challenges simulated through a fixed Packet Drop Ratio (PDR).

Finally, we have measured the accuracy in terms of the root mean squared error (RMSE) compared to the ground truth location data from our simulations. We obtained results by performing 100 trials for each experiment set. The results shown are averaged over all these trials.

4.4.1 Scalability and Dense Cooperation

First, we obtain the level of improvement the proposed MRSL algorithm provides when the number of robots varies from one to three in the multi-robot

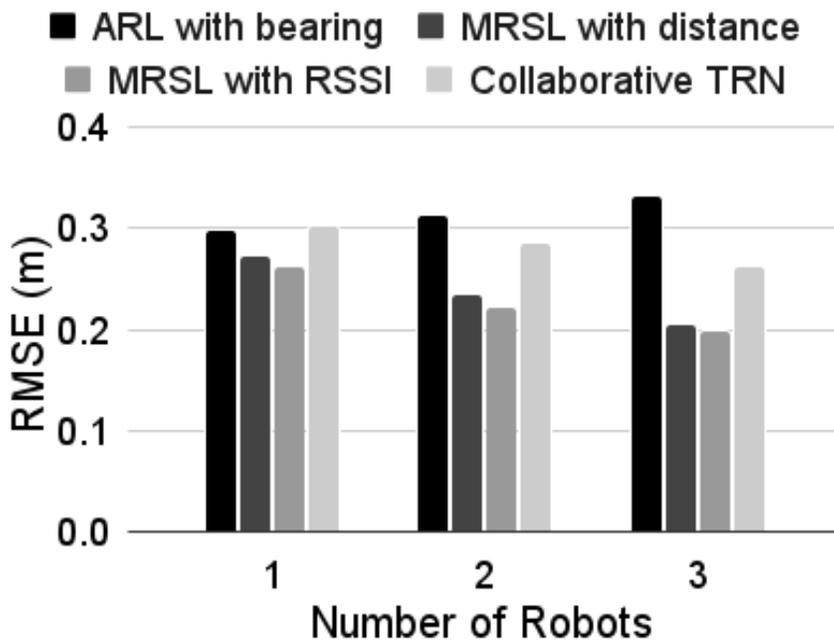


Figure 4.4: Localization performance of different algorithms regarding Root Mean Squared Error (RMSE) compared to the ground truth location in the 6m x 6m simulation experiments.

team, with four fixed wireless sensor network anchor stations. Fig. 4.4 shows the results of this experiment. It shows the superior performance of the proposed MRSL variants compared to the benchmark. Also, we can see improvement in localization accuracy with more robots in the system. This shows the effectiveness of the synergy achieved by the multi-robot team using the proposed collaborative scheme.

In Fig. 4.5, we can see the performance of the localization in detail by comparing the trajectory generated by different multi-robot localization algorithms.

Next, we obtain more information regarding the algorithm’s scalability and performance with many agents deployed with high density in a small environment. In this case, we evaluate the scalability of MRSL by increasing the number of robots from 1 to 100 with an increment of 2. Experimentation was performed on the same bounded region, which uses four WSNs distributed on the corners of the simulation workspace, with the initial position of the robots to be known with some uncertainty (Fig. 4.2 provides the simulation workspace setup).

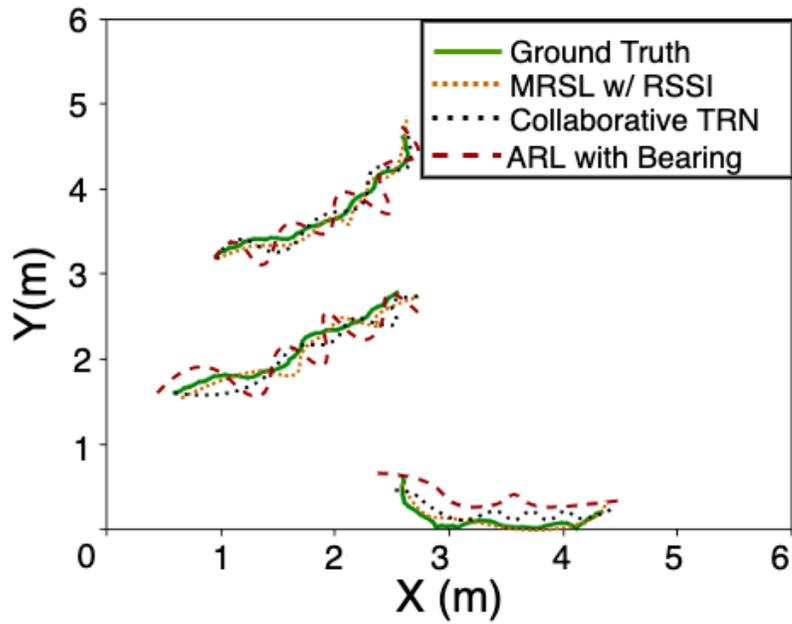


Figure 4.5: Trajectory of three cooperative robots in the 6m x 6m simulation experiments are shown here. The outputs of localization algorithms are plotted to comparatively observe their performance against the ground truth.

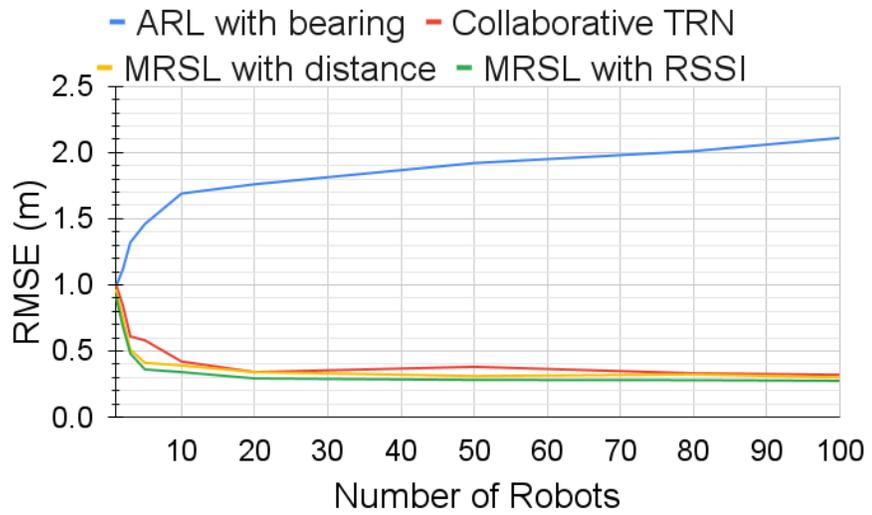


Figure 4.6: Localization accuracy comparison for varying number of robots

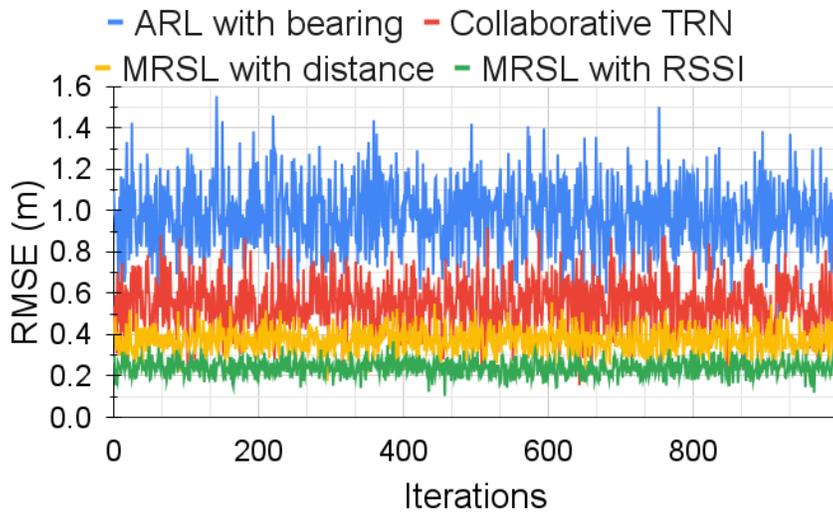


Figure 4.7: Localization accuracy comparison for non-line-of-site scenario

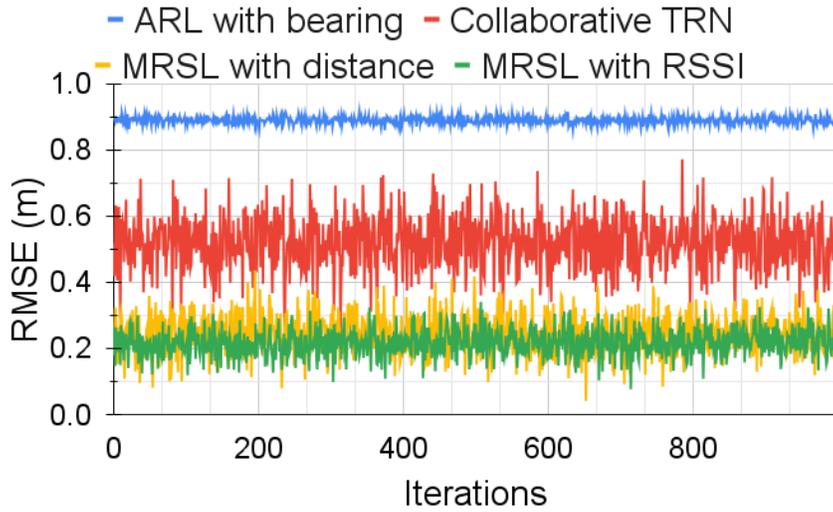


Figure 4.8: Localization accuracy comparison for communication challenge: 70% Packet Drops

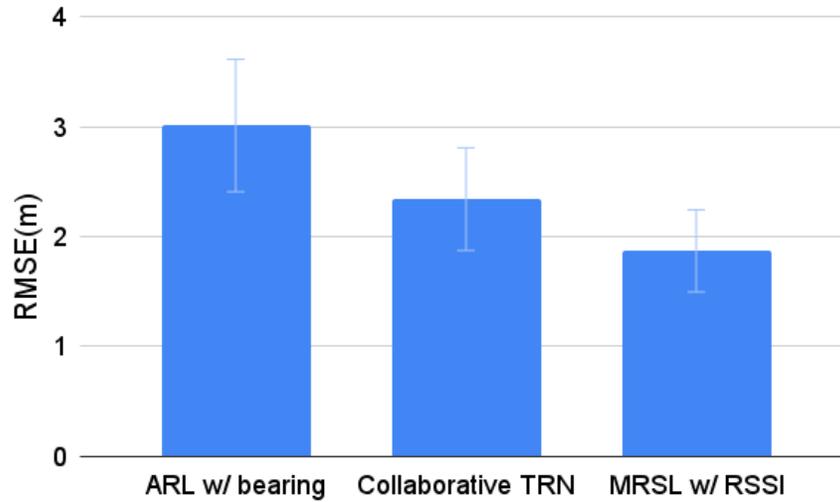


Figure 4.9: Localization performance of different algorithms regarding Root Mean Squared Error with three robots in the 60m x 60m simulation workspace.

With an increasing number of robots from 1 to 10, We have observed 35% reduction in pose estimation error, and afterward, the error curve turned into a plateau. ARL has shown an exponential increase in error by increasing the number of robots; however, CTRN has shown similar behaviors but less accuracy than MRSL, which can be seen in Fig. 4.6. This plateau behavior depicts the convergence for localization accuracy in multi-robotic systems and scalability with meter-level accuracy. Results also provide evidence for the sufficiency of four wireless nodes accessible to each node for multi-robot localization in a large region containing many collaborating robots.

In addition, we performed experiments with a larger simulation workspace area to test the practicality of MRSL in larger environments and compare its localization accuracy with other approaches. A testbed of 60 m \times 60 m has been set up for simulation with three connected moving robots with varying linear and angular velocities while considering standard dynamics and physical properties. We performed ten trials with the same simulation settings and recorded the RMSE in meters to analyze the localization accuracy. Results in Fig. 4.9 have validated the claim about the practicality of MRSL for larger spaces, with a 40% and 21% high localization accuracy than ARL and TRN, respectively. A standalone view of the localization of MRSL is also promising, with the absolute 1.8 meters of localization error for a 3600m² region, which adds more evidence for the practicality of MRSL for real-world systems.

4.4.2 Dynamic Environments and Non-Line-Of-Site Conditions

To simulate dynamic and non-line-of-sight (NLOS) environmental conditions, we added two objects in the environment: fixed (virtual) walls in the simulated region (shown in Fig. 4.3); and dynamic non-cooperative robots acting as random obstacles. Here, three non-cooperating robots executing random walks were deployed on the testbed to simulate the dynamic objects. We used three cooperating robots ($N = 3$) in these experiments. The wall or robot obstacle will attenuate the RSSI a further 10 dBm if the objects come in between two cooperating robots' direct line of sight or the WSN nodes during communication.

Fig. 4.7 shows the results in the NLOS conditions. We can observe consistent meter-level accuracy even in the reduced line of sight environment for MRSL; however, ARL and CTRN 50% higher pose estimation error. Hence, the proposed MRSL algorithm delineates the robustness of localization in a dynamic environment.

4.4.3 Robustness to Communication Challenges

Communication difficulty is simulated through randomly dropping packets with 70% probability, i.e., a Packet Drop Ratio (PDR) of 70%. When packets are dropped, the robots cannot receive information from all robots in the environment. Five robots ($N = 5$) were simulated for these experiments and randomly dropped shared information packets with a fixed PDR to examine the fault tolerance of MRSL.

Fig. 4.8 presents the results with minor fluctuation in RMSE for MRSL. Nevertheless, MRSL is reliable while keeping the RMSE in a range. ARL was indifferent to the communication challenge, as there was no coordination among robots. However, CTRN has shown more reduction in localization accuracy than MRSL.

4.5 Summary

This chapter discusses a novel method for robot localization in dynamic and dense environments, focusing on multi-robot systems (MRS). It acknowledges the challenges of existing solutions such as SLAM and GNSS-denied localization, particularly in highly dynamic environments like warehouses and logistics centers. The proposed solution, Multi-Robot Synergistic Localization

(MRSL), is a decentralized system where robots update their position estimation based on shared local information from neighboring robots. The system uses a Bayesian rule for shared information integration, proving to be computationally efficient and well-suited to asynchronous robotics communication. Each robot in the MRS can independently localize itself, updating its position estimation whenever new information is received from its connected neighbors. This MRSL approach extends the particle filter localization algorithm, enabling robots to utilize the Radio Signal Strength Indicator (RSSI) as a source for bearing calculations.

The MRSL algorithm has been tested through extensive numerical simulations, with results showing a significant increase in localization accuracy compared to other methods, outperforming autonomous robot localization (ARL) and other collaborative multi-robot localization algorithms. The simulations also highlighted the need for real-world experiments. The work discussed in this chapter relied on the infrastructure information for global localization; to overcome this limitation, we exploited the graph optimization for relative localization, discussed in the next chapter.

CHAPTER 5

DGORL: DISTRIBUTED GRAPH OPTIMIZATION BASED RELATIVE LOCALIZATION OF MULTI-ROBOT SYSTEMS

5.1 Introduction

This chapter discusses a graph optimization-based strategy to overcome the limitation of infrastructure information of the global localization solution discussed in the previous section. The estimation of a relative pose, including position and orientation, (Islam et al., 2021), for multi-robot systems (MRS) (Xianjia et al., 2021) is the foundation for higher-level tasks like collision avoidance, cooperative transportation, and object manipulation. Motion capture systems (Najafi et al., 2019), ultra-wideband (UWB) systems with anchors, and RTK-GPS systems are a few examples of established multi-robot relative positioning solutions that currently rely on the deployment of physical anchor or base stations in the application. These plans, however, are not suitable for large areas or interior settings where it is difficult to convert the infrastructure, which limits the overall performance and application possibilities of multi-robot systems and makes their use more difficult and expensive. Furthermore, extraction of range and bearing measurements from cameras and visual markers, while another practical approach, has the drawbacks of having a small field of vision, a short-range, obscured by nearby objects, and maybe requiring much computational power. The use of distance measurements from sensors like radars, Lidars, and UWB to achieve relative localization, on the other hand, has recently attracted more significant interest.

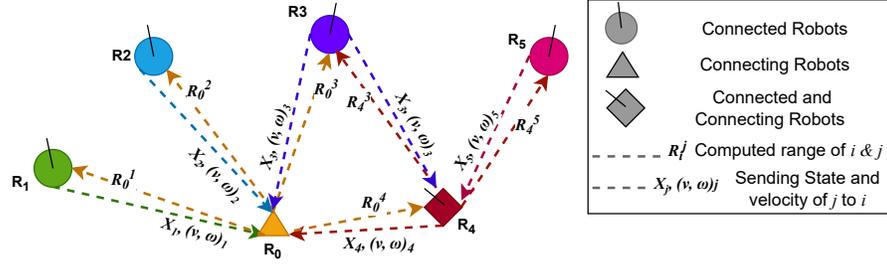


Figure 5.1: Overview of configuration space of a multi-robot system, sharing their pose (x_i) and relative range (R_i^j) measurements in our DGORL solution.

The multi-robot relative localization (MRL) problem, which refers to detecting and locating the relative configurations of mobile agents (typically with fewer sensor data such as relative range or bearing) concerning other agents or landmarks, is critical in MRS because it is required for robot teaming and swarming (Fink et al., 2012; Guo et al., 2017). As a result, many applications are frequently confronted with the relative localization problem, including formation control, cooperative transportation, perimeter surveillance, area coverage, and situational awareness. Relative localization and mapping (aka multi-robot SLAM) is an extension of the MRL problem. While several researchers have proposed novel solutions to the multi-robot map merging problem using pose graph matching and optimization techniques, they rely on extensive sensor data inputs (such as point clouds or Lidar scans) (Dube et al., 2017; Mangelson et al., 2018; Tian et al., 2022). Therefore, solving the MRL problem with relative range or bearing in a distributed manner is desirable and scalable in MRS (Latif & Parasuraman, 2022b).

Distributed optimization is the problem of minimizing a joint objective function that is the sum of many local objective functions, each corresponding to a computer node. We can model many fundamental activities in this area as distributed optimization problems, which have significant implications for multi-robot systems. Examples include cooperative estimation (Shorinwa et al., 2020), multiagent learning (Wai et al., 2018), and collaborative motion planning. The distributed optimization formulation provides a versatile and effective paradigm for creating algorithms for numerous multi-robot problems.

Each of these problems can be represented by a graph. Each node in the graph represents a state variable that needs to be optimized, whereas each edge linking two variables is a paired observation of the two nodes it connects. In the literature, various tactics have been proposed to address this set of prob-

lems. A straightforward implementation using well-known methods like Gauss-Newton, Levenberg-Marquardt (LM), Gauss-Seidel relaxation, or different iterations of gradient descent typically produces satisfactory results for most applications. However, extensive effort and subject-matter knowledge are required to achieve the highest outcome.

In consumer electronics, Wi-Fi is one of the most extensively utilized wireless technology for indoor wireless networks. The ubiquitous availability of Received Signal Strength Indicator (RSSI) measurement on such inexpensive commercial devices is the RSSI measured from an Access Point (AP) or a Wireless Sensor Robot (WSN). The RSSI value can be used in various applications, including relative localization (Latif & Parasuraman, 2023b; Parashar & Parasuraman, 2020; W. Wang et al., 2019), cooperative control (S. Luo et al., 2019; Parasuraman & Min, 2019), and communication optimization (Parasuraman et al., 2018).

In this paper, we formulate the MRL problem as a graph optimization problem and solve it in a distributed manner using a polynomial-time optimizer called the General Graph Optimization (g^2o (Kummerle et al., 2011)). g^2o is an open-source graph-based framework to handle the nonlinear error problems and is used to optimize global measurement pose using the initial global measurement poses and local relative pose constraints.

Our solution, termed DGORL, aims to achieve high localization accuracy efficiently in a distributed fashion. DGORL forms relative position-weighted connectivity graphs using RSSI as local sensor data then expands graphs based on possible positions at an instant and further optimizes to fetch relative position estimates for all connected robots. See Fig. 5.1 for an overview of the configuration space of DGORL.

The main contributions of this paper are listed below.

1. A novel distributed, efficient, and precise relative localization system based on shared inertial measurements and RSSI inputs from connected robots.
2. Position-weighted connectivity graph construction and optimization strategy tailored specifically for obtaining reliable relative pose estimates.
3. Theoretical and numerical analysis to evaluate the performance of the algorithm.
4. Validation of accuracy and efficiency of the DGORL compared to the recent collaborative multi-robot localization algorithm (Wiktor & Rock, 2020), which used covariance intersection technique to address the temporal correlation between received signals.

⁵ <http://github.com/herolab-uga/DGORL>

5. Open-sourcing of the codes⁵ for use and improvement by the research community.

5.2 Related Work

Most recent solutions to the simultaneous localization and mapping (SLAM) and MRL problem are based on graph optimization (i.e., all robot poses and landmark positions compose the graph's nodes, while each edge encodes a measurement constraint) (Kummerle et al., 2011). A conventional graph formulation, on the other hand, may suffer from unbounded processing and memory complexity, which might constantly expand over time. This is because new robot poses (and new landmarks in the case of feature-based SLAM) are constantly being added to the graph, resulting in an increase in the number of nodes over time; additionally, if frequent loop-closing events occur in SLAM, loop-closure constraints (edges) can significantly increase the graph density (Huang et al., 2013). For example, this could be the case if a service robot works for an extended time inside an office building.

Particularly, graph optimization and factoring have been recently proposed in the literature to solve different variants of the MRL problem (Hao et al., 2022; Sahawneh & Brink, 2017; Zheng et al., 2022). Even though the issue of reducing the complexity of graph optimization has recently been addressed (Carlevaris-Bianco & Eustice, 2013; Johannsson et al., 2013), to the best of our knowledge, little work has yet explicitly taken into account estimation consistency (i.e., unbiased estimates and an estimated covariance more significant than or equal to the actual covariance (Bar-Shalom et al., 2004)) in the design of graph reduction (sparsification) schemes. This is a critical flaw because if an estimator is inconsistent, the accuracy of the derived state estimations is unclear, making it untrustworthy (Indelman et al., 2014). Moreover, the performance and efficiency of approaches to the localization problem in dynamic environments are significantly traded off.

Most cooperative localization methods entail robot communication and observation, which makes any step prone to inaccuracy. In a recent attempt at multi-robot localization, many robots can locate themselves jointly using Terrain Relative Navigation (TRN) (Wiktor & Rock, 2020). The localization estimation utilizing shared information fusion has been improved by using an estimator structure that takes advantage of covariance intersection (CI) to reduce one source of measurement correlation while properly including others. Similarly, a work (Chang et al., 2021) developed a CI-based localization method with an explicit communication update and guaranteed estimation consistency

simultaneously would increase the robustness of multi-robot cooperative localization methods in a dispersed setting. However, robots keep a system-wide estimate in their algorithm, which relative observations can instantly update. Consequently, it increases the computational complexity at a centralized server and over-burdened the robot to keep track of dynamics for global positioning accurately. Unlike the explicit CI modeling methods, our objective is to localize the robot relative to other robots in a distributed fashion by utilizing the polynomial-time graph optimization technique with high accuracy and efficiency.

Therefore, this paper proposes a graph-based optimization algorithm to address the gaps mentioned above. To form a graph using shared RSSI information among robots, we employ a Relative Pose Measurement Graph (RPMG) using observability analysis (Hao et al., 2022). Once we have created a connected, reliable graph, we exploit particle filtering over the motion model to expand the graph based on mobility constraints. Finally, we construct a k-possible combination of graphs for each robot which needs to be optimized. We use the polynomial-time graph optimizer (g^2o (Kummerle et al., 2011)) for graph optimization with a distributed constraint optimization.

5.3 Problem Formulation and the Proposed DGORL Solution

An overview of the typical MRS components is given in this section, along with a thorough explanation of the suggested distributed graph optimization formulation.

Multi-Robot System Robotic members of an MRS are divided into disjoint (isolated/disconnected from others) and connected (operating collaboratively) robots, which can be either ground-based or aerial. The robot that enters the measuring range of the monitoring robot at any given moment is referred to as the observed robot. A robot that takes measurements of a random robot, among other robots, is the observation robot. The following qualities are presumptive for the considered MRS’s robotic members:

- Wireless communication is used to share information across the MRS.
- The watching robot can extract the neighboring robot’s relative range (e.g., through RSSI measurements) and can uniquely identify each robot in its field of view.

- While the observed robots have limited sensory and computational capabilities, the observing robot may use high-precision sensors to carry out its self-localization.
- We restrict the movement of the robots within a two-dimensional planar space.

Assume at a given time t , a team of robots contains $n \in \mathbb{N}$ connected robots can form a weighted undirected graph, denoted by $G = (V, E, A)$, of order n consists of a vertex set $V = \{v_1, \dots, v_n\}$, an undirected edge set $E \subseteq V \times V$ is a range between connected robots and an adjacency matrix $A = \{a_{ij}\}_{n \times n}$ with non-negative element $a_{ij} > 0$, if $(v_i, v_j) \in E$ and $a_{ij} = 0$ otherwise. An undirected edge e_{ij} in the graph G is denoted by the unordered pair of robots (v_i, v_j) , which means that robots v_i and v_j can exchange information with each other.

Here, we only consider the undirected graphs, indicating that the robots' communications are all bidirectional. Then, the connection weight between robots v_i and v_j in graph G satisfies $a_{ij} = a_{ji} > 0$ if they are connected; otherwise, $a_{ij} = a_{ji} = 0$. Without loss of generality, it is noted that $a_{ii} = 0$ indicates no self-connection in the graph. The degree of robot v_i is defined by $d(v_i) = \sum_{j=1}^n a_{ij}$ where $j \neq i$ and $i = 1, 2, \dots, n$. The Laplacian matrix of the graph G is defined as $L_n = D - A$, where D is the diagonal with $D = \text{diag}\{d(v_1), d(v_2), \dots, d(v_n)\}$. If the graph G is undirected, L_n is symmetric and positive semi-definite. A path between robots v_i and v_j in a graph G is a sequence of edges $(v_i, v_{i1}), (v_{i1}, v_{i2}), \dots, (v_{ik}, v_j)$ in the graph with distinct robots $v_{il} \in V$. An undirected graph G is connected if a path exists between any pair of distinct robots v_i and v_j where $(i, j = 1, \dots, n)$.

In this paper, we formulated a multi-robot system as a graph problem to find the optimized solution for relative localization based on inequality constraints with a time-varying domain. Furthermore, we have segmented the solution into three different components, which can be solved simultaneously for every iteration over distributed constraints: 1) graph formation, 2) expansion through transition, and 3) optimization. See Fig. 5.2 for sequential procedure of DGORL.

5.3.1 Graph Formation

An undirected graph $G = (V, E)$, where $(i, j) \in E$ if and only if (i, j) and (j, i) both are in E , is called the underlying undirected graph of G . G is connected if its underlying undirected graph is connected. An isolated and connected subgraph of G is called a component. Suppose that an undirected graph G has

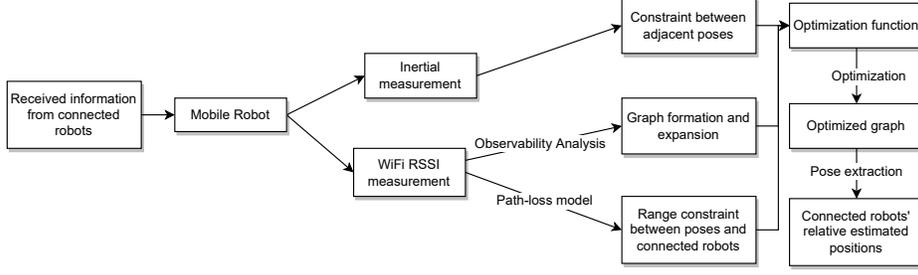


Figure 5.2: The Distributed Graph Optimization for Relative Localization of Multi-Robot Systems (DGORL) system architecture shows input (Received connected robot's motion information and RSSI) and output (relative pose estimation) for robot i . The robot generates and expands graphs based on observability for optimization along with the constraints set up through local and received inertial measurement, which further fed into the optimization function. The optimized graph yields relative pose estimates for all connected robots.

n nodes and m edges. The incidence matrix of G , denoted by $A(G)$, is an $m \times n$ matrix whose m rows and n columns correspond to the m edges and n nodes of G , respectively. Each element a_{ij} of $A(G)$ is defined as:

$$a_{ij} = \begin{cases} 1 & \text{if node } v_j \text{ is the tail/head of edge } e_i, \\ 0 & \text{otherwise} \end{cases}$$

The following lemma describes the relation between the rank of $A(G)$ and the connectivity of G .

Lemma 1. Let G be an undirected graph with n nodes and $A(G)$ be its incidence matrix. If G is composed of λ components, the rank of $A(G)$ satisfies

$$\text{rank}(A(G)) = n - \lambda \leq n - 1$$

The overall algorithm is mainly composed of two steps, as depicted in Algorithms 6 and 7, respectively. The first step is to generate the Estimated Relative Position Measurement Graph (ERPMG); $G_E = (V_E, E_E, A_E)$ based on the Relative Position Estimation Graph (RPMG); $G = (V, E, A)$, which can be built-up using received range information from connected robots of an n -robot system, which describes the relative position measurements among robots. In this step, the node-set V_E of the $ERPMG = G_E$ is initialized to be the same as that of the $RPMG = G$, and the edge set E_E of the $ERPMG = G_E$ is initialized to be empty. Then, add all relative position measurement edges of E into the

edge set E_E if edge e_{ij} satisfy the motion constraint; $J^{F_j}V_{B_j} \neq 0$. A concise description of this step is illustrated in Algorithm 6.

Algorithm 6: ERPMG Formation

```

1 Input: State  $x$ , input  $u$ , RPMG  $G = (V, E, A)$ ;
2 Output: ERPMG  $G_E$ ;
3 Initialize ERPMG  $G_E = (V_E, E_E)$  with  $V_E = V, E_E = \phi$ ;
4 for each node  $v_i \in V$  do
5   for each edge  $e_{ij} \in E$  do
6     if  $J^{F_j}V_{B_j} \neq 0$  then
7       add edge  $e_{ij}$  to  $E_E$ ;
8     end
9   end
10 end

```

The second step (See Alg. 7) is to examine the observability of the n-robot system according to the $ERPMG = G_E$ generated by Alg. 6. We initialize the incidence matrix $A(G_E)$ and the diagonal matrix L_n to be zero matrices, with their dimensions determined by the number of nodes and edges in G_E . Then construct the incidence matrix $A(G_E)$ and the diagonal matrix L_n . The incidence matrix $A(G_E)$ describes which nodes are connected by which edges and is constructed based on the topology of G_E . The diagonal matrix L_n describes the weight of the edges in G_E with the weight vector of each edge. The edges in $A(G_E)$ and L_n take the same order. Then, we can obtain the spectral matrix $C(G_E)$ based on $A(G_E)$ and L_n . The observability of the n-robot system can be determined by evaluating the rank of $C(G_E)$.

5.3.2 Expansion through Transition

Initial position and velocity constraints are known to each robot; hence locally constructed ERPMG tends to change based on relative motion in the network. We have limited the number of possible positions of an individual robot to k by exploiting the concept of particle filtering. The motion process is carried out in the 2-D state-space $X_{n,t}$ includes position $(x_{n,t}, y_{n,t})$ and orientation $\phi_{n,t}$. The robot model $f(*)$ can be written as:

$$\begin{aligned}
x_{n,t+1} &= x_{n,t} + v_{n,t}\Delta t \cos(\phi_{n,t}) \\
y_{n,t+1} &= y_{n,t} + v_{n,t}\Delta t \sin(\phi_{n,t}) \\
\phi_{n,t+1} &= \phi_{n,t} + \omega_{n,t}\Delta t
\end{aligned} \tag{5.1}$$

Algorithm 7: Observability Checking

```
1 Input:  $n = |V_E|, m = |E_E|$  ;
2 Initialize the incidence matrix:  $A(G_E) = m \times n$ ;
3 Initialize the diagonal matrix:  $L_n = n \times n$  ;
4  $k = 0$ ;
5 for each node  $v_i \in V_E$  do
6   | for each edge  $(i, j) \in E_E$  do
7     |    $k = k + 1$ ;
8     |    $A(G_E)[k, i] = 1$ ;
9     |    $W[3k - 2 : 3k, 3k - 2 : 3k] = \text{diag}(L_n((i, j)))$ ;
10    | end
11  end
12 if  $\text{rank}(C(G_E))$  is equal to  $4(n - 1)$  then
13   | return True;
14 end
15 else
16   | return False;
17 end
```

In the Eq. 5.1, $v_{n,t}$ and $\omega_{n,t}$ are the velocity and angular velocity of the robot n at time, t respectively. δt represents the time interval between two control inputs. Based on the robot model $f(*)$, the control input of the robot n at time t is defined as:

$$u_{n,t} = [v_{n,t}, \omega_{n,t}] \quad (5.2)$$

It is worth mentioning that Eq. 5.2 is the ideal kinematic model of the robot. Under pure rolling conditions, the robot's motion follows this equation. However, imperfections are unavoidable in real devices. Complex wheel-ground interactions and system noises cause disturbances to the robots. Moreover, these disturbances are modeled as Gaussian random variables, characterized by their mean and covariance matrices. Thus, the model of a given robot n at time t is defined as:

$$x_{n,t+1} = f(x_{n,t}, u_{n,t}) + \mathcal{N}_{\text{noise}_{n,t}} \quad (5.3)$$

Here, $x_{n,t} \in \mathbb{R}_{nx}$ and $u_{n,t} \in \mathbb{R}_{nu}$ are the state and control input of the robot n at the time, t respectively. $\mathcal{N}_{\text{noise}} \in \mathbb{R}_{nx}$ is an unknown disturbance with Gaussian probability distribution. Considering the disturbance level in the real environment, $\mathcal{N}_{n,t}$ noise in simulations is set to $\text{diag}(0.1 \text{ m}, 0.1 \text{ m}, 0.5 \text{ deg})$.

As each robot also receives RSSI from other robots, we can find the intersecting region as an area of interest to map the estimated relative position for each robot using the model and find k soft max out of them. Once we have k possible positions of each robot, we can generate $< n^k$ solvable graphs for optimization.

5.3.3 Optimization

Constraints

We are interested in solving the constrained convex optimization problem over a multi-robot network in a distributed fashion. More specifically, we consider a network with n robots, labeled by $V = \{1, 2, \dots, n\}$ and k possible connections to other robots. Every robot i has a local convex objective function and a global constraint set. The network cost function is given by:

$$\text{minimize } f(\mathbf{x}) = \sum_{i=1}^N f_i(\mathbf{x}) \text{ subject to } \mathbf{x} \in \mathcal{D} = \left\{ \mathbf{x} \in \mathbf{R}^k : c(\mathbf{x}) \leq 0 \right\}$$

Here, $x \in \mathbf{R}^k$ is a global decision vector; $f_i : \mathbf{R}^k \rightarrow \mathbf{R}$ is the convex objective function of robot i known only by robot i ; D is a bounded convex domain, which is (without loss of generality) characterized by an inequality constraint, i.e., $c(x) \leq 0$, where $c : \mathbf{R}^k \rightarrow \mathbf{R}$ is a convex constraint function. All the robots know it. We assume that D is contained in a Euclidean ball of radius R , that is:

$$\mathcal{D} \subseteq \mathcal{B} = \left\{ \mathbf{x} \in \mathbf{R}^k : \|\mathbf{x}\|_2 \leq R \right\}. \quad (5.5)$$

We also assume that there exists a point \hat{x} such that the inequality constraint is strictly feasible, i.e., $c(\hat{x}) < 0$. We introduce a regularized Lagrangian function to deal with the inequality constraint $c(x)$.

$$\begin{aligned} L(\mathbf{x}, \lambda) &= \sum_{i=1}^N f_i(\mathbf{x}) + \lambda N c(\mathbf{x}) - \frac{\gamma}{2} N \lambda^2 = \sum_{i=1}^N \left[f_i(\mathbf{x}) + \lambda c(\mathbf{x}) - \frac{\gamma}{2} \lambda^2 \right] \\ &= \sum_{i=1}^N L_i(\mathbf{x}, \lambda), \end{aligned} \quad (5.6)$$

where, we have replaced the inequality constraint $c(x) \leq 0$ with $N_c(x) \leq 0$, and $\gamma > 0$ is some parameter. It is noted that $L_i(x, \lambda)$ is known only by robot i .

Model

We consider a time-varying network model that has been widely considered in the literature (Huang et al., 2013; Jiang et al., 2022; Kummerle et al., 2011). The robots' connectivity at time t can be represented by an undirected graph $G(t) = (V, E(t), A(t))$, where $E(t)$ is the set of activated edges at time t , i.e., edge $(i, j) \in E(t)$ if and only if robot i can receive data from robot j , and we assign a weight $[A(t)]_{ij} > 0$ to the data on edge (i, j) at time k . Note that the set $E(t)$ includes self-edges (i, i) for all i . We make the following standard assumption on the graph $G(t)$.

Assumptions: The graph $G(t) = (V, E(t), A(t))$ satisfies the following.

1. For all $t \geq 0$, the weight matrix $A(t)$ is doubly stochastic.
2. There exists a positive scalar ξ such that $[A(t)]_{ii} \geq \xi$ for all i and $t \geq 0$, and $[A(t)]_{ij} \geq \xi$ if $[A(t)]_{ij} > 0$.
3. There exists an integer $T \geq 1$ such that the graph $(V, E(s_T) \cup \dots \cup E((s+1)T-1))$ is strongly connected for all $s \geq 0$.

Once the robot i model the network and constraints, g^2o (Kummerle et al., 2011) optimize the provided k-ERPMG to find the best possible position estimations for connected n robots in the form of an optimized graph $G_o = (V_o, E_o, A_o)$. V_o are the possible node positions concerning i , E_o contains the possible distance range to other nodes in terms of weighted edges, and A_o is an optimized adjacency matrix from i to every other node. Algorithm 8 describes the entire localization strategy for a given MRS, with a graph-based optimizer denoting the algorithm's three main iterative steps.

5.4 Theoretical Analysis

Graph optimization-based relative pose localization has three submodules; Formation, Expansion, and Optimization. The formation depends on the number of robots n , which causes the generation of ERPMG $G = (V, E, A)$ and expansion relies on the number of possible position instances, k , which produces n^k potential graphs to be optimized. As both initial steps can be serially done in polynomial time, one can deduce that the algorithm is polynomial and scalable with the number of robots. However, the optimization of n^k possible graphs using the g^2o optimizer needs to be analyzed for its NP-hardness.

NP-hardness: We will show that the problem of finding optimizers is NP-hard in general. An example shows that L1-optimality is NP-hard for non-submodular energies.

Algorithm 8: Relative Localization Based on Graph Optimization

```
1 At given time interval  $t$ , for robot  $i$ ;  
2 Observe connected robots  $n$ ;  
3 Receive motion and observability information from connected robots;  
4 Generate ERPMG from RPMG using Alg. 6;  
5 Check observability of  $n$ -robot systems using Alg. 7;  
6 for every sampling instance  $t_k, k = 0, 1, 2, \dots$  do  
7   for each connected robot  $R_j, j \neq i$  and  $j = 1, 2, \dots, n$  do  
8     Get previous odometry information;  
9     Get previous relative positioning information;  
10    Generate  $n^k$  possible graph over estimation horizon;  
11  end  
12 end  
13 Set constraints as Eq. 5.4;  
14 Generate a time-varying network model for  $n$  connected robots;  
15 Solve the graph optimization problem using distributed solver  
    (Kummerle et al., 2011) over a predicted horizon;  
16 for each connected robot  $R_j, j \neq i$  and  $j = 1, 2, \dots, n$  do  
17   Extract position estimation from optimized graph:  
     $X_{j,t+1} = (x_{j,t+1}, y_{j,t+1}, \phi_{j,t+1})$ ;  
18 end
```

Remember that if no two vertices of a graph $G = (V, E, A)$ are connected by an edge, the set U of vertices is independent. The problem of finding the maximal independent set of vertices of an arbitrary graph is known to be NP-hard (Cormen et al., 2022). Consider the following local costs as an example:

- Give each vertex v of label l the cost l_i .
- For each edge with both vertices of label l , let the cost be $N = |V| + l$.
- Connect the cost of 0 to any other edge.

It is worth noting that if and only if the set $U = l_1(1)$ is independent, the maximum cost of any labeling l is N . All labeling l associated with an independent U have a maximum cost of N . Furthermore, the labeling l is a strict minimizer when the number of cost 1 atom for U , which is $|V| - |U|$, is minimal, i.e., when the size of U is maximal. To put it another way, if we use the previously mentioned local cost assignments for a graph G , then l is a strict minimizer if and only if $U := l_1(1)$ is a maximal independent set of vertices. As a result,

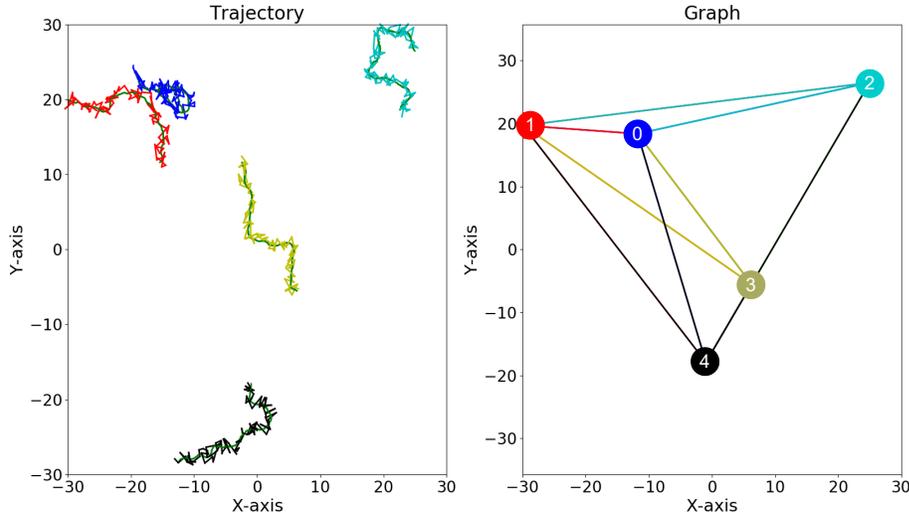


Figure 5.3: Trajectories and the Optimized Graph. **Left:** Simulation with the ground truth (green) trajectory along with the colored predicted trajectory of a robot; **Right:** Robots as colored vertices and colored edges between connected robots with the respective color for optimized graphs.

our problem, like the problem of finding the most extensive independent set of vertices, is NP-hard. Thus, the optimization problem is proved to be NP-hard, and we also know that such graph problems can be verified in polynomial time (Cormen et al., 2022) hence the problem is NP-complete.

Convergence: In order to control convergence, the g^2o method adds a damping factor and backup operations to Gauss-Newton, for which g^2o solves a damped version of the optimization function. A damping factor λ is present here; the more significant it is, the smaller the λ is. In the case of nonlinear surfaces, this helps control the step size. The g^2o algorithm's goal is to regulate the damping factor dynamically. The error of the new setup is tracked throughout each iteration. The following iteration is reduced if the new error is smaller than the prior one. If not, lambda is raised, and the solution is reversed. We refer to (Kummerle et al., 2011) for a more thorough discussion of how the g^2o algorithm guarantees convergence.

5.5 Simulation Experiments and Results

We performed extensive simulation experiments in a 60 x 60 meters bounded region under certain conditions to analyze the algorithm. We have set up the

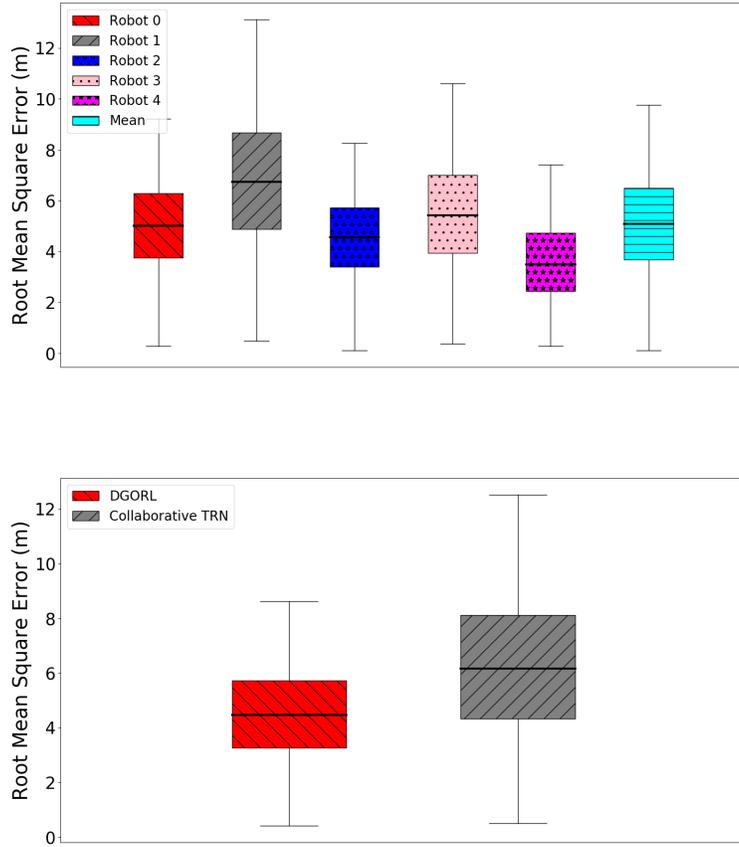


Figure 5.4: DGORL performance of five robots in terms of Root Mean Squared Error (RMSE) in the 60m x 60m simulation workspace (**Top**). Comparison with Collaborative Terrain Relative Navigation (TRN) based on Covariance Intersection (**Bottom**).

MRS as discussed in Sec. 5.3, where each robot shares inertial measurement and RSSI with connected robots. Each robot in the MRS performs relative localization based on its frame of reference. we can calculate weights of edges e_{ij} in $G = (V, E)$ as a range d the between R_i and R_j using path loss model using perceived RSSI:

$$d = 10^{\frac{A-RSSI}{10n}} \quad (5.7)$$

Here, n denotes the signal propagation exponent, which varies between 2 (free space) and 6 (complex indoor environment), d denotes the distance between robots i and j , and A denotes received signal strength at a reference distance of

one meter. On every iteration, r_i generates ERPMG based on its observation and performs observability analysis (at least four nodes connected, based on the observability Checking in Alg. 7). Later, it will expand ERPMG and generate input for the g^2o optimizer.

The initial global measuring posture, the local relative pose limitations, and its information matrix make up g^2o 's input. Once obtained an optimized graph, r_i estimated relative positions from vertex poses. Although in MRS, each robot performs localization distributedly. To measure error, we have captured initial positions on a global scale and compared predicted trajectories with ground truth as RMSE in meters. We used range sensors to compare our strategy to Terrain Relative Navigation (TRN) from (Wiktor & Rock, 2020).

Localization Accuracy We performed experiments with five robots driving them on a random walk for 100 iterations, and obtained consistent results through 10 repetitive trials under identical conditions. Fig. 5.3 visualizes the predicted trajectories and connected graphs for experimentation. Results in Fig. 5.4 have validated the claim about the accuracy of DGORL in a larger space, with 8% localization error.

Furthermore, experimentation was carried out in the same simulated workspace area to evaluate the localization accuracy of DGORL in contrast to TRN and to confirm its applicability in more significant scenarios. We performed ten attempts using identical simulation conditions and obtained the RMSE in meters to assess the localization accuracy. Results in Fig. 5.4, which demonstrate a 23% percent improvement in localization accuracy over TRN, have supported the assertion that DGORL is viable in larger contexts. Furthermore, the absolute localization error of 4.2 meters for a $3600m^2$ zone with five robots in a standalone view of the localization of DGORL is highly encouraging. Still, it needs further analysis to understand the limitations and sufficient conditions. Nevertheless, the results showcase the viability of DGORL for MRS applications.

Computational Demand The mean optimization time (referring to the time taken by the solver) and the mean CPU usage of the whole process are $10.2 \pm 2.7ms$ and $143 \pm 49\%$, respectively. The results are evaluated for five robots from 10 runs over 100 iterations. The computational complexity of covariance intersection-based TRN is close to DGORL, which further highlights the efficiency of DGORL. It is worth noticing that the performance of DGORL provides evidence for its practical use in small resource-constrained robots (i.e., equipped with tiny computers, e.g., Raspberry Pi zero with Quad

Core running at 1 GHz). Hence, DGORL can be used as an efficient relative localization algorithm for most multi-robot systems, including swarm robotics.

5.6 Summary

This chapter presents a novel graph-theoretic approach to solving the multi-robot relative localization problem, which is pivotal for collision avoidance, cooperative transportation, and object manipulation in multi-robot systems (MRS). The researchers seek to address limitations in current MRS solutions, which are unsuitable for large or indoor areas due to reliance on physical anchors or base stations, as well as limitations of the camera and sensor-based approaches. The solution, Distributed Graph Optimization for Relative Localization (DGORL), uses received signal strength indicator (RSSI) measurements, commonly found in Wi-Fi-enabled devices, as local sensor data to form relative position-weighted connectivity graphs. The process involves three main steps: formation of the connectivity graph, graph expansion through applying a motion model, and optimization of the graph to extract relative position estimates for connected robots.

The method is theoretically and numerically analyzed, with results indicating that the proposed DGORL algorithm demonstrates practicality and superiority in accuracy over a recent state-of-the-art collaborative localization algorithm while also maintaining comparable optimization time. Moreover, the approach can be applied in a distributed manner, making it scalable for multi-robot systems. The authors also open-source their codes for further use and development by the research community. Future work will focus on conducting hardware experimentation to confirm the real-world applicability of the proposed graph optimization approach. The computational complexity is still a concern for small robots. It can be improved using the Gaussian process regression model and vector transformation for relative localization, discussed in the next chapter.

CHAPTER 6

GPRL: MULTI-ROBOT RELATIVE LOCALIZATION USING HIERARCHICAL GAUSSIAN PROCESSES INFERRING ON RSSI MAP OF THE WIRELESS ACCESS POINT

6.1 Introduction

Multi-robot systems (MRS) have recently drawn significant attention for various use cases, including logistics, surveillance, and search and rescue. In GPS-denied environments or applications where the privacy of absolute (global) location must be protected, using the robot’s relative position to other robots or environment markers is essential as the robots need to cooperate, share data, and complete jobs effectively based on their relative localization. While the progress in simultaneous localization and mapping (SLAM) techniques has reached a significant research maturity (Cadena et al., [2016](#); Tian et al., [2022](#)), they rely on computationally-expensive sensors such as RGB-D cameras and LIDARs. The research challenges are even more prominent for resource-constrained robots, which have limited computation and sensing capabilities. In fact, relative localization without the need for intensive SLAM-based mapping can be sufficient to perform major cooperative multi-robot tasks like rendezvous, formation con-

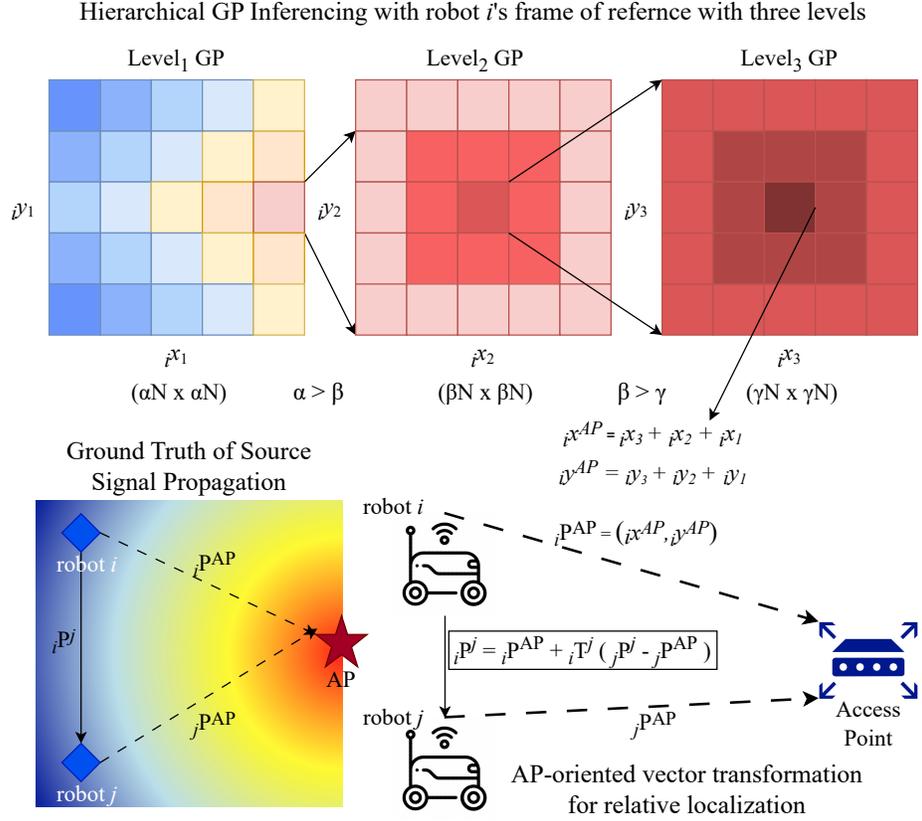


Figure 6.1: An overview of GPRL with hierarchical inferencing with three levels and relative localization with vector transformation using source position. α , β and γ are resolution parameters for $N \times N$ grid space, i^p^{AP} and j^p^{AP} are position vectors from the robot's position to the predicted AP position, and i^p^j is a relative position vector from robot i to j .

trol, etc. (S. Chen et al., 2022; Parasuraman & Min, 2019). Therefore, we focus on alternative sensor modalities such as UWB and Wi-Fi for multi-robot relative localization without environmental mapping (Latif & Parasuraman, 2022a; Tardioli et al., 2010; Wiktor & Rock, 2020).

Due to their accessibility and ubiquity, Wi-Fi signals present a promising data supply for relative localization tasks. The distance between the robot and the Wi-Fi Access Point (AP) can be estimated using the Received Signal Strength Indicator (RSSI) of Wi-Fi signals (Parasuraman & Min, 2019). However, variables, including multipath fading, shadowing, and background noise, frequently impact RSSI-based localization. This drives the need for reliable and effective learning algorithms that can accurately obtain relative localization by utilizing the given RSSI data in the face of these difficulties.

In this paper, we propose a novel distributed algorithm that overcomes the drawbacks of existing approaches to relative localization. Our approach uses hierarchical inferencing on the Gaussian Processes regression (GPR (Quinonero-Candela & Rasmussen, 2005)) map of RSSI to accurately forecast the location of the Wi-Fi AP in its local frame of reference with high computational and real-time efficiency. Then, we apply an AP-oriented vector transformation process that allows robots to accurately localize against each other by transforming other robots' coordinates into their own reference frame. Fig. 6.1 delineates the high-level overview of the proposed GP-based Relative Localization (GPRL), illustrating the process behind the hierarchical inferencing for accurate source position prediction and using it for estimating the relative locations of robots.

The main contributions of GPRL are as follows:

- We propose hierarchical inferencing over GPR for efficient and accurate source position prediction.
- We propose a novel vector transformation mechanism to localize connected robots in a simplified learning fashion relatively.
- We theoretically analyze the approach to evaluate the efficiency and accuracy of GPRL.
- We extensively validate the accuracy and efficiency of our GPRL approach's performance of active learning for relative localization in simulated Robotarium⁶ world and compared against state-of-the-art localization approaches: state-of-the-art Terrain Relative Navigation (TRN) (Wiktor & Rock, 2020), Distributed Graph Optimization for Relative Localization (DGORL) (Latif & Parasuraman, 2022a), and Modified Error Gaussian Process Regression (MEGPR) (M. Xue et al., 2019).
- To further analyze the practicality of localization using real robots working on ROS.
- We open source⁷ our method as a ROS package for use and further development by the robotics community.

⁶ <https://www.robotarium.gatech.edu/>

⁷ <https://github.com/herolab-uga/gp-multi-robot-localization>

The attached video also demonstrates the proposed approach in real robots in a rendezvous application.

6.2 Related Work

In the literature, relative localization has received substantial study, and several multi-robot system concepts have been put forth. Relative localization and its

significance in collaborative robotic systems were thoroughly covered by Wanasinghe et al. (Wanasinghe et al., 2015) and Rone and Ben-Tzvi (Rone & Ben-Tzvi, 2013). These methods, however, frequently lacked the flexibility to adapt to changing settings. A Jacobian-free strategy for multi-robot relative localization was introduced by Wanasinghe et al. (Wanasinghe et al., 2014b), which increased processing efficiency but still had difficulties adjusting to changing settings. A low-cost embedded system for relative localization in robotic swarms was proposed by Faigl et al. (Faigl et al., 2013) to lower hardware costs. Still, the localization performance remained sensitive to environmental changes. Wanasinghe et al. (Wanasinghe et al., 2014a) investigated distributed collaborative localization, and Latif and Parasuraman (Latif & Parasuraman, 2022a) proposed DGORL, which concentrated on distributed graph optimization for multi-robot systems, addressing some scalability issues but still encountering difficulties in dynamic environments and high computational complexity. Wiktor and Rock (Wiktor & Rock, 2020) presented a Bayesian optimization based approach for collaborative multi-robot localization in natural terrain, but the method comes with a high complexity of information fusion and computational costs.

Add (N. Xu et al., 2014) - GP-Localize paper.

Approaches based on learning have been suggested to address relative localization issues and adapt to changing settings. Hsieh et al. (Hsieh et al., 2019) used deep learning for indoor localization using received signal intensity and channel state information to provide an adaptable solution. Abbas et al. (Abbas et al., 2019) introduced WiDeep, a WiFi-based indoor localization system using deep learning. Hoang et al. (Hoang et al., 2019) employed recurrent neural networks for accurate RSSI indoor localization, and Li et al. (S. Li et al., 2022) proposed self-supervised monocular multi-robot relative localization using efficient deep neural networks. Deep learning techniques, however, frequently need huge volumes of labeled data and may experience overfitting or generalization problems in situations with little data. Surveys on relative localization using machine learning techniques (S. Chen et al., 2022) for indoor positioning highlighted the potential of machine learning for localization and the need for more effective and reliable algorithms.

This paper proposes a novel relative localization strategy (GPRL) that addresses the limitations of optimization, learning, and GPR-based approaches by avoiding optimization and offline fingerprinting overhead. GPRL offers online learning relative localization for multi-robot systems by efficiently predicting the location of the Wi-Fi source while precisely combining GPR and hierarchical inferencing. In addition, we present a vector transformation method to convert the local positions of other robots into the robot's local frame of

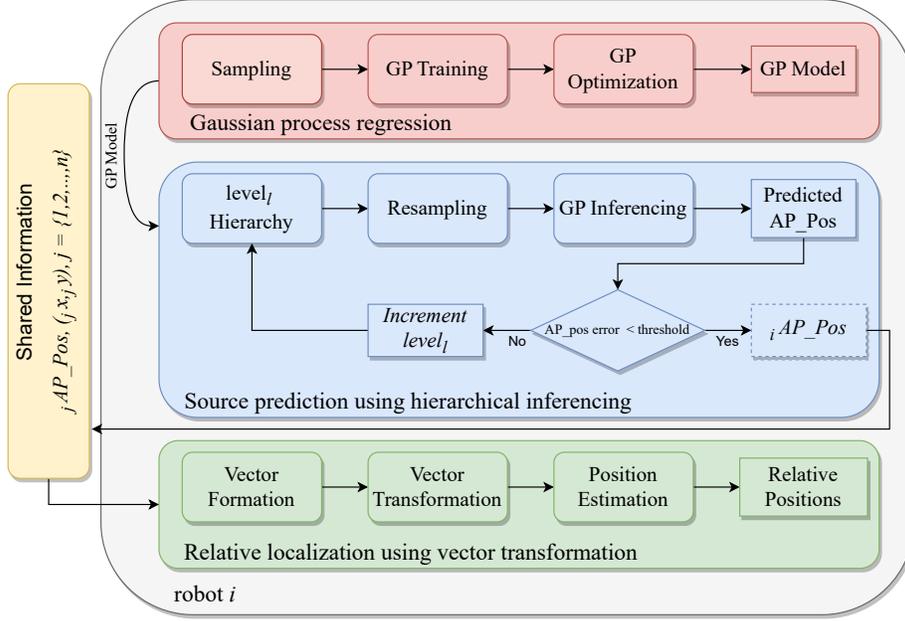


Figure 6.2: A distributed system architecture of GPRL for robot i to perform relative localization using GPR and hierarchical GP inferencing.

reference, enabling precise relative localization. GPRL provides a reliable and efficient learning-based solution for relative localization using sensory data, departing from the limitations of existing state-of-the-art approaches.

6.3 Problem Formulation

Given a set of N robots observing a single Wi-Fi source, the goal is to develop a Gaussian Process (GP) based learning method to perform relative localization among the robots. Each robot i receives a set of measurements, including the Received Signal Strength Indicator (RSSI) from the Wi-Fi source, its known initial orientation ${}_i\theta$, and the shared predicted access point (AP) position ${}_i\mathbf{p}^{AP}$ along with its position ${}_i\mathbf{p}$ in its local frame. The robot aims to predict the Wi-Fi source position ${}_i\mathbf{p}^{*AP}$ and transform other robots' positions ${}_i\mathbf{p}^j$ into its local frame using vector transformation for relative localization.

Assumptions:

- All robots share the predicted AP position and their positions in the local frame.
- The Wi-Fi source position is stationary during the experiment.

- All robots are observing the same Wi-Fi source at any instance.

Fig. 6.2 delineates the system architecture of the proposed GPR system concerning robot i . The system contains three major components; GPR to train and optimize the GP model, source prediction using hierarchical inferencing using the GP model, and vector transformation for relative localization.

6.4 Source Positioning using Hierarchical Inferencing over GPR

We first present our hierarchical inferencing over GPR for accurate source position prediction.

6.4.1 RSSI-based GP regression

In the proposed approach, we use Received Signal Strength Indicator (RSSI) readings along with GPR to forecast the location of the Wi-Fi source. Due to their accessibility and simplicity, RSSI-based localization techniques have been widely used in indoor and outdoor applications. However, environmental changes, multipath propagation, and signal fading can all impact how well these techniques perform. To address the difficulties posed by RSSI-based localization, GPR offers a versatile and reliable method for modeling intricate interactions between inputs and outputs (Yiu et al., 2017). Wi-Fi RSSI readings have been used to apply GPR to radio mapping and localization applications successfully (Elgui et al., 2020; Fink & Kumar, 2010). The technique has proven successful in figuring out the spatial distribution of RSSI values and estimating the user’s location based on measured data. Using a single Wi-Fi source, we extend the application of GPR to the challenge of relative localization among multiple robots in our system.

RSSI-based GPR

Let $\mathbf{X} = \mathbf{x}_1, \dots, \mathbf{x}_N$ represent the N random positions within the environment, and let $\mathbf{Y} = y_1, \dots, y_N$ correspond to the RSSI measurements acquired at these positions. Our goal is to train the GP model \mathcal{M} utilizing the dataset $\mathcal{D} = (\mathbf{x}_q, y_q), q = 1 \dots N$ and predict the Wi-Fi source position by leveraging the learned associations between the positions and their respective RSSI values.

The mean function $m(\mathbf{x})$ of a Gaussian Process captures the expected value of the function at a given input \mathbf{x} . In our case, we can choose a simple constant

mean function to represent the expected RSSI value at any position:

$$m(\mathbf{x}) = \mu, \quad (6.1)$$

where μ is a constant representing the average RSSI value in the environment.

The kernel functions $k(\mathbf{x}, \mathbf{x}')$ of a Gaussian Process defines the covariance between the function values at different input points \mathbf{x} and \mathbf{x}' . In our case, we can use the popular squared exponential (SE) kernel, also known as the Radial Basis Function (RBF) kernel, which measures the similarity between the positions based on their Euclidean distance:

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2l^2}\right), \quad (6.2)$$

where σ_f^2 is the signal variance, l is the length scale parameter, and $\|\mathbf{x} - \mathbf{x}'\|^2$ is the squared Euclidean distance between \mathbf{x} and \mathbf{x}' . This kernel function encodes the assumption that the RSSI values at nearby positions are more correlated than those at distant positions.

Source Position Prediction

Using the trained GP model \mathcal{M} with the mean function $m(\mathbf{x})$ and the kernel function $k(\mathbf{x}, \mathbf{x}')$, we can predict the Wi-Fi source position $i\mathbf{p}^{*AP}$ in the frame of robot i by finding the position that maximizes the posterior distribution of the source position given the data \mathcal{D} :

$$i\mathbf{p}^{*AP} = \arg \max_{\mathbf{p}} p(\mathbf{p} \mid \mathcal{D}, m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')), \quad (6.3)$$

Where $p(\mathbf{p} \mid \mathcal{D}, m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$ represents the posterior distribution of the source position conditioned on the data \mathcal{D} , the mean function $m(\mathbf{x})$, and the kernel function $k(\mathbf{x}, \mathbf{x}')$. $i\mathbf{p}^{*AP}$ considered as initial source position estimation, which can be improved will be improved hierarchical inferencing.

In our proposed solution, the GP model is trained by taking RSSI samples at random points and then optimizing the model for better inferencing. The GP model learns the link by incorporating the mean function, kernel function, and the collected data.

6.4.2 Hierarchical inferencing for source position prediction

Over the GP regression model, we use a hierarchical inferencing strategy to predict the Wi-Fi Access Point (AP) position precisely. A multi-resolution search technique, hierarchical inferencing, refines the search for the ideal AP position from coarse to fine (Yin et al., 2019). We can quickly determine the position that maximizes the posterior distribution of the AP position while taking advantage of the structure and smoothness of the GP model by combining hierarchical inferencing with GP regression.

In the GP regression model, we aim to determine the position $i\mathbf{p}^{*AP}$ that maximizes the posterior distribution of the AP position. The process begins with the coarsest level of inference, which we'll denote as \mathbf{x}_1 .

At level 1, we have $\mathbf{x}_1 = (ix_1, iy_1)$. This pair of coordinates gives us the coarsest estimate of the AP position. We insert these values into our Gaussian Process (GP) regression model, which in mathematical terms, gives us $p(\mathbf{p} | \mathcal{D}, m(\mathbf{x}_1), k(\mathbf{x}_1, \mathbf{x}_1'))$. This posterior distribution represents the likelihood of our AP position given our first-level data.

For level 2, we refine the resolution and have $\mathbf{x}_2 = (ix_2 + res, iy_2 + res)$. The term 'res' indicates our resolution refinement at this stage. This yields the following posterior distribution: $p(\mathbf{p} | \mathcal{D}, m(\mathbf{x}_2), k(\mathbf{x}_2, \mathbf{x}_2'))$. This is a more refined likelihood of our AP position based on the second-level data.

At level 3, we further refine the resolution and have $\mathbf{x}_3 = (ix_1 + ix_2 + res, iy_1 + iy_2 + res)$. This provides the posterior distribution as $p(\mathbf{p} | \mathcal{D}, m(\mathbf{x}_3), k(\mathbf{x}_3, \mathbf{x}_3'))$, which is an even finer likelihood of our AP position.

In the end, the final data is obtained by summing the data from all levels: $\mathbf{x} = (ix_1 + ix_2 + ix_3, iy_1 + iy_2 + iy_3)$. Plugging this data into the GP model gives us the final posterior distribution $p(\mathbf{p} | \mathbf{x}, m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$.

Finally, we can state that the AP position that maximizes this final posterior distribution is our desired $i\mathbf{p}^{*AP}$, which will be used in Eqn.6.3.

In essence, the hierarchical inferencing strategy refines the position estimate at each level. It ultimately yields a position that maximizes the posterior distribution, ensuring high precision in predicting the Wi-Fi AP position.

We may efficiently use the smoothness and structure of the GP model by using hierarchical inferencing over GP regression, which enables the precise prediction of the Wi-Fi source position. Due to the approach's excellent computing efficiency and resistance to local optima, the robots can localize themselves with great accuracy.

Algorithm 9: Distributed Implementation of GP-based Relative Localization (GPRL) on Every Robot i

```

1 Input:  $K$  initial training samples (e.g., from random walk) with a GP
   training dataset  $\mathcal{D} = (\mathbf{x}_k, rssi_k)$ , where  $x_k = 1 \dots K$  are robot
   positions in local frames, and  $rssi_k$  are respective the RSSI values.
2 Output:  ${}_i\mathbf{p}^{AP}$  and  ${}_i\mathbf{p}^j$ , the position of AP and all neighbor robots  $j$  in
   the local frame of robot  $i$ .
3 begin
4   Train/Re-train GP model  $\mathcal{M}$  using RSSI-based regression with
   training samples  $\mathcal{D}$ , mean function  $m(\cdot)$ , and kernel function
    $k(\cdot, \cdot)$ ;
5   Perform hierarchical inferencing over the GP model  $\mathcal{M}$  to predict
   the Wi-Fi source position  ${}_i\mathbf{p}^{AP}$  in the local frame of robot  $i$ ;
6   Communicate  ${}_i\mathbf{p}^{AP}$  and robot odometry positions  ${}_i\mathbf{p}^i$  in local
   frames with other robots;
7   foreach neighbor robot  $j$  do
8     Compute the position of robot  $j$  in the local frame of robot  $i$ 
     using AP-oriented vector transformation in Eq. (6.4);
9   end
10 end

```

6.4.3 Relative localization

In relative localization, we assume that every robot monitors the same Wi-Fi Access Point (AP) and can send robot i information about its local AP predictions and location. With this configuration, we can use vector transformation to determine where robot j is about robot i .

Let ${}_i\mathbf{p}^{AP}$ represent the position of the Wi-Fi AP as predicted by robot j in the frame of reference of robot i , and let ${}_j\mathbf{p}^{AP}$ denote the position of the Wi-Fi AP as predicted by robot j in its frame of reference. We assume that $({}_i\mathbf{p}^i - {}_i\mathbf{p}^{AP}) = {}_i\mathbf{R}^j({}_j\mathbf{p}^j - {}_j\mathbf{p}^{AP})$ (as they are both observing the same Wi-Fi source). In the given system, robot j shares ${}_j\mathbf{p}^{AP}$ to robot i along with its current position ${}_j\mathbf{p}$. Furthermore, let ${}_i\mathbf{p}^{AP}$ represent the position of the Wi-Fi AP as predicted by robot i in its frame of reference.

Let ${}_i\mathbf{R}^j$ denote the rotation matrix that transforms a vector from robot j 's frame of reference to robot i 's frame of reference, so we can then define the transformation matrix in 2-D space as ${}_i\mathbf{T}^j = {}_i\mathbf{R}^j$:

Using this transformation matrix, we can transform the position of robot j from its frame of reference into the frame of reference of robot i using the following equation:

$${}_i\mathbf{p}^j = {}_i\mathbf{p}^{AP} + \mathbf{T}^j({}_i\mathbf{p}^j - {}_j\mathbf{p}^{AP}), \quad (6.4)$$

Thanks to this transformation, robots i and j can execute relative localization about one another, which enables robot i to pinpoint robot j 's location in its local frame of reference. All robots in the system may effectively acquire relative localization by using this vector transformation method, which is necessary for cooperative activities and swarm behaviors.

6.4.4 Overview

The Alg. 9 summarizes the proposed GP-based relative localization learning method. It starts with training the GP model using RSSI-based regression, followed by hierarchical inferencing to predict the Wi-Fi source position. The robots then communicate their local AP predictions and positions in local frames. Using the transformation matrix computed from initial robot orientations and robot positions, the position of robot j in the local frame of robot i is computed using vector transformation.

6.4.5 Theoretical Analysis

We can show that the hierarchical inferencing approach converges to the same results as the GPR with full resolution while achieving high computational efficiency, ensuring that the source position prediction is accurate and efficient. We derive the following lemma.

Lemma 6. *Given a dense resolution GPR model as defined in Section 6.4.1, hierarchical inferencing across GPR yields source position predictions with an approximation within a prediction threshold ε for sufficiently large t . Hierarchical inferencing allows for significantly improved computational efficiency.*

Proof. We first denote \mathbf{p}_{full} as the true source position predicted using full resolution GPR and \mathbf{p}_{hier} as the source position predicted using hierarchical inferencing. Our goal is to show that for an infinitesimally small threshold $\varepsilon > 0$, there exists $\mathcal{T} > 0$ such that for all $t > \mathcal{T}$, we have $\|\mathbf{p}_{hier}(t) - \mathbf{p}_{full}(t)\| < \varepsilon$, where t denotes the number of iterations of hierarchical inferencing. This implies that as we perform more iterations, the predictions of the hierarchical inferencing approach get arbitrarily close to the full-resolution GPR model

predictions. We note that the precision of this approximation is a function of the number of levels in the hierarchy and the resolution at each level.

The full resolution GPR model is defined in Section 6.4.1, where we express $f(x) \sim GP(0, k(x, x'))$ and $y = f(x) + \varepsilon$, where $\varepsilon \sim N(0, \sigma^2)$.

Now, given n observations in a dataset $D = (x_q, y_q)_{q=1}^n$, the joint distribution of the observed output values $\mathbf{y} = (y_1, \dots, y_n)$ and the function values f at new points X under a Gaussian process prior is:

$$\begin{pmatrix} \mathbf{y} \\ \mathbf{f} \end{pmatrix} \sim N \left(\mathbf{0}, \begin{pmatrix} K(\mathbf{X}, \mathbf{X}) + \sigma^2 I & K(\mathbf{X}, X') \\ K(X, \mathbf{X}) & K(X, X') \end{pmatrix} \right) \quad (6.5)$$

Herein $K(\mathbf{X}, \mathbf{X})$ is the covariance matrix computed using the covariance function k on our observations, $K(\mathbf{X}, X')$ is the covariance between our observations and the new points, and $K(X, X')$ is the covariance of the new points. After observing \mathbf{y} , we calculate the posterior predictive distribution for \mathbf{f} at new points \mathbf{X} , which provides the expected function values and the uncertainty of those predictions.

The computational complexity of calculating the posterior predictive distribution for all grid points in the full-resolution GPR model is $\mathcal{O}(n^3)$, due to the need to invert an $n \times n$ matrix, $K + \sigma^2 I$.

The hierarchical inferencing method begins with a coarse grid and refines iteratively, focusing on areas with higher posterior probability. This process reduces the number of grid points required at each level. Therefore, the computational complexity of the hierarchical inferencing method is $\mathcal{O}(mk^3)$, where m is the number of grid points at each level of the hierarchy and k is the number of levels. Since $m \ll n$ and k are often much smaller than n , the hierarchical inferencing method offers higher computational efficiency than the full-resolution GPR model. \square

Now we derive the main theoretical results of this paper.

Theorem 3. *Given two robots i and j with known rotation matrix, and their respective local Wi-Fi AP predictions ${}_i\mathbf{p}^{AP}$ and ${}_j\mathbf{p}^{AP}$, as well as the shared Wi-Fi AP prediction, the vector transformation using the transformation matrix ${}_i\mathbf{T}^j$ correctly computes the position of robot j in the frame of reference of robot i , denoted by ${}_i\mathbf{p}^j$.*

Proof. Trivial, followed by Lemma 6 and relative position transformations in Eq. (6.4). \square

Table 6.1: Resolution (m) of the GPR prediction map in Robotarium simulations. Note the environment size is 3m x 2m.

MEGPR/GPR		GPRL
Sparse	Dense	Hierarchical (up to 4 Levels)
0.1	0.0125	[0.1, 0.05, 0.025, 0.0125]

6.4.6 Time Complexity

GPR, hierarchical inferencing, and vector transformation for relative localization are the three main computational elements that can be used to examine the temporal complexity of the proposed relative localization system. Assuming that the GP model is pre-trained, the complexity of GP regression, given N training points of the RSSI map, is $\mathcal{O}(\alpha N^2)$ and $\alpha \in (0, 1]$ is reduced resolution factor. For inferencing, We reduced the N to λN , where $\lambda \in (0, 1]$ is the resolution reduction factor for each hierarchy, hence overall inferencing time for l -levels of hierarchy is $\sum_{i=1}^l \mathcal{O}(\lambda N^2)$. Given n robots, vector transformation has a $\mathcal{O}(n)$ time complexity. As a result, the suggested system's overall time complexity is $\mathcal{O}(\sum_{i=1}^l (\lambda N^2) + n)$. Even though the GP regression and hierarchical inferencing components are the most difficult, the system can be efficiently run by limiting the number of training points and grid points and evenly dividing the computational burden among the robots.

6.5 Experimental Setup

This paper presents an experimental setup for testing a proposed technique using a single integrator controller in the Robotarium. The experiments are conducted for 300 iterations and ten trials. In the experiments, the robots move randomly around the simulated area while forecasting the source location from the GPR model.

6.5.1 Robotarium Setup

In our experiments, a 3.2×2 meter rectangle is simulated using the Robotarium platform to test the suggested GPR model-based localization technique. A predetermined number of robots populate the simulation environment, and

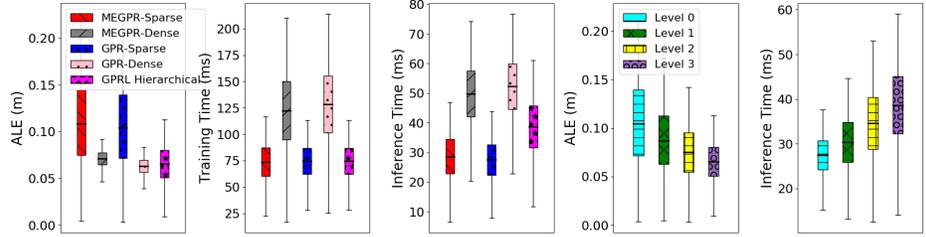


Figure 6.3: Source Prediction performance evaluation comparison plots of SOTA approaches per iteration; MEGPR (M. Xue et al., 2019) (sparse and dense AP subset) and the proposed approach (GP-based source inferencing with hierarchical inferencing and without hierarchical inferencing) along with the effect of hierarchy.

each robot’s initial position is randomly chosen. The robots may randomly stroll in the predetermined area because they are set up to use a single integrator controller. To maintain a controlled experimental setup, virtual limits are added to the simulation environment to prevent robots from straying outside the designated area.

The Robotarium’s server communicates with the robots during the tests, enabling the exchange of data necessary for localization and the suggested learning process. The platform gathers and records information about the positions of the robots as well as source location forecasts derived from the GPR model. The performance of the suggested localization approach is assessed through the processing and analysis of this data. The Robotarium’s remote accessibility enables a reliable and regulated setting for carrying out our tests, fostering the repeatability and confirmation of the findings made in this study.

6.5.2 GPR Model

Based on the available information from the robots’ positions and their related source location measurements, the GPR model is used to forecast the source location. A non-parametric probabilistic generalized projection rule (GPR) model can capture complicated relationships between input and output data, offering predictions and estimating the uncertainty involved. GPR models are ideally suited for jobs involving erratic data or noisy observations because of this property, which is why they successfully predicted the source position in our trials. The GPR model is then updated using this information by adjusting

its hyperparameters and adding new data. The experimental and hierarchical configurations can be seen in Table. 6.1.

In our trials, the GPR model is initially trained on ten random samples, each measuring the source location and robot position. After each iteration, the model is updated, improving its predictive power by incorporating the most recent data collected from the robots.

6.6 Results and Discussion

Based on the experimental setup discussed in Sec. 6.5, we have performed experiments and analyzed the performance in this section.

6.6.1 AP Source Prediction Results

Compared to the SOTA approaches, the suggested GPRL strategy with hierarchy performs admirably, outperforming them in terms of accuracy and efficiency, as shown in Fig. 6.3. The findings show that in terms of source ALE (Average Localization Error), training time, and inference time, GPRL with hierarchy outperforms its competitors.

The ALE of source (AP) localization is improved by GPRL with hierarchy, with an error roughly 36% lower than MEGPR-Sparse and roughly 8% lower than MEGPR-Dense. This indicates our method’s increased accuracy, which is necessary for relative localization jobs. The training time for our suggested strategy is reasonable. It takes about 40% less time to train than MEGPR-Dense while matching the GPRL training time without hierarchy. This demonstrates the effectiveness of our strategy because it can deliver quick and precise instruction without sacrificing accuracy.

Additionally, GPRL with hierarchy performs better than other approaches in inference time. Our method is around 22% quicker than the MEGPR-Dense approach. The higher accuracy in source ALE justifies this minor increase in inference time, even if GPRL with hierarchy takes a little longer than GPRL without hierarchy. Furthermore, the effect of hierarchy is linear with the number of levels and provides evidence for its effectiveness with improved source prediction accuracy. These findings demonstrate the dominance of the GPRL with hierarchy and the possibility of our technique in real-world settings where accuracy and efficiency are key considerations.

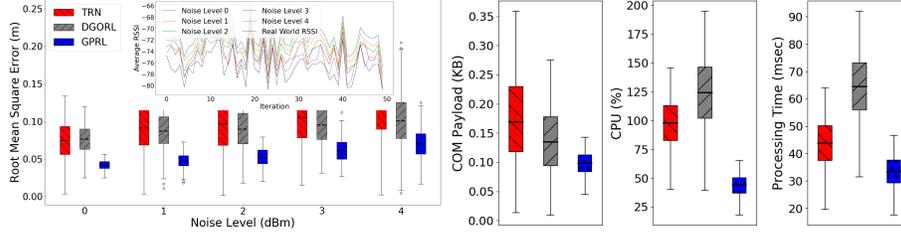


Figure 6.4: Results of various performance metrics. From left to right, we show the localization accuracy (RMSE) under different simulated RSSI noise levels (the embedded plot represents the respective RSSI variations), the communication payload, the computation overhead, and the run time performance of various relative localization approaches. It can be seen that the GPRL consistently outperforms other approaches in all the metrics.

6.6.2 Relative Localization Experiments

For relative localization experimentation, we compare the performance of our proposed approach with state-of-the-art Terrain Relative Navigation (TRN) (Wiktor & Rock, 2020) and Distributed Graph Optimization for Relative Localization (DGORL) (Latif & Parasuraman, 2022a) to validate the accuracy and robustness. We employ the Root Mean Square Error (RMSE) assessment metric to assess how well the proposed approach performs. The average localization error is quantified by RMSE, which also captures differences between the robots’ anticipated relative positions and their actual positions in the simulated environment. Insights into the convergence rate and robustness of the learning approach are shown by analyzing RMSE values throughout the experiments. This reveals how well the learning method works to improve the GPR model and adapt to new data under various initial conditions and random walks. We can confirm the efficacy and dependability of the suggested localization strategy in the Robotarium trials thanks to this thorough study using RMSE. We also have computed computation complexity in terms of CPU percentage utilization and processing time (ms) per iteration for each approach to validate the efficiency of the proposed approach.

Accuracy

Our analysis of the RMSE values for various approaches highlights the greater accuracy of our GPRL strategy (with hierarchy) in predicting robots’ relative placements. Compared to alternative techniques, GPRL with Hierarchy ex-

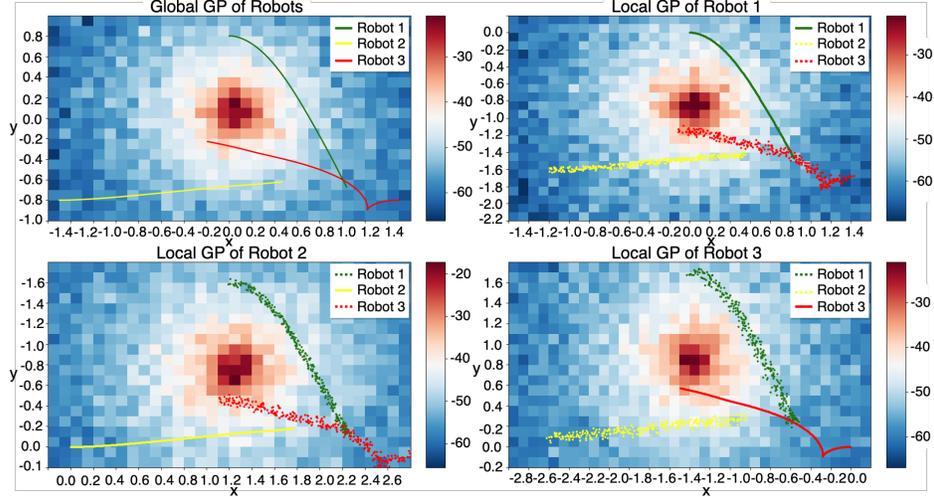


Figure 6.5: Gaussian process inferencing for global and local frames along with ground truth and predicted robot trajectories.

hibits a considerable improvement in the average RMSE. It is more accurate than TRN by roughly 41.73 percent, or a ratio of about 1.72. GPRL with Hierarchy is roughly 30.84% more accurate than DGORL, which translates to a ratio of about 1.45. Its accuracy is roughly 34.01% more than the Sparse version of GPRL without Hierarchy or a ratio of about 1.52. The Hierarchy version of GPRL has a little higher average RMSE than the Dense version of the algorithm without Hierarchy, though. As shown in Fig. 6.4, the reduced RMSE values obtained by GPRL with hierarchy prove its advantage in various settings and starting conditions. Fig. 6.5 shows the Gaussian prediction for global and local frames along with the ground truth and predicted trajectories.

In addition to having better accuracy than previous approaches, the GPRL with hierarchy technique also shows less fluctuation in RMSE values among the three robots. The reliability of this consistency for localization tasks is highlighted. The GPRL, with hierarchy, establishes itself as a good contender for practical robot localization tasks by offering higher accuracy and consistency.

Efficiency

Our proposed GPRL with the hierarchy method greatly outperforms others when comparing the processing times and CPU use in various ways. The GPRL with Hierarchy uses 43.62% of the CPU, which is significantly less than other methods. It uses 45.05 % less CPU than TRN (97.36%), making it around 2.23 times more efficient. The GPRL with Hierarchy shows a reduction of 64.67%

compared to DGORL’s CPU consumption of 123.56%, making it nearly three times (2.83 times) as effective. The Hierarchy version of GPRL, however, uses 23.18% more CPU power than the Sparse version of the same algorithm without Hierarchy. However, it outperforms the Dense version of GPRL without Hierarchy in terms of efficiency, requiring 51.13% less CPU and being nearly 2.05 times as efficient. As shown in Fig. 6.4, the reduced CPU utilization and processing time values obtained by GPRL with hierarchy prove its efficiency in a practical environment.

GPRL with Hierarchy completes the task in 33.35 milliseconds of processing time. GPRL with Hierarchy is more time-effective due to this processing time being 23.59% faster than TRN’s processing time of 43.65 msec. The hierarchy model also surpasses DGORL, which processes in 64.34 msec and is nearly twice as efficient (≈ 1.93 times), outperforming it by $\approx 48.20\%$. In contrast, the Sparse version of GPRL without Hierarchy exhibits a 55.69% increase in processing time when compared to the GPRL with Hierarchy, showing that the Sparse version is more effective in this regard. However, the processing time efficiency of the GPRL with Hierarchy is 48.91% greater than that of the Dense version of the GPRL without Hierarchy, or almost 1.96 times.

Impact of the RSSI noise levels

As we rely on the RSSI measurements to build the GPR map, the noise level in the RSSI measurements can affect the accuracy of the localization. To analyze the localization accuracy (RMSE) for different simulated noise levels in the measured RSSI values. The overlay plot in Fig. 6.4 shows that the simulated RSSI at noise level 4dBm represents real-world RSSI observations. Furthermore, the results have shown that the proposed approaches have lower RMSE (high accuracy) among all techniques, even under high noise levels. The improvement in accuracy is more pronounced when the noise level is increased. DGORL performed better than the TRN; however, both have 2x lower accuracy than the proposed GPRL for all trails.

6.6.3 Real-world Demonstration

To validate the practicality and generalizability, we have applied the GPRL approach to perform a multi-robot rendezvous task using three Tuttlebot2e robots in a large 10m x 13m multi-room lab environment. All robots are configured as fully connected to the same WiFi AP. The robots are initially located in different rooms (without visibility to each other and the AP). Robots operate in a distributed way to find the direction to pursue using perceived relative locations

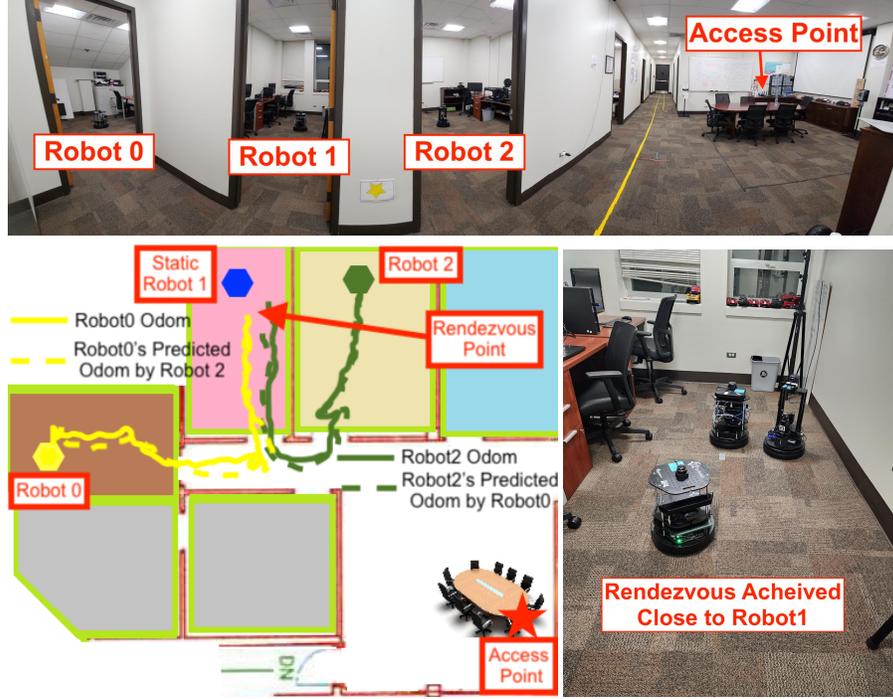


Figure 6.6: Example trial from the multi-robot rendezvous experiment: Initial (top) and final (bottom) state of the robots and their trajectories using GPRL.

of other robots in their local frame of reference using ${}_i\hat{\mathbf{p}} = \frac{1}{N} \sum_{j \in \mathcal{N}_i} ({}_i\mathbf{p}^j - {}_i\mathbf{p}^i)$, where \mathcal{N}_i is the set of neighbor robots of the robot i . This direction is then given to the robot's mapping and autonomous motion planner for low-level path planning and obstacle avoidance (we used ROS gmapping + move_base packages for this implementation).

We performed five trials and successfully achieved rendezvous of all robots within a small threshold. A sample of the initial and rendezvous positions and their trajectories can be seen in Fig. 6.6. The robots were able to locate the AP within 0.573 ± 0.12 m accuracy, and the relative trajectory error (between the odometry and predicted trajectory of other robots) was within 0.426 ± 0.098 m on average. The attached video shows the performance of GRPL in real time. The experiment validated the practicality of the approach in handling real-world scenarios with noisy RSSI values with occluded non-line-of-sight conditions. It also demonstrates the applicability of GPRL to most multi-robot operations, such as exploration and formation control.

6.6.4 Discussion

Our experiments' outcomes show that the proposed GPRL approach performs better than the TRN and DGORL methods regarding accuracy and effectiveness. The GPRL method's increased accuracy can be attributed to its capacity to understand the intricate connections between the positions of the robots and the sources utilizing the GPR model. The proposed method's improved efficiency results from its effective model updating and consensus-based localization strategy. For practical application in robot localization tasks, the GPRL approach is a strong contender thanks to these benefits, especially in situations where real-time performance and computing effectiveness are essential. Moreover, the proposed GPRL is a scalable approach regarding the number of robots as there is no global GPR search involved, and space dimensions are constrained to the range of wireless signal transmitters, as the hierarchical approach is adaptive to the dimensions and will have no significant impact on the overall performance of the system.

Although the proposed GPRL approach produced some encouraging results, some issues still need to be resolved. The approach depends on Wi-Fi source measurements, which are susceptible to interference, multipath propagation, and signal deterioration, potentially reducing the precision of the localization predictions. The method also assumes that the robot orientations are known, which may not be true in some real-world situations where the robot's orientation data is incorrect or missing. Furthermore, the suggested method does not explicitly consider non-line-of-sight circumstances, which can affect the predicted source locations and, as a result, the estimated relative positions. The accuracy of the Wi-Fi source measurements, for instance, can be increased by utilizing reliable signal processing techniques, such as synergistic information fusion (Latif & Parasuraman, 2022b). Methods like Angle of Arrival (AoA) (Jadhav et al., 2022) can be coupled with the GPR model to produce more reliable localization predictions in environments with barriers or signal blockages, addressing the issue of known robot orientation and non-line-of-sight and improving estimates of the robots' poses. The proposed GPRL strategy can be made even better by combining these methods, increasing its practical usability and performance in real-world situations.

6.7 Summary

This chapter delves into the development and testing of a novel learning-based method for relative localization in multi-robot systems using WiFi signals' Re-

ceived Signal Strength Indicator (RSSI) data, known as GP-Loc. The necessity for efficient and accurate relative localization methods stems from the rising utilization of autonomous mobile robots in various applications such as logistics, surveillance, and search and rescue. Despite the existence of several methods leveraging reinforcement learning, Bayesian optimization, and deep learning, these often require large quantities of data and may experience overfitting or generalization problems.

GP-Loc circumvents these limitations through a unique combination of hierarchical inferencing and Gaussian Process Regression (GPR), alongside a vector transformation method, enabling accurate relative localization even when robots aren't in direct line of sight or have different initial orientations. The efficiency and accuracy of GP-Loc are validated through extensive experiments in a simulated Robotarium world and with real robots, showing superior performance to state-of-the-art methods such as Terrain Relative Navigation (TRN) and Distributed Graph Optimization for Relative Localization (DGORL).

Despite the promising results, the GP-Loc approach does face challenges, including its reliance on WiFi source measurements that are susceptible to interference and the assumption of known robot orientations. Future improvements will address these constraints by incorporating signal processing techniques, Angle of Arrival (AoA) methods, and synergistic information fusion, enhancing the overall practicality and performance of GP-Loc in real-world scenarios. This chapter discussed the final version of relative localization using the Gaussian process and proposed an efficient solution that will further be integrated with exploration strategies discussed in the coming chapters.

CHAPTER 7

COMMUNICATION-EFFICIENT REINFORCEMENT LEARNING IN SWARM ROBOTIC NETWORKS FOR MAZE EXPLORATION

7.1 Introduction

From this chapter, we redirect the dissertation to the exploration strategies, this chapter in particular explores the use of reinforcement learning for maze exploration in an efficient way. Maze coverage and exploration-related challenges have always piqued the interest of humans. Path planning algorithms are typically used to regulate how robots move through mazes if the structure of the labyrinth (including its walls and paths) is already known. If the maze structure is unknown, the robots must first use sensors to find a portion of the maze nearby before planning their next move based on the knowledge they have gained. Exploring an unknown maze in pursuit of stationary targets is thus a challenge associated with coverage, search, and path-finding issues (Alamri et al., 2021). Typical maze coverage solutions such as A* (Tjiharjadi et al., 2017), and Depth First Search (Y.-H. Chen & Wu, 2020) apply to a single robot system and require constant updates and optimization. However, most current research focuses on coordinated multi-robot and swarm robotic networks (Youssefi & Rouhani, 2021).

Numerous research has solved the issue of coordinating and controlling several robots for mapping and exploration. However, most strategies rely on

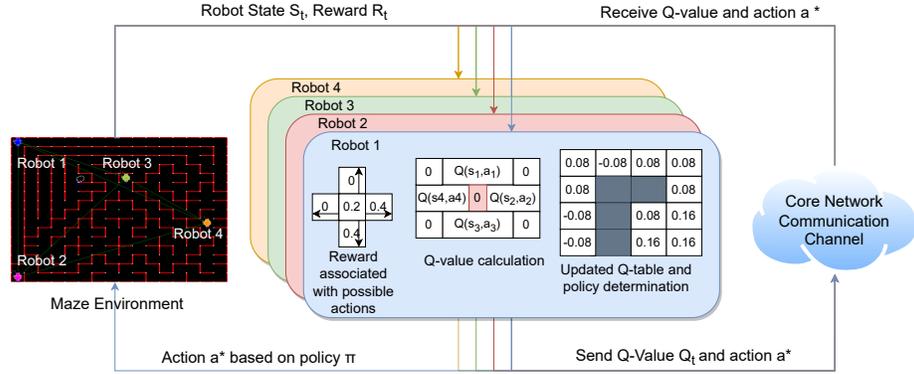


Figure 7.1: Overview of the Communication-Efficient Reinforcement Learning.

centralized control to guide each robot in a swarm because centralized coordination enables almost optimal behaviors in surroundings that are well understood. However, in the novel, unknown contexts, a distributed approach for swarm coordination is necessary, as was put forth by Bono et al. (Bono et al., 2021), where it produced a reliable result even if one or more of the robots were lost. Specifically, sharing locally observed maze information among the connected robots is a crucial component of distributed approaches so that each agent can carry out its plan without interference from the others. However, a reliable information fusion technique needs to be integrated with such approaches for communication efficiency and a reliable system for information fusion (Kalinowska et al., 2022).

Reinforcement learning (RL) is a machine learning technique that trains agents to take action in an environment to maximize a reward signal. In swarm robotic networks, the robots can be trained to explore the maze effectively by learning from their interactions with the environment. This problem is challenging because the robots may have different capabilities, and the maze is unknown.

Multi-robot maze coverage using RL has been recently explored for solving the problem of coordinating a team of robots to explore and map a maze (Gu & Mao, 2021). However, recent RL algorithms fail to scale up quickly for a swarm system and do not work efficiently in communication-constrained environments where the packet loss rates are high, and the communication range may be limited. Therefore, to remedy this issue, we contribute a new efficient novelty search through deep reinforcement learning (DRL) (L. Shi et al., 2020)

to coordinate a group of robots and solve the maze exploration problem distributedly; an overview of the proposed approach can be seen in Fig. 7.1.

One approach to solving the swarm robotic maze exploration using RL is to define the problem as a Markov Decision Process (MDP), where the states represent the positions of the robots and the walls of the maze, the actions represent the movements of the robots, and the rewards represent the progress made towards exploring the maze. The robots can then use an RL algorithm, such as Q-learning, to learn a policy that maps states to actions and maximizes the rewards in a distributed manner.

We theoretically analyze our approach with other methods from the literature. We extensively evaluated the proposed algorithm using CORE network simulations in different maze worlds. This paper focuses on the efficiency of the proposed approach for micromouse maze coverage under standard networking conditions; hence we are not discussing networking aspects of the work. Comparing the results from a graph-based Depth First Search (DFS) (Y.-H. Chen & Wu, 2020) and the recent memory-greedy reinforcement learning (Yu et al., 2021) approaches, the proposed RL algorithm outperforms in terms of coverage efficiency and minimizing overlapping regions, especially in communication-degraded environments. We open-source the codes in GitHub at <https://github.com/herolab-uga/MazeCommRL> for use and further development by the robotics community. The algorithm can be generalized to other application domains where communication efficiency is crucial for better swarm coordination.

7.2 Related Work

In scenarios where both the world (map of the maze) is unknown and multiple robots need to coordinate effectively to explore the full map, the robots must search the uncharted maze to locate a target. Many algorithms have been proposed in the literature. For instance, a search-based method to locate a path out of the maze may not always provide the shortest path (Tjiharjadi et al., 2022). It is also becoming increasingly important to plan how to address the multi-robot pathfinding that more effectively bypasses cost estimations during preparation (Luis et al., 2020). In (W. Shi et al., 2019), an optimization-based UAV trajectory planning method is proposed to enable a drone-assisted open radio access network. In (L. Wang & Guo, 2019), a language-based explicit communication is introduced for swarm robots to coordinate and plan their tasks with low communication costs.

However, most of them assume a centralized control or coordination system. The robots do not work together effectively, making it impossible for a swarm robotic network to perform well without significant memory and communication overload.

Reinforcement learning provides a superior way to tackle communication-efficient swarm coordination challenges. This is because robots' actions impact other robots' actions and vice versa, which encourages robots to collaborate by utilizing other robots' information through communication (Pozza et al., 2021; Q. Yang & Parasuraman, 2021). Such information benefits the robots' cooperation, but they cannot ensure they will communicate all the information they need to exchange to collaborate.

In the maze problem, the robot will receive a positive reward when it arrives at its destination, meaning that the reward it just received will allow it to return to the original region numerous times. These prizes, however, only show up one time in the maze. This circumstance, known as a "reward loop," significantly extends the time needed to navigate the maze. A Q-learning method with multiple Q tables is suggested in a study (Kantasewi et al., 2019). This approach reduces map area overlap between robots by creating a Q table each time the robot explores a new region, allowing the robot to navigate to the target optimally. The Q-learning is expanded in (Uwano & Takadama, 2017), and an RL for multi-robot cooperative tasks is proposed. The standard reward is transformed into an "internal reward" when robots in the maze cannot communicate with one another. This allows the robots to learn following this "internal reward" type and achieve task collaboration under such challenging circumstances.

A unique RL maze navigation technique is proposed in (Yu et al., 2019) to address the labyrinth navigation problem for robotic vehicles. First, the picture data of the random maze is collected using a drone's bottom-mounted camera. Then the virtual maze is created in the simulation environment using an image processing technique. An enhanced Q-learning algorithm is suggested to address the issue that the original greedy strategy repeatedly forces the robot to linger in the past state. However, it cannot tolerate significant communication degradation. Alternatively, our proposed approach considerably differs from the literature by considering an efficient information transfer mechanism combined with Q-learning for multi-robot cooperation to solve maze exploration tasks.

7.3 Proposed Approach

Let us assume multiple robots of different types are tasked with exploring and mapping a maze. Each robot has unique capabilities and strengths, such as different speeds and sensor ranges. Using reinforcement learning, the robots learn to cooperatively navigate the maze and cover as much as possible. The robots communicate and coordinate with each other to efficiently cover the maze and avoid collisions. We assume the robots can localize in a global frame of reference (e.g., GPS); however, recent relative localization techniques like (Latif & Parasuraman, 2022a) can be employed if global positioning is unavailable.

As the robots explore the maze, they receive rewards for discovering new areas and penalties for collisions. Over time, the robots learn the optimal strategies for exploring the maze and maximize their coverage. Each robot decides its action plan based on the below approach.

7.3.1 Reinforcement Learning

Popular machine learning algorithms like reinforcement learning use interactions with the environment to teach robots new skills. Markov decision processes are frequently used to model it. The robot's state is one of them and is denoted by the letter s . A state transition from state s_t to state s_{t+1} occurs due to the reinforcement learning robot selecting and carrying out an action based on its state at time t . The robot will receive a reward for each action. The robot will have learned the course of action to take in each condition and will be able to maximize the reward of the entire process through repetitive trial and error. Fig. 7.1 depicts the fundamental idea of RL, and the symbols and concepts used in RL are as follows:

- $S = \{s_1, s_2, s_3, \dots, s_n\}$ is a discrete set of n states, where $s_t \in S$ describes the state of the robot in the environment at a time t .
- $A = \{a_1, a_2, a_3, \dots, a_n\}$ is a discrete set of n actions, where $a_t \in A$ describes the action which the robot chooses at a time t .
- $T : S \times A \times S \rightarrow [0, 1]$ is a stochastic state transition function, where the state of the robot is transitioned to state s^* with a probability $p \in [0, 1]$ when choosing action a in state s . We use $s^* \leftarrow T(s, a)$ to represent the above process.
- $R : S \times A \times S \rightarrow R$ is the reward function. It represents the robot's reward in its state transition to s^* after executing action a in state s .

- $\gamma \in [0, 1)$ is the discount factor is the relative importance of future and present rewards.

Kantasewi et al., (Kantasewi et al., 2019) suggest a Q-learning with a multi-table approach. It is a technique that can be continuously updated in light of prior knowledge to ultimately arrive at the most likely accurate choice. The Q table is the Q learning’s secret. All possible states and actions are created using the Q table, which then updates each value through iterative learning. The robot then chooses the best course of action for each state based on the values in the table. This approach is frequently utilized in path planning, chess, card games, and other activities. The objective of the proposed approach is to perform maximum coverage in less time and avoid overlapping exploration, which can be numerically defined as:

$$\max_{\pi} \{P_a^{\pi}(t) - \lambda E_t(a|\pi)\}, \quad (7.1)$$

where $P_a^{\pi}(t)$ is the probability of covering the region for action a using policy π at time t , $E_t(a|\pi)$ is an average number of steps to cover associated with action a at time t in policy π and λ is the cost associated with each step.

The algorithm updates the Q value by the following formula:

$$Q_{t+1}(s_t, a_t) = (1 - \alpha)Q_t(s_t, a_t) + \alpha [r_t + \gamma \max Q_t(s_{t+1}, a_t)], \quad (7)$$

where s_t and s_{t+1} are the current states and the next state, respectively, it is the action executed, $\alpha \in (0, 1]$ controls the balance between the coverage and delay, and γ is the discount factor. At each discrete time step t , the agent acquires an observation s_t from the environment, selects a corresponding action a_t , then receives feedback from the environment in the form of a reward $r_t = R(s_t, a_t)$ as:

$$R_t = \sum_{i=1}^N \left[\alpha_i \cdot \left(\frac{1}{d_{i,t}} - \frac{1}{d_{i,t-1}} \right) + \beta_i \cdot \left(1 - \frac{1}{d_{i,t}} \right) \right], \quad (7.2)$$

where N is the number of robots, $d_{i,t}$ is the distance of a robot i from the goal at time t , α_i and β_i are constants that represent the reward for each robot.

The updated state information s_{t+1} . The goal of the RL agent is to select policy π to maximize the discounted sum of future rewards, i.e., $Q_{\pi}(s_1) = \sum_{\tau=1}^{\infty} \gamma^{\tau} R(s_{\tau}, a_{\tau})$, which according to the Bellman optimality principle satisfies.

This reward function produces a negative reward whenever the agent has looped back and no reward otherwise. Alg. 10 presents the pseudocode description of the approach using reinforcement learning.

Algorithm 10: Communication-Efficient RL for Exploration

```
1 Initialize the maze environment and the agent's state  $s_0$ ;  
2 Initialize the reward functions  $R_1, R_2, \dots, R_n$ ;  
3 foreach episode do  
4   Initialize the episode's total reward  $R = 0$ ;  
5   for each step in the episode do  
6     Take action  $a$  according to the agent's policy  $\pi$ ;  
7     Observe the reward  $r$  from the environment;  
8     Update the total reward  $R = R + r$ ;  
9     Update the agent's state  $s_{t+1} = f(s_t, a)$ ;  
10  end  
11  foreach reward function  $R_i$  do  
12    Calculate the reward  $r_i = R_i(s_0, s_T)$ ;  
13    Update the total reward  $R = R + r_i$ ;  
14  end  
15  Update agent's policy  $\pi$  using gradient ascent on  $R$ ;  
16 end
```

7.3.2 Swarm robot cooperation

The next issue we have addressed is the communication amongst individual exploration-capable robots in our multi-robot system. Our proposed RL-based exploration system is designed for a single robot, allowing each robot to make independent decisions based on local information and with little interaction from other robots. We introduce efficient communication amongst nearby robots to encourage cooperation. We develop a discovery approach based on the distance between simulated robots to replicate the network range in which we only share the current position of a *robot_i*, its Q-Value for each direction, and mark the current situation as explored to avoid repetitive exploration. When another robot receives the information, it will update the received Q value in its Q table and update the local map.

Table 7.1: Comparative complexity analysis of maze exploration.

Algorithm	Category	Complexity	Coverage	Update Cost	Abbreviations
Ant colony optimization (Viseras et al., 2016)	Centralized	$O(2n \times m)$	partial	low	b-obstacles, m-robots, n-cells
Cooperative Random Walk (Ozdemir et al., 2019)	Decentralized	$O(n^3)$	partial	low	n - number of cells
Improved SAT (Surynek et al., 2016)	Centralized	$O(\mu \times n \times m)$	full	high	μ - makespan factor
Efficient DRL (L. Shi et al., 2020)	Decentralized	$O(\epsilon \times n)$	full	high	ϵ - depth factor
Memory-Greedy RL (Yu et al., 2021)	Decentralized	$O(\epsilon_2 \times n)$	full	high	ϵ_2 - greedy degree
Proposed RL Approach	Decentralized	$O(k)$	full	moderate	k - number of sub-mazes

7.3.3 Procedure

Based on the RL and cooperation strategy discussed above, our approach incorporates cooperation into Q learning. A multi-robot cooperation strategy learning method based on RL has been suggested to solve the issue that the state action space is too large due to the complex maze environment when using traditional Q-learning to solve maze problems, resulting in too many iterations and too long in the entire learning process. These are the precise stages:

- Given the target maze M , it is divided into multiple sub-mazes. We regard these sub mazes as a set m , where $m = m_1, m_2, m_3, \dots, m_n$.
- Multiple robots use Q-learning to explore these sub-mazes at the same time. Because the sub-maze is much simpler than the target maze, it only needs a few iterations to complete the learning.
- Each robot updates its Q table at every iteration based on Q-value update Eq. 1. After learning, we can get a set of Q tables, where $Q = \{Q_1, Q_2, Q_3, \dots, Q_n\}$.
- Combine these Q tables as the initialized Q table of the target maze, called Q_{cop} .
- Finally, based on Q_{cop} , the robot completes the exploration of the target maze M .

7.3.4 Theoretical Analysis

We have compared our proposed approach with state-of-the-art algorithms from four major categories for maze exploration: Ant colony optimization (Viseras et al., 2016) (search-based); cooperative random walk (Ozdemir et al., 2019) (randomized); Efficient SAT (Surynek et al., 2016) (reduction-based); and Efficient DRL (L. Shi et al., 2020) (machine learning). Ant colony optimization is a centralized algorithm that provides partial coverage with a complexity of $O(2n \times m)$ for n robots in a maze of m cells. Cooperative random walk operates purely randomly and cannot guarantee full maze coverage with a high time complexity of (n^3) for worst-case scenarios. Improved SAT using a lower bound on the sum of costs and an upper bound on the makespan provides complete coverage of the maze with high update cost. Efficient DRL is the most comparative approach and has ϵ depth factor for exploration and shares the whole Q table in every iteration but can operate efficiently to cover the entire map in less time with full cooperation. Our proposed approach works in a decentralized fashion; each robot calculates the Q value for its current cell, shares only the

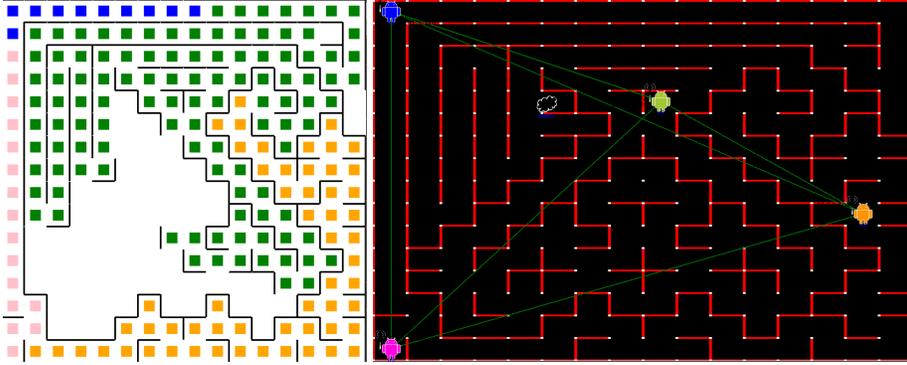


Figure 7.2: Core Simulator (**Right**) with exploration map (**Left**).

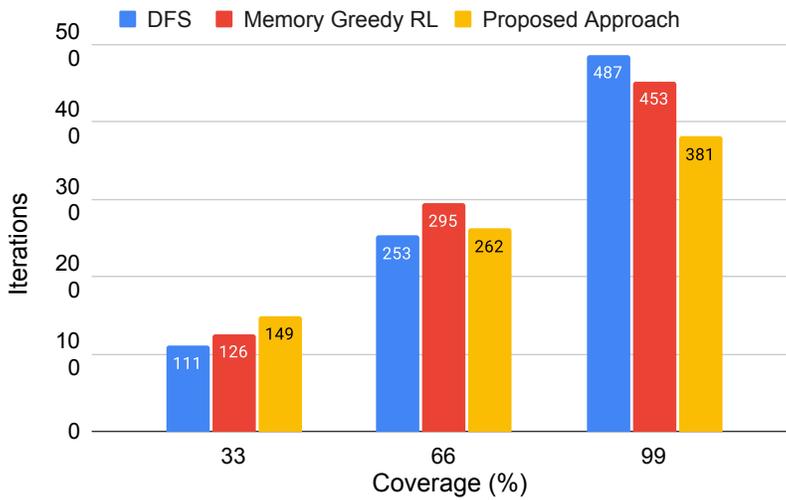


Figure 7.3: Time taken by maze coverage strategies for three levels of coverage (one-third, two-third, and full).

current value, and updates the Q table accordingly; hence, our approach has low communication/update cost and only traverses a single time through each cell to provide full maze coverage. Table 7.1 depicts that the proposed approach provides full maze coverage with reasonable computational efficiency and low update cost.

7.4 Experimental Evaluation

We have performed simulation experiments using CORE⁸ Networking emulator. All experiments were performed under standard CORE wireless network configurations other than the parameters mentioned below. In our experiments,

⁸ <http://coreemu.github.io/core/>

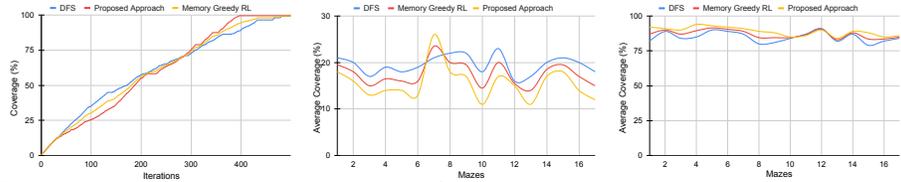


Figure 7.4: Coverage percentage performance metric: Average results over all mazes (**Left**), after 100 iterations (**Center**); after 400 iterations (**Right**).

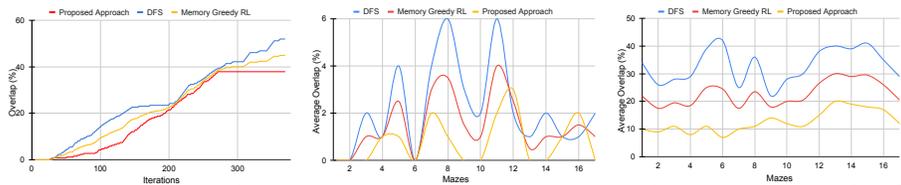


Figure 7.5: Map overlap percentage metric: Average results over all mazes (**Left**), after 100 iterations (**Center**); after 400 iterations (**Right**).

we set up four robots on four corners of the maze (see Fig. 7.2). Initial positions, motion constraints, communication channels, and maze dimensions are known for experimentation.

- Range: 1500 pixel distance between nodes
- Delay: $0.2 \mu\text{sec}$
- Bandwidth: 54 Mbps

We validated the performance of the proposed approach in comparison with DFS (Y.-H. Chen & Wu, 2020), and the memory-greedy RL (Yu et al., 2021) approaches. In the implemented approaches, the robots maintain synchronous communication and share only the current state and observed region with other robots in the maze. DFS does not incorporate received information other than updating the current exploration map, which increases the overlapping of exploration over iterations. We have performed experiments for 17 mazes available online⁹, each with a 16×16 grid size.

⁹ <http://www.tcp4me.com/mmr/mazes/>

7.4.1 Results

Coverage: We look at the time efficiency to complete the maze with one-third (33%), two-thirds (66%), and full coverage (99%); results can be seen in Fig. 7.3. DFS outperformed other approaches for one-third coverage as its processing and propagation does not involve robot cooperation and explore the maze

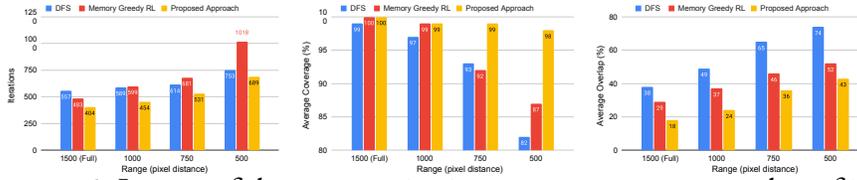


Figure 7.6: Impact of the communication range parameter; number of iterations (**Left**), average coverage percentage (**Center**), average overlap percentage (**Right**).

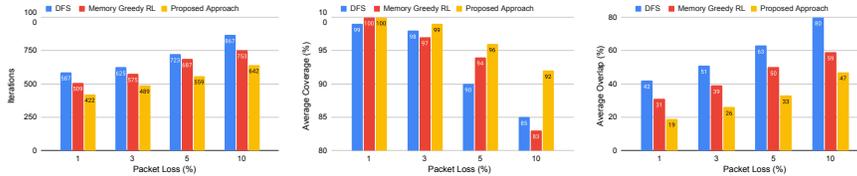


Figure 7.7: Impact of the packet loss parameter; number of iterations (**Left**), average coverage percentage (**Center**), average overlap percentage (**Right**).

rapidly. However, DFS and our approach have comparable completion times for two-thirds coverage. The trend changes towards the end of coverage completion due to a lack of cooperation; DFS takes approx. 100 and 25 more iterations than the proposed and memory-greedy RL approaches. We also have recorded performance microscopically over every iteration. Standalone DFS works better for a few initial iterations for coverage, as its processing does not involve decision-making. However, after half of the maze coverage, the proposed approach can cover the region more rapidly than DFS. Overall, both approaches can cover 99% of the explorable region within 400 iterations for all mazes.

We have analyzed the performance of all approaches for one maze for complete coverage and observed performance for 100 and 400 iterations for all the experiment mazes. Results in Fig. 7.4 have shown that DFS and memory-greedy RL approaches have better coverage percentages than the proposed approach initially at 100 iterations; however, at 400 iterations, the proposed method is approximately 10% better in coverage.

Overlap: We have analyzed the performance based on the inter-robot overlapping coverage space. Results (see Fig. 7.5) have shown that DFS has a higher overlap than the proposed approach at both iteration cycles, which is 3% and 20% for 100 and 400 iterations, respectively. Interestingly, the memory-greedy RL approach balanced the charts by providing a faster coverage rate but at a loss of map overlap ratio.

7.4.2 Ablation studies on communication efficiency

Communication Range: We have varied the communication range from 1500 to 500-pixel units between nodes in the simulator to evaluate the performance of maze coverage strategies over completion time, percentage coverage, and percentage overlap. Fig. 7.6 shows that our approach has the most negligible impact of weak communication for coverage time due to preplanned policy utilization and strong cooperation among robots. Conversely, reduced range drastically affects the memory-greedy RL approach, and DFS moderately affects weak communication on maze coverage time. The coverage percentage reduces with the reduction in range for all approaches differently. DFS can only cover 82% of the overall maze with the 500 range, and memory-greedy RL covers a slightly larger area of 87% maze than DFS. The proposed approach has shown the most negligible impact of content on the coverage and can cover the maximum region even with a range of 500 units. Regarding overlap percentage, robots in the proposed strategy intentionally avoid already explored areas due to the optimized reward function, resulting in less overlap than other strategies. DFS has high overlap during overall coverage and almost re-explored three-quarters of the whole maze in the slightest communication range.

Packet Loss: To evaluate the effectiveness of maze coverage strategies across completion time, percentage coverage, and percentage overlap, we increased the packet loss percentage in the CORE simulator from 1 to 10%. According to the data, the proposed strategy has the most negligible influence on packet loss for coverage time (see Fig. 7.7) since it uses a preplanned policy and has excellent robot collaboration. On the other hand, a higher packet loss of 10% significantly impacts the DFS technique and requires 300 more iterations to cover the labyrinth space. The memory-greedy RL had a minor impact on packet loss on the maze coverage time, but it could still cover the entire area in 753 iterations. Additionally, each technique's average coverage percentage decreases as the packet loss percentage rises. With a 10% packet loss, memory-greedy RL covers 83% of the entire maze, whereas DFS covers a slightly larger area. The proposed method can cover the maximum region (92%) even with a 10 percent packet loss and has the most negligible noticeable impact of range on coverage percentage. Due to the optimized reward function, robots in the proposed method purposefully avoid already investigated locations, resulting in less overlap than other strategies.

7.5 Summary

This chapter introduces a new decentralized reinforcement learning algorithm to facilitate efficient coordination among swarm robots. This novel approach hierarchically uses local information exchanges, making the entire process highly efficient. The method is applied to a maze-solving task conducted by a group of robots to minimize the time, and cost, avoid inter-robot collisions, and prevent path overlaps during the exploration process.

The chapter provides a comprehensive theoretical algorithm analysis and evaluates it against state-of-the-art solutions using CORE network simulations. The results show that the proposed algorithm offers significantly higher coverage accuracy and efficiency even in communication-degraded environments, marked by high packet loss and low communication range. However, this algorithm's application isn't restricted to maze-solving tasks. It can be generalized to other domains where communication efficiency is pivotal in swarm coordination. The algorithm outperforms existing solutions like graph-based Depth First Search (DFS) and memory-greedy reinforcement learning regarding coverage efficiency and minimizing overlapping regions, particularly in environments with compromised communication conditions.

The chapter concludes by suggesting that the promising results open new avenues for further research and extensions of the proposed algorithm to a broader spectrum of swarm robotic network applications. The source code of the proposed algorithm is made available for use and further development by the robotics community. This chapter discussed the exploration strategy for the case study of the maze and validated the efficiency of reinforcement learning-based exploration techniques. The next chapter will explore reinforcement learning with an advanced reward function and efficient map exploration technique for multi-robot systems.

CHAPTER 8

CQLITE: COMMUNICATION-EFFICIENT MULTI-ROBOT EXPLORATION USING COVERAGE-BIASED DISTRIBUTED Q-LEARNING

8.1 Introduction

This chapter further explores the utilization of reinforcement learning with advanced coverage biased reward function for maximum exploration. Map-based coverage and exploration is a significant problem of interest in the robotics and multi-robot systems (MRS) community (Burgard et al., 2005). In this problem, robots continuously explore to obtain the full environmental map in a new bounded environment without prior information. It can be helpful in various applications, including search and rescue, domestic service, survey and operations, field robotics, etc. Autonomous exploration and surveillance solutions can also demonstrate the adaptability of the MRS since robots can carry out these missions in different and uncharted areas.

Recent works have been influential in realizing an efficient exploration objective. For example, information-based methods (e.g., (B. Fang et al., 2019)) typically use the Shannon entropy to describe the uncertainty of the environmental map and construct the optimization problems such that the robot's control variable (e.g., velocity) is continuously optimized during the exploration process. On the other hand, frontier-based methods (e.g., (Dai et al., 2020)) involve deciding the robot's next move (or path) by searching the frontier points

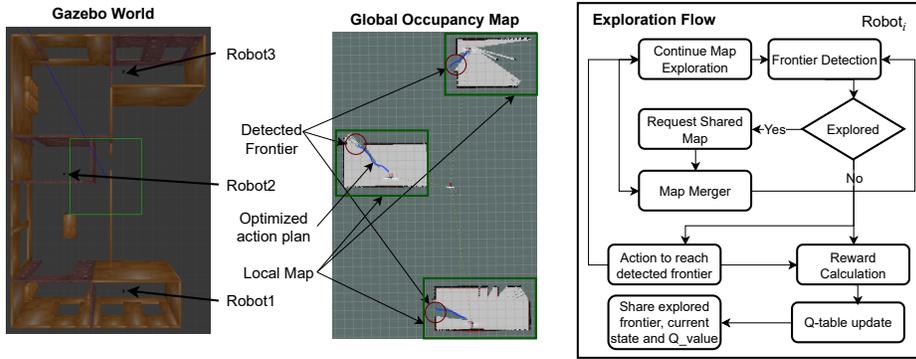


Figure 8.1: Overview of the distributed CQLite method for efficient multi-robot exploration, shown with an illustrative simulation.

on the border of free and unknown points. Often, these methods only produce approximate solutions due to optimization.

Integrating learning with planning solutions is promising, especially for robot exploration (Shrestha et al., 2019). In the reinforcement learning (RL) paradigm, robots can continuously improve competence and adapt to the dynamics of natural surroundings by observing the results of navigational choices made in the actual world (Z. Zhang, Wang, et al., 2022). On the other hand, cooperation among robots in an MRS can help achieve a complex mission through simple distributed approaches (Tolstaya et al., 2021).

This paper explores the intersection between learning and cooperation, designs a combined solution to achieve efficient map exploration, and provides theoretical support for fast convergence and time complexity. We leverage the benefits of learning-based paradigms for joint exploration. We aim to create a distributed algorithm that gains knowledge through robot-robot information sharing while minimizing communication and computing overheads. Specifically, we utilize a distributed Q-learning methodology with a coverage-biased reward function with a light communication and information fusion strategy. In our approach, we reduce communication complexity by sharing only the current state information, i.e., Q-value, instead of the complete Q-table as done in (Sadhu & Konar, 2018) and explored frontier. Fig. 8.1 provides an overview of the proposed method implemented in the Robot Operating System (ROS) framework. The main contributions of this paper are summarized below.

- We propose a novel distributed coverage-biased Q-learning approach (*CQLite*) for efficient multi-robot map exploration with limited data exchanges.

- We substantiate the potential of our method with theoretical guarantees and extensive simulation experiments. We evaluate the performance of our approach against two state-of-the-art (SOTA) multi-robot exploration methods: Rapidly-exploring Random Trees (RRT) for Optimized Exploration (L. Zhang et al., 2020) and Deep Reinforcement Learning (DRL) for Voronoi-based Exploration (Hu et al., 2020).
- We open source¹⁰ the CQLite as a ROS package for use and further development by the robotics community.

¹⁰ <https://github.com/herolab-uga/cqlite.git>

The key idea behind the CQLite uses a coverage-biased reward function to perform efficient exploration by sharing limited information among robots in a distributed fashion. Our method achieves fast convergence with the best coverage performance, reduced communication, and update costs compared to the baselines¹¹.

¹¹ Video of simulation experiments and real-world demonstrations are also available at <https://hero.uga.edu/research/cqlite/>

8.2 Related Work

Map exploration problems focus on frontier-based and learning-based coverage planning approaches. A robot can be greedily pushed in an occupancy grid-map to the closest boundaries (W. Gao et al., 2018) or to the most uncertain (or informative) regions (Bouman et al., 2020). In frontier-based strategies, the robots will look to expand coverage into the unexplored regions by choosing the next waypoints based on the frontier of the explored map boundaries. For instance, in (L. Zhang et al., 2020), the multi-robot map exploration objective is integrated into an optimization framework incorporating Rapidly-exploring Random Trees (RRTs) to increase the effectiveness and efficiency of multi-robot map exploration. However, the constraints of such frontier-based approaches are the computing expense of the optimization methods and the possibility of non-optimal outcomes resulting from RRTs' stochastic characteristics.

In the collaborative map exploration problem, the robot will receive a positive reward when it finds the "treasure," meaning that the reward it just received will allow it to return to the original region numerous times. These prizes, however, only show up one time on the map. This circumstance, known as a "reward loop," significantly extends the time needed to navigate the map. A multi-Q-table Q-learning method is suggested in study (Kantasewi et al., 2019) as a solution to this issue. This approach eliminates recurrent investigation of the area by creating a Q table each time the robot receives a good reward. It also allows the robot to find the "treasure."

Most learning-based autonomous navigation approaches rely mainly on supervised training data. Examples include learning route planners, obstacle detectors, reactive controllers over a map-based planner (Richter & Roy, 2017), and driving affordances (D. Wang et al., 2019). However, many approaches are based on human supervision, which is fundamentally bound by the amount of human data available. The capabilities of intense and expressive models like deep neural networks are frequently constrained significantly by the data available.

Researchers exploited self-supervised learning with generalized graph computation to achieve autonomous map exploration. Gregory et al. (Kahn et al., 2018) provided a generalized computation graph that includes value-based and model-based techniques, with specific instantiations interpolating between model-free and model-based, to address the need to learn complicated policies with few samples. They then created an instance of this network to create a sample-efficient navigation model that can be taught from unprocessed photos. However, in the case of a complex environment, graph computation can become a complicated problem to solve and is not feasible for real-time exploration with resource-constrained robots.

In a recent study, Jinyoung et al. (Choi et al., 2019) examined the potential for depth cameras to take the place of pricey sensor devices. They discussed the impact of the DRL agents' restricted field of view and suggested a local-map Critic designed to teach effective navigation in a challenging environment.

A recent study (H. Li et al., 2019) looked at the issue of automatic exploration in an unfamiliar setting, which is crucial when using a robotic system to perform specific social duties. By breaking down the exploration process into the decision, planning, and mapping modules, research created a general exploration framework that increases the adaptability of the robotic system. They proposed a DRL-based decision algorithm based on that framework, which uses a deep neural network to learn an exploration strategy from a partial map. Similarly, for exploration region selection in the decision-making process, a study (D. Zhu et al., 2018) used DRL. The agent can forecast a long-term visitation order for unknown subregions thanks to the presentation of exploration knowledge over office designs. Based on this method, they suggest an exploration architecture that, to enhance exploration performance further, combines a DRL model, a next-best-view selection strategy, and a structural integrity measurement.

A target-driven indoor visual navigation approach (Y. Zhu et al., 2017) also utilized DRL to address generality and slow convergence. They suggested an actor-critic model handles the generalization problem, whose policy is a function of the aim and the current state and allows for improved generalization.

Finally, they suggested using a third-party framework, which offers a setting with top-notch 3D scenes and a physics engine for quick convergence.

Researchers also presented communication-efficient solutions for exploration in multi-robot systems. For instance, Zhang et al. (Z. Zhang, Yu, et al., 2022) introduced the MR-TopoMap based on a topological map, which can independently explore the robot's surroundings while sporadically exchanging topological maps when communication is possible. But, path planning through topological mapping can lead to a sub-optimal path and, specifically in the case where robots start exploring from the same position, exploring the same map, making it difficult to divide the map into topologies. Corah et al. (Corah et al., 2019) use information-based distributed planning considering communication restrictions. However, the planner's finite-horizon nature could lead to suboptimal exploration paths because it doesn't consider long-term planning beyond the given horizon making it more difficult for the system to make decisions based on knowledge in the future. This might prevent robots from efficiently exploring or discovering key regions of interest. More recently, Gao et al. (Y. Gao et al., 2022) reduced inter-robot communication costs by utilizing a mission-based protocol and centralized planning, where the former can actively disconnect robots to proceed with distributed (independent exploration) and the latter will help them achieve rendezvous to reconnect and share information. However, computing the super-frontier information is computationally expensive, and the active disconnection strategy may limit the robots from sharing other critical data during the mission.

A body of research concentrates on Reinforcement Learning (RL) and Q-learning for multi-robot tasks, modifying the learning mechanism in low communication scenarios for better navigation and exploration (Latif et al., 2023; Serra-Gomez et al., 2020) and utilizing deep reinforcement learning to achieve optimality in robotic exploration (R. Han et al., 2020). However, this method calls for frequent map merging, which raises the cost of updates. A Deep RL (DRL) approach for cooperative multi-robot exploration using Voronoi cells was proposed in (Hu et al., 2020). Despite its intriguing concept, it was constrained by training difficulties and sub-optimal solution tendencies. Further, DRL has shown promise in some problem spaces, but they frequently offer less-than-ideal solutions outside those contexts. They cannot guarantee convergence in infinite horizons.

In research (Uwano & Takadama, 2017), Q-learning is expanded, and an RL for multi-robot cooperative tasks is proposed. The standard reward is transformed into an "internal reward" when robots on the map cannot communicate with one another. This allows the robots to learn following this "internal re-

ward" type and achieve task collaboration under such challenging circumstances. Another study (Pozza et al., 2021) suggests that robots can use the knowledge acquired through learning more effectively by mixing RL with traditional deep neural networks. Compared to classical RL, the algorithm can help the robot find the best way to get out of the map more quickly but requires merging maps on every time step, which increases the update cost.

A unique RL map navigation technique is proposed in research (Yu et al., 2019) to address the unknown map navigation problem for autonomous ground vehicles. First, the picture data of the random map is collected using the quadrotor's bottom-mounted camera. Then the virtual map is created in the simulation environment using an image processing technique. Finally, an enhanced Q-learning algorithm is suggested to address the issue that the original greedy strategy repeatedly forces the robot to linger in the past state.

Though the DRL and Q-learning approaches are popular and solve specific research questions, they provide sub-optimal solutions outside the task space (or domain). They cannot offer convergence in cases of an infinite horizon. DRL approaches address traditional map exploration issues but provide solutions with high computation, communication, and update costs. When it comes to RL, to solve the problems of slow learning speed and non-convergence of traditional Q-learning, many scholars have improved RL in different aspects. In (Hu et al., 2020), a Voronoi-based approach is proposed that uses DRL for cooperative multi-robot exploration. By splitting the environment into Voronoi cells and allocating a robot to each cell for investigation, the authors hope to increase exploration efficiency. However, due to the complexity of the environment and the unknowable mapping of the environment, this strategy is limited by the difficulty of training deep RL models and the possibility of finding less-than-ideal answers. Although some methods cannot accelerate the convergence speed of Q-learning, they provide us with exciting ideas.

Another related topic to the cooperative exploration problems is the cooperative Simultaneous Localization and Navigation (SLAM) approaches. We briefly comment on this literature from the perspective of communication efficiency. For example, Liu et al. (J. Liu et al., 2022) proposed a multi-agent SLAM approach that uses efficient communication to reduce bandwidth consumption but lags in computational efficiency. In contrast, the authors in (Zhao et al., 2021) proposed a lifelong localization and mapping framework that adapts to changing environments but cannot optimize the communication and computational cost for mapping. Bernreiter et al. (Aurangzeb et al., 2013) used spectral graph analysis to enable robots to collaborate on mapping tasks but didn't discuss the computational cost of graph formation and optimization, a

key challenge in real-world applications. On the other hand, the cooperative RL approach proposed in (Jia et al., 2021) faces limitations in terms of computational complexity, as it may become increasingly complex as the number of state spaces increases. Training the RL agents to collaborate in the SLAM tasks may be challenging. These limitations regarding communication have been carefully considered in our proposed CQLite to ensure its practicality in real-world applications and prove computational and communication efficiency by sharing limited data ad-hoc and employing efficient Q-learning to determine optimal exploration strategy.

To address these gaps in the literature, CQLite considers an efficient information transfer mechanism combined with distributed Q-learning with a coverage-biased reward function for achieving communication and computationally efficient multi-robot cooperation to solve map exploration tasks. CQLite departs from RRT (frontier-based) and DRL (learning-based) regarding exploration strategy by reducing recurrent frontier exploration to avoid mapping overlap and Q-learning update strategy for communication efficiency by only sharing and utilizing recently calculated Q-value to the robots, respectively. Additionally, in both RRT and DRL, robots share locally explored maps on every iteration and apply map merging, which gives rise to computational complexity consequently. We reduced this overhead by only sharing and applying map merging in an ad-hoc manner. By incorporating these novelties, our proposed CQLite method addresses the limitations of the above approaches, even in cases of limited multi-robot connectivity.

Another line of research focuses on cooperative simultaneous localization and mapping (SLAM) techniques, which emphasize communication effectiveness. Although computational efficiency is still a problem, Liu et al. (J. Liu et al., 2022) presented a multi-agent SLAM technique that lowers bandwidth use. Others use spectral graph analysis for cooperative mapping but overlook the computational costs of graph formation and optimization (Bernreiter et al., 2022). In contrast, others concentrate on lifelong localization and mapping but fail to optimize the communication and computational cost (Zhao et al., 2021). Cooperative RL techniques, as those in (Jia et al., 2021), have difficulty keeping up with the rising computational complexity of growing state spaces. By delivering computational and communication efficiency through selective data sharing and utilizing effective Q-learning for determining the best exploration strategy, our proposed CQLite method addresses these limitations.

It is worth noting that the objectives of SLAM and exploration approaches are fundamentally different. The SLAM problem focuses on accurately building and merging the map, while the exploration problem focuses on using the

available map to determine waypoints to maximize coverage area. In our work, we use an existing map merging method¹² from the literature to perform multi-robot SLAM. At the same time, our proposed CQLite is designed to maximize exploration with low communication and computation costs.

¹² https://wiki.ros.org/multirobot_map_merge

Section III.A mentioned, coverage biased reward function for Q-learning tends to avoid re-exploration and ensure maximum coverage, which only considers contemporary observations reducing the overall computational cost of policy determination. Section III.B discussed lite cooperation among robots that ensures communication efficiency by only sharing limited data among robots on demand. Generating accurate maps by only sharing and merging maps in case of overlap to avoid re-exploration without frequent feature-based map merging makes CQLite a more efficient mapping approach than state-of-the-art cooperative SLAM approaches, which is also theoretically and experimentally proved in the paper.

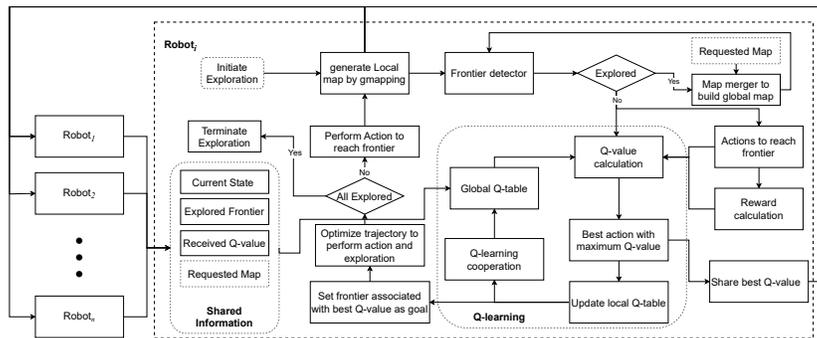


Figure 8.2: System architecture of CQLite distributed across several robots. It shows the Robot i 's process showing the mapping, frontier detection, and Q-learning operations along with the communication of local map and updated Q-value information to n connected robots.

8.3 Proposed Approach

The problem formulation employed here deploys many robots at random starting locations in an unknown environment. The robots must navigate towards the frontier position detected by local sensing information as a standard map exploration strategy. To accomplish this efficiently, a robot must decide which frontier to navigate after leaving its current explored region. In doing so, it is hoped to reduce the number of steps to take and the size of data exchanges with connected robots while considerably enhancing the effectiveness of each robot's random exploration. Robots only share updated Q-value and the newly explored frontier with other robots. Each robot keeps track of its local and shared

frontiers to avoid re-exploration. Robots continue to generate local maps and share the newly developed map only when asked by other robots in case of the already explored frontier. Robots who cannot find new frontiers merge their local map with a map received from peers to build a global map using the feature similarity-based map merging technique (Mangelson et al., 2018). The robot's decisions regarding an action plan are based on the shared information and Q-learning computation.

8.3.1 Q-learning

Markov's decision processes frequently model the robot's interaction with the environment. A robot's state is $(x, y, \theta, \text{active/inactive})$ in a global frame. Robots are localized and initialized in a global frame, and positions are known concerning virtually defined bounded regions which can be expanded based on exploration requirements. We consider the frontier's position as states for exploration by applying efficient frontier detection (Keidar & Kaminka, 2014). A robot can transition from state $s_t \in S$ to state s_{t+1} due to acting $a_t \in A$ based on its state at time t . Robot action a_t to reach s_{t+1} from s_t can be determined using discrete-time Hopfield function (Uykan, 2019) as:

$$a_t = \sum_{j=1}^N (s_{t,j-1} - s_{t,j})^2 + 2 \sum_{j=1}^N (\theta_{t,j}) \quad (8.1)$$

where $s_{t,0} = s_t$, $s_{t,j}$ are the intermediate states to reach s_{t+1} , θ is the orientation of state s_{t+1} from the state s_t , and N is the number of intermediate states which are extracted by applying the shortest path algorithm (J. Li, 2020). The transition probability is defined as $T : S \times A \times S \rightarrow [0, 1]$. The robot will receive a reward for each action using a reward function $R : S \times A \times S \rightarrow R$ specific to the task. The robot will have learned the course of action to take in each state and will be able to maximize the reward of the entire interaction process.

In Q-learning, all possible states and actions are created using the Q table, which then updates each value through iterative learning. The robot then chooses the best course of action for each state based on the values in the table. This approach is frequently utilized in path planning, chess, card games, and other activities.

Concepts used in Q-learning are as follows:

- $S = \{s_1, s_2, s_3, \dots, s_n\}$ is a discrete set of n states, where $s_t \in S$ describes the state of the robot in the environment at a time t .

- $A = \{a_1, a_2, a_3, \dots, a_n\}$ is a discrete set of n actions, where $a_t \in A$ describes the action which the robot chooses at a time t .
- $T : S \times A \times S \rightarrow [0, 1]$ is a stochastic state transition function, where the state of the robot is transitioned to state s^* with a probability $p \in [0, 1]$ when choosing action a in state s . We use $s^* \leftarrow T(s, a)$ to represent the above process.
- $R : S \times A \times S \rightarrow R$ is the reward function. It represents the reward the robot obtains in its state transition to s^* after the robot executes action a in state s .
- $\gamma \in [0, 1)$ is the discount factor is the relative importance of future and present rewards.

Assumptions: For simplicity, we assume a flat ground terrain environment for exploration.

- The environment is both globally closed and locally opened, preventing the robot from escaping the overall amount of space available to it and from becoming stuck in a particular area like a room.
- To exchange data, the robots are constantly in full wireless communication with a centralized computer.
- The robot has an RGB-D camera to detect other robots and a range finder sensor to detect its surroundings.
- Other robots can recognize the robot by looking for the fiducial tags printed on its body.
- Although the initial placements of the robots are uncertain, they must be dispersed first so they do not immediately run into one another.
- The robots have an omnidirectional sensory system that can detect the boundary of an obstruction within the maximum sensing range r_s and provides a description of the open space around the robot.
- Each robot has a communication range $r_c \gg r_s$ that it can use to broadcast the data stored in its memory. The robot can constantly receive information about its relative position from a neighbor robot inside the r_c communication range.

- Robots are connected through the wireless communication channel and assumed to form a connected graph throughout exploration, which is practical to achieve in a multi-robot application. Nevertheless, the proposed solution is distributed and ensures maximum coverage even in partial disconnectivity with the trade-off in time of exploration and re-exploration.
- The sensing range r_s is smaller than the communication range r_c .

Here, we introduce CQLite as a distributed method for robot i , which is now at state s_t at time t and selects the following state as s_{t+1} to explore independently. Finding the action a that maximizes the Q-value for a specific state s is the goal of the maximum optimization function for Q-learning, i.e.,

$$\begin{aligned} a^* &= \underset{a}{\operatorname{argmax}} Q_i(s, a), \\ &= \underset{a}{\operatorname{argmax}} \left[(1 - \alpha) Q_i(s, a) + \alpha [r_{i,t} + \gamma \max_{a'} Q_i(s', a')] \right] \end{aligned}$$

where a^* is the optimal action for a given state s .

The Q-learning algorithm updates the Q value as

$$Q_{i,t+1}(s_t, a_t) = (1 - \alpha) Q_{i,t}(s_t, a_t) + \alpha [r_{i,t} + \gamma Q_{i,t}(s_{t+1}, a^*)], \quad (8.2)$$

where $r_{i,t}$ is the reward received for taking action a in state s_t . The $\alpha \in (0, 1]$ controls the balance between the coverage and delay, and γ is the discount factor to prioritize present vs. future rewards. This optimization function is used in the action selection step of the Q-learning, where the agent selects the action that maximizes its expected future reward.

The objective of the CQLite is to perform maximum coverage in less time and avoid overlapping exploration, which can be numerically defined as

$$\max_{\pi} \{P_a^{\pi}(t) - \lambda_i E_t(a|\pi)\}, \quad (8.3)$$

where $P_a^{\pi}(t)$ is the probability to cover the unexplored region using for action a using policy π at time t , $E_t(a|\pi)$ is estimated time to reach the state s_t by taking action a at time t in policy π and λ_i is the cost associated with each step taken by robot i . We have a vector *path* extracted by containing position waypoints connecting s_t to s_{t+1} associated with a (J. Li, 2020). For each dimension of

path at each control instant $t = t_j$, we first compute the velocity command as:

$$v_{t_j} = K_p \cdot e_{t_j} + K_I \sum_{t=t_0}^{t_j} (e_{t_j}), \quad (8.4)$$

where $e_{t_j} = s_{t,j} - s_{t,j-1}$ represents the instantaneous error between the intermediate states associated with action a (i.e., the feedback) at time $t = t_j$. Further, K_p and K_I are the so-called proportional and integral gains of the motor controller that regulate the contributions of the corrections induced by the actual error and the error accumulated over time, respectively. These constant values determination is based on the motion constraints of our differential drive robots as discussed by Li et al. (J. Li, 2020). They can be different based on the robot's physical and motion characteristics. In our case, we predetermined values of K_p and K_I as 2 and 0.5, respectively. Now we apply simple kinematic to find the estimate $E_t(a|\pi)$ as:

$$E_t(a|\pi) = \sum_{j=1}^N \frac{(e_{t_j})^2}{v_{t_j}} \quad (8.5)$$

To avoid the exploration of already explored region for state s_t , we determine $P(s_t \cap ES_t)$ as:

$$P(s_t \cap ES_t) = \sum_{j=1}^m \frac{P(s_t \cap es_j)}{m} \quad (8.6)$$

Where $es_j \in ES_t$, and m is number of explored state in ES_t and overlap probability of each explored state in ES_t can be determined as:

$$P(s_t \cap es_j) = \begin{cases} 1 & \text{dist}(s_t, es_j) \leq r_{i,s} \\ 0 & \text{dist}(s_t, es_j) > r_{i,s} \end{cases} \quad (8.7)$$

At each discrete time step t , the robot i acquires an observation s_t from the environment, selects a corresponding action a_t , then receives feedback from the environment in the form of a reward $r_{it} = R(s_t, a_t)$ as shown below:

$$r_{it} = \begin{cases} -\lambda_i & s_t \in ES_t \\ \lambda_i - Q_{i,t} + \rho(1 - P(s_t \cap ES_t)) + \sigma r_{i,c} & s_t \notin ES_t \end{cases} \quad (8.8)$$

Where $P(s_t \cap ES_t)$ is the probability of overlap between the current state s_t , and the already explored states ES_t by *robot_i* and other robots, ρ is a scaling factor that controls the importance of minimizing the overlap, r_c is the communication range, and σ is the scaling factor that determines the importance

of maximizing the communication range. σ depends upon the robot’s sensing capabilities and makes the reward function modular for heterogeneous robots with different sensing capabilities. Then the state information is updated s_{t+1} . The goal of the RL is to select policy π that maximizes the discounted sum of future rewards, i.e., $Q_\pi(s_1) = \sum_{\tau=1}^{\infty} \gamma^\tau R(s_t, a_t)$, which according to the Bellman optimality principle satisfies.

The reward function in Eq. (8.8) produces a negative reward whenever the agent has looped back, and the calculated reward is based on the step-cost, Q-value, probability of overlap, and scaling factor otherwise.

Multi-Robot Lite Cooperation: We reduce the communication overhead amongst individual exploration-capable robots through a distributed approach, allowing each robot to make independent decisions based on local information and with little interaction from other robots. In our lite version of Q-learning, only the current state and Q-value are communicated amongst nearby robots to encourage cooperation. When another robot receives the information, it will update the received Q value in its Q table and update the local map. We develop a discovery approach based on the distance between simulated robots to replicate the network range in which we only share the current position of a robot i , its Q-Value for each direction, and mark the current situation as explored to avoid repetitive exploration.

8.3.2 Exploration Strategy

Robots create a global Q table for each cell and action after searching the map and experiencing several experiences. The Q table is then turned into a weighted graph $G = (\mathcal{S}, \mathcal{E}, \mathcal{C})$, where $\mathcal{S} = \{s_1, s_2, \dots, s_n\}$ denotes the set of states, and $\mathcal{E} \in |\mathcal{S}| \times |\mathcal{S}|$ signifies the set of edges whose elements indicate whether or not a path exists between the center points of each pair of states. It is assumed that robots do not exchange nodes during exploration, and Voronoi boundaries are fixed. Furthermore, \mathcal{C} is the weight matrix indicating the edge metric cost. The primary goal of discovering this study’s reduced graph and significant states is to optimally disperse robots over the coverage region by minimizing the relevant cost function. Because robots move at varying speeds, we formulate the cost as a function of the defined traveling time as $t_{(s_{p_i}, s_q)} = \frac{d_{(s_{p_i}, s_q)}}{v_i}$, where v_i is the i^{th} robot’s speed, and $d_{(s_{p_i}, s_q)} \in \mathcal{C}$ is the Euclidean distance between the i^{th} robot’s current state p_i and state q . Furthermore, knowing the optimal path from state p_i to state s , each robot’s overall optimal traveling time is the sum of the trip times (costs) from state p_i to state s . This study’s shortest path between each pair of states is computed using the A* algorithm. Then the total time τ is

calculated by knowing path $\mathcal{P} = \{p_1, p_2, \dots, p_n\}$ as

$$\tau_{(s_p, s_q)} = t_{(s_p, s_{p_1})} + t_{(s_{p_1}, s_{p_2})} + \dots + t_{(s_{p_n}, s_q)}. \quad (8.9)$$

After determining the shortest time between each pair of states, the field is partitioned into M Voronoi subgraphs g_{r_i} for $i \in \{1, 2, \dots, M\}$ to distribute work proportionally among M robots. To that aim, the ideal Voronoi diagram g_{r_i} for i^{th} robot, according to Lloyd's algorithm, is a split of the area determined as:

$$g_{r_i} = \{s_q \in S \mid \tau_{(s_{p_i}, s_q)} \leq \tau_{(s_{p_j}, s_q)}, \forall i \neq j\}. \quad (8.10)$$

Where j is the other connected robot. The i^{th} robot is responsible for covering the state s (associated robot) in its sub-graph g_{r_i} using the Voronoi partitioning result. The entire cost is then calculated as

$$\lambda_{i, (p, g_r)} = \sum_{j=1}^m \sum_{q \in g_{r_i}} \tau_{(s_{p_j}, s_q)} \phi_q, \quad (8.11)$$

where ϕ_q is the priority value associated with state s_q . As the map turns into a graph, higher priority values are assigned to target states, while lower priority values are assigned to states far and already explored from the current state. The entire travel time (cost) will therefore be minimized, and an optimal solution will be obtained only when the current distance between the robot i and the target state s_q , $d_{(s_{p_i}, s_q)}$ converges to zero. Algorithm II provides the pseudocode description of CQLite for efficient map exploration implemented in a distributed manner on each robot i .

Algorithm II: Distributed CQLite Exploration

```
1 Data: Reward matrix  $R$ , learning rate  $\alpha$ , discount factor  $\gamma$ , step cost  $\lambda_i$ ;  
2 Number of iterations:  $t = 0$ ;  
3 Initialize empty Q-table as  $Q_i$  for robot  $i$ ; Initialize empty explored  
  frontier list  $EF_t$ ;  
4 Generate a local map using range sensor;  
5 Initialize Explored Frontier Detected  $f_d$  as 0;  
6 while ( $t \leq t_{max}$ ) and  $f_d < 2$  do  
7    $S_t \rightarrow S_{start}$ , step = 1;  
8   Find new frontiers at new  $ES_t$  and update  $F_t$  using locally  
  explored map;  
9    $Q_{i,t} = \text{list}(0)$ ;  
10  for each frontier  $f$  in  $F_t$  do  
11    if  $f$  not in  $EF_t$  then  
12      calculate Q-value as  $q_f$  for actions  $a_f$  to reach frontier  $f$   
        using Eq. (8.2);  
13      Append  $Q_{i,t}(s = f, a_f)$  with  $q_f$  for action  $a_f$ ;  
14    else  
15      request for explored Maps from connected robots;  
16      merge maps into local maps;  
17       $f_d = f_d + 1$ ;  
18    end  
19  end  
20  if  $f_d < 2$  then  
21    set updated Q-value  $q_{update}$  as  $\max_a(Q_{i,t}(s, a))$ ;  
22    Update  $Q_i$  with  $q_{update}$ ;  
23    Take action  $a_t$  associated with  $Q_i$ ;  
24    Share  $q_{update}$  with connected robots;  
25    Receive Q-value for  $ES_t$  from connected robots;  
26    Receive explored frontiers  $ef_{1:n-1}$  from connected robots and  
      update  $EF_t$  with  $f_t$  and  $ef_{1:n-1}$ ;  
27    set new state associated with  $a_t$  as  $S_t$ ;  
28    Reset  $f_d$  as 0;  
29  end  
30 end
```

8.3.3 Convergence Analysis

We analyze the convergence of the target Q value update function Eq. (8.2). We denote the error ratio $\delta_t = \frac{MSE(Q_t)}{E_t(a|\pi)}$, where $MSE(Q_t)$ is the calculated mean square error for Q-table at time t and $E_t(a|\pi)$ is the average number of steps to cover the region by taking action a at time t for policy π .

Theorem 1 (Convergence of Q-values): Using Eq. (8.2) for Q-value updates, then if $0 \leq \delta_t \leq 1$, with probability $1 - e$, we have the estimated time to reach a given state as:

$$E_t(a|\pi) \leq \omega E_1(a|\pi) + \sqrt{\frac{\ln(1/e) \sum_{i=0}^{t-1} \psi_i^2(\delta_{t-i:t})}{2}} \quad (8.12)$$

Here, $\psi_i(\delta_{t-1:t}) = \frac{\prod_{j=t-i}^{t-1} (j+\gamma\delta_j)}{\prod_{j=t-i}^t j}$, $\alpha_t = \frac{\prod_{j=1}^{t-1} (j+\gamma\delta_j)}{\prod_{j=2}^t j}$ and $\gamma = 0.95$.

Proof. our analysis is derived based on the subsequent synchronous Q-learning. In contrast to the conventional synchronous Q-learning, we swap out the current Q_t for the independent Q-function $Q'(s, a)$ for the target $Q_t(s, a_t)$.

$$\begin{aligned} Q_t(s_t, a_t) &= \left(\frac{t-1}{t}\right) Q_{t-1}(s, a) + \frac{1}{t} (r_t + \gamma \max_{a'} Q'_{t-1}(s', a')) \end{aligned}$$

Note that if $Q'_t(s, a) = Q_{source}^*$, we know that $0 \leq \delta_t \leq 1$.

First, we break down the update rule into:

$$\begin{aligned} Q_t(s_t, a_t) &= \left(\frac{t-1}{t}\right) Q_{t-1}(s, a) + \frac{1}{t} (r_t + \gamma \max_{a'} Q'_{t-1}(s', a')) \\ &= \left(\frac{t-1}{t}\right) Q_{t-1}(s, a) + \frac{1}{t} (r_t + \gamma \max_{a'} Q'_{t-1}(s', a')) \\ &\quad + \gamma \max_{a'} Q_{t-1}^*(s', a') - \gamma \max_{a'} Q'_{t-1}(s', a') \end{aligned}$$

Let $\epsilon_t(s, a) = Q_t(s, a) - Q^*(s, a)$ and

$\xi(s') = \gamma \times \max_{a'} (Q_{t-1}^*(s', a'))$ then recall the definition of δ_t , we will have

$$\begin{aligned} &\leq \frac{t-1}{t} \epsilon_{t-1}(s, a) + \frac{1}{t} (\xi(s') - E_{s'} \xi(s')) + \frac{1}{t} \gamma \delta_t \epsilon_{t-1}(s', a') \\ &\leq \frac{t-1}{t} \epsilon_{t-1}(s, a) + \frac{1}{t} (\xi(s') - E_{s'} \xi(s')) + \frac{1}{t} \gamma \delta_t E_{t-1} \end{aligned}$$

As we know $\varepsilon_t(s, a) \leq E_t$, by applying maximization and recursion of E , we will have:

$$\begin{aligned}
E_t &\leq \frac{t-1 + \gamma\delta_t}{t} E_{t-1} + \frac{1}{t} (\xi(s') - E_{s'}\xi(s')) \\
&\leq \frac{\prod_{j=1}^{t-1} (j + \gamma\delta_j)}{\prod_{j=2}^t j} E_1 + \sum_{i=1}^{t-1} \frac{\prod_{j=t-i}^{t-1} (j + \gamma\delta_j)}{\prod_{j=t-i}^t j} \times (\xi(s') - E_{s'}\xi(s')) \\
&= \alpha_t E_1 + \sum_{i=1}^{t-1} \psi_i(\delta) (\xi(s') - E_{s'}\xi(s'))
\end{aligned}$$

According to weighted Hoeffding inequality (Duda et al., 2014), with probability $1 - e$, we can prove Eq (8.12) for Theorem 1. \square

This convergence result demonstrates the influence of the error ratio on the convergence rate. In other words, learning will go more quickly for our chosen Q value update function. Even though CQLite shares only updated Q-value, it still achieves the required convergence and provides an optimal strategy for robots to explore the map efficiently.

Proposition 1 (Efficiency of CQLite): The CQLite Exploration method enhances efficiency in terms of computation and communication, as well as mapping. It does this by adopting a selective Q-table updating strategy that reduces communication and computation cost to $\frac{1}{n}$ of the cost of the state-of-the-art (SOTA) approaches and a probabilistic map sharing and merging strategy that significantly reduces the frequency of mapping operations.

Proof. For computational and communication efficiency, CQLite uses a strategy where only updated Q-values and newly discovered frontiers are shared and appended to the local Q-table. This leads to a decrease in communication and computation costs by a factor of $\frac{1}{n}$, where n is the size of the Q-table or the total number of possible states. Hence, the cost relation is given by:

$$C_{i,CQLite} = \frac{1}{n} C_{i,SOTA},$$

herein, $C_{i,SOTA}$ is the cost of updating and sharing the whole Q-table in SOTA exploration methods, and $C_{i,CQLite}$ is the corresponding cost in the CQLite Exploration method.

The cost of sending the entire Q-table Q over a network in SOTA methods is given by:

$$C_{i,SOTA} = \kappa \cdot \sum_{j=1}^n |Q_{i,j}|,$$

In CQLite, only updated Q-values and newly found frontiers, represented by a matrix Q' , are shared, leading to the cost:

$$C_{i,CQLite} = \kappa \cdot \sum_{j=1}^n |Q'_{i,j}|,$$

Since Q' is a subset of Q , we can conclude:

$$C_{i,CQLite} \leq \frac{1}{n} C_{i,SOTA},$$

For mapping efficiency, the frequency of map merging operations in CQLite, f_{CQLite} , is given by:

$$f_{CQLite} = P(s_t \cap ES_t) \cdot iterations,$$

where $P(s_t \cap ES_t)$ is the probability of overlap between the current state s_t and the already explored states ES_t . Using Bayes' theorem, we can express $P(s_t \cap ES_t)$ as:

$$P(s_t \cap ES_t) = P(ES_t | s_t) \cdot P(s_t),$$

Assuming the exploration process to be a random walk, we can represent $P(s_t)$ as a uniform distribution over the state space:

$$P(s_t) = \frac{1}{n}.$$

The conditional probability $P(ES_t | s_t)$ can be estimated as the frequency of occurrence of the current state s_t in the already explored states ES_t , denoted as f_{s_t} :

$$P(ES_t | s_t) = \frac{f_{s_t}}{n_e},$$

where n_e is the total number of states in ES_t , substituting the expressions for $P(ES_t | s_t)$ and $P(s_t)$ into $P(s_t \cap ES_t)$, we get:

$$P(s_t \cap ES_t) = \frac{f_{s_t}}{n_e} \cdot \frac{1}{n} = \frac{f_{s_t}}{nn_e}.$$

Because $f_{s_t} < nn_e$ and nn_e is equal to the total number of iterations if each state is visited in each iteration, the frequency of updates in CQLite is less than the total number of iterations. Hence, it can be concluded that:

$$f_{CQLite} = \frac{f_{s_t}}{nn_e} \ll iterations.$$

This shows that the frequency of map merging in the CQLite exploration method is significantly less than in state-of-the-art exploration methods, which confirms that the CQLite exploration method is more efficient in computation, communication, and mapping operations. \square

Proposition 1 (Q-table Update Efficiency): The CQLite Exploration method reduces the communication and computation cost for exploration by sharing and appending only updated Q-values and newly discovered frontiers to the local Q-table, which reduces communication and computation cost by $\frac{1}{n}$ than the cost of the SOTA approaches. Where n is the total number of possible states (size of Q-table).

Proof. First, we prove the efficiency of CQLite by updating and sharing only one value in the Q-table. The shared Q-value for a given state-action combination (i, j) in the Q-table will be $Q_{i,j}$. The CQLite Exploration approach reduces the size of the table and the amount of data that needs to be transmitted between robots by sharing and appending only the updated Q-values and recently found frontiers to the Q-table.

CQLite only updates Q-value once during the whole exploration, in contrast to SOTA as it updates each value in every iteration. Compared to sharing and updating the whole Q-table, the communication and computing costs are decreased by $\frac{1}{n}$. The update cost of CQLite for Q-table with size n is n , but the update cost of SOTA is n^2 ; hence the cost reduction relation can be written as:

$$C_{i,CQLite} = \frac{1}{n}C_{i,SOTA},$$

where $C_{i,SOTA}$ is the communication and calculation cost of updating and sharing the whole Q-table in SOTA exploration techniques, and $C_{i,CQLite}$ is the communication and computation cost of the CQLite Exploration method.

To further determine the effectiveness of the Q-table update in CQLite, the cost of sending the matrix Q over a network can be used to indicate the cost of sharing and updating the whole Q-table and can be stated as follows:

$$C_{i,SOTA} = \kappa \cdot \sum_{j=1}^n |Q_{i,j}|,$$

where $|Q_{i,n}|$ is the absolute value of the Q-value of state j for robot i , and κ is a constant that denotes the cost of sending one unit of data across the network. SOTA requires all Q-values for policy determination; hence all Q-values are shared to update Q-table in every iteration.

The CQLite Exploration approach reduces the size of the matrix and the quantity of data that needs to be transferred by sharing and appending only the updated Q-values and newly found frontiers. Let Q' be the updated matrix that only includes the new frontiers and updated Q-values. This modified matrix's transmission cost can be expressed as follows:

$$C_{i,CQLite} = \kappa \cdot \sum_{j=1}^n |Q'_{i,j}|$$

Since Q' is a subset of Q , it can be concluded that $\sum_{j=1}^l |Q'_{i,j}| \leq \sum_{j=1}^l |Q_{i,j}|$, and therefore:

$$C_{i,CQLite} \leq \frac{1}{n} C_{i,SOTA}$$

The above derivation proves that the CQLite exploration approach is more efficient regarding Q-table updating than the SOTA exploration methods like RRT and DRL. \square

Proposition 2 (Mapping Efficiency): CQLite performs map sharing and merging with the probability $P(s_t \cap ES_t)$, which requires \ll *iterations* compared to relevant SOTA exploration approaches (e.g., RRT and DRL) for maximum exploration. Here, $P(s_t \cap ES_t)$ is the probability of overlap between the current state s_t , and the already explored states ES_t by *robot_i* and other robots *iterations* is the total number of iterations carried out by the algorithm.

Proof. The probability of overlap $P(s_t \cap ES_t)$ between the current state s_t of robot i and the previously explored states ES_t by other robots is used to determine if map sharing and merging will take place in the CQLite Exploration technique. This map merging and sharing aims to reduce the number of iterations and steps the algorithm must perform.

As part of the CQLite Exploration approach, the algorithm updates the map by combining shared maps regularly as follows:

$$f_{CQLite} = P(s_t \cap ES_t) \cdot \textit{iterations}$$

Where f_{CQLite} is the frequency of map merging carried out by the algorithm in the CQLite Exploration method, and *iterations* is the mapping frequency of SOTA exploration methods like RRT and DRL.

The probability $P(s_t \cap ES_t)$ can be derived using Bayes' theorem as follows:

$$P(s_t \cap ES_t) = P(ES_t | s_t) \cdot P(s_t)$$

Given the previously explored states ES_t , $P(ES_t | s_t)$ is the conditional probability of the current state s_t , and $P(s_t)$ is the probability of the current state s_t .

$P(s_t)$ can be represented as a uniform distribution over the state space, assuming that the exploration process is a random walk, with:

$$P(s_t) = \frac{1}{n}$$

where the state space's overall state count is n .

The frequency of occurrence of the current state s_t in the previously investigated states ES_t can be used to estimate the conditional probability $P(ES_t | s_t)$. If the frequency with which the present state s_t occurs in the previously studied states ES_t is f_{s_t} , then:

$$P(ES_t | s_t) = \frac{f_{s_t}}{n_e},$$

where n_e is the total number of states in the already explored states ES_t . Substituting the above expressions into the equation for $P(s_t \cap ES_t)$ gives:

$$P(s_t \cap ES_t) = \frac{f_{s_t}}{n_e} \cdot \frac{1}{n} = \frac{f_{s_t}}{nn_e} \ll \textit{iterations}$$

For the total number of *iterations* CQLite only updates the map for $\frac{f_{s_t}}{nn_e}$ times and $f_{s_t} < nn_e$ and nn_e is equal to the *iterations* in case of visiting each state at each *iteration*. Hence the above derivation proved that the CQLite method is more efficient as CQLite's update frequency is $\ll \textit{iterations}$ in map sharing and merging than SOTA exploration methods like RRT and DRL. \square

8.3.4 Time Complexity

The primary distinction between the various map coverage algorithms is the method to determine the best policy. A robot educated using the Q-learning algorithm can choose the optimum policy by consulting the trained Q-value table—the method of selecting the optimum policy in a self-centered way. Most solve the best approach for the conventional Q-learning-based map coverage algorithm following the current coverage state. The solution procedure for policy determination mainly determines the computational difficulty of such an algorithm, and the computational complexity of the control model can be disregarded. Assume that the grid factor is k_g (resolution of the grid map on which the grid is divided) and that the target sub-map is $k \times l$ in size. The grid map's size is $k_g k \times k_g l$, and the total number of points is $k_g^2 kl$. The operations to find

Q-values must be carried out cyclically $k_g^2 kl$ times. kl can calculate and represent the CQLite's state space size; however CQLite doesn't perform a merging and searching strategy at every iteration; hence the length of the Q-value table is significantly less than (kl) through selecting the training process, and the computing complexity of the algorithm is considerably less than $O(kl)$. M_i is used in the computation as a history of the agents' positions, and the size of the information map and the number of agents determine the state's capacity. After all the states have been trained, the best policy can be found by choosing the action space with the highest Q-value. This is done by querying a list of length (kl) to determine the Q-value of the current state. As a result, searching a list and locating the highest Q-value in the action space constitute the computational complexity of the Q-Traversal algorithm. In actuality, both full search and list query comprises various comparison processes. The list length and the comparison data's nature influence how many comparisons are made.

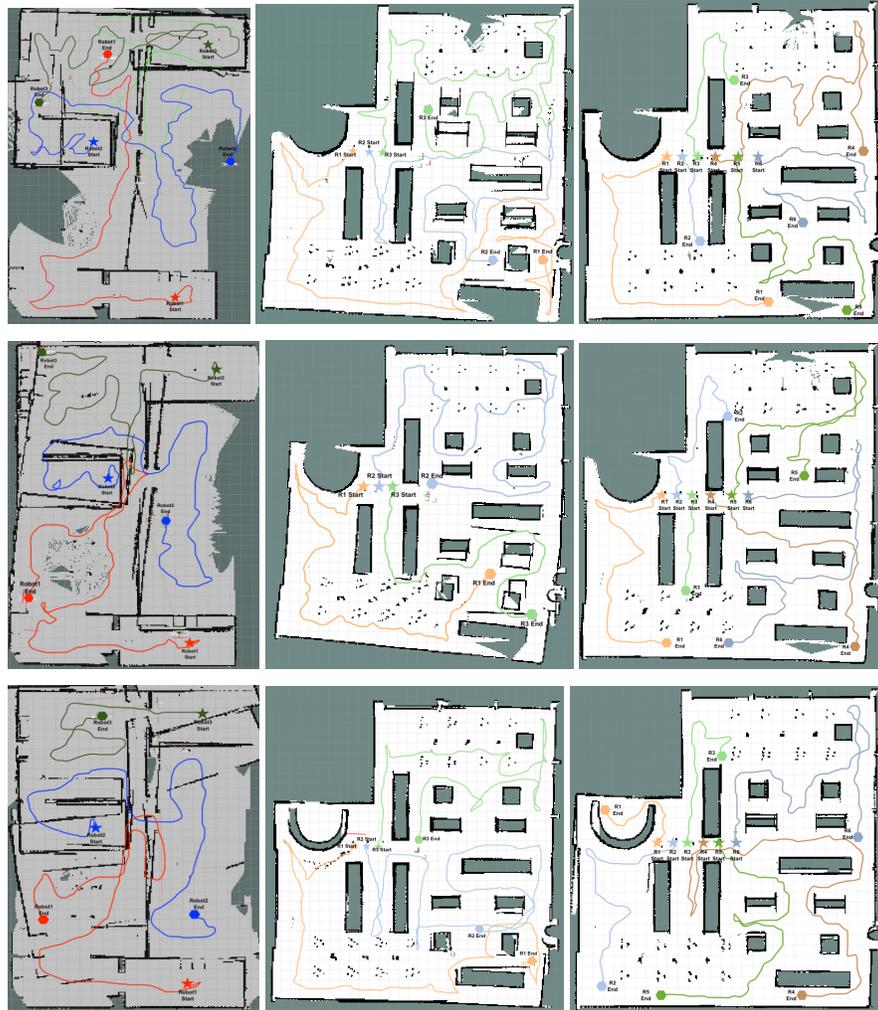


Figure 8.3: A depiction of the outcome in a sample trial. It shows the map generated by three robots in the house world (left column) and three and six robots in the bookstore world (center and right column, respectively) created by the three compared approaches; RRT (top), DRL (center), and CQLite (bottom), with robots moving in a simulated House and Bookstore worlds along with the following trajectories, start and end locations.

8.4 Experimental Validation

Turtlebot3 robots are used to carry out the exploration plan, implemented using the ROS framework. The Kobuki base of the Turtlebot offers odometer readings, cliff detection sensors, bumpers, and battery voltage readings. The open-source *openslam-gmapping*¹³ technique of the ROS *gmapping* package is used to create 2D maps. It uses odometer data and a particle filter method as its foundation. The local maps created by each robot are combined to create

¹³ <https://openslam-org.github.io/gmapping.html>

the global map. Feature-based map merging¹⁴ is employed to merge maps when required. Frame conversion between the local map frames is necessary for map merger. The coordinate transformation correlation between the robots must be calibrated before combining local maps. In the current work, the global frame is one robot's frame, and the relative positions and orientations of the robots are initialized to a known state. The ROS *movebase*¹⁵ package allows the robot to move toward the goal point while securely avoiding barriers between robots. The Dijkstra algorithm for global path planning and the Dynamic Window Approach (DWA) for local dynamic obstacle avoidance are both implemented in this package. In this study, the units of time and distance are in seconds and meters, respectively.

¹⁴ http://wiki.ros.org/multirobot_map_merge

¹⁵ <https://github.com/ros-planning/navigation>

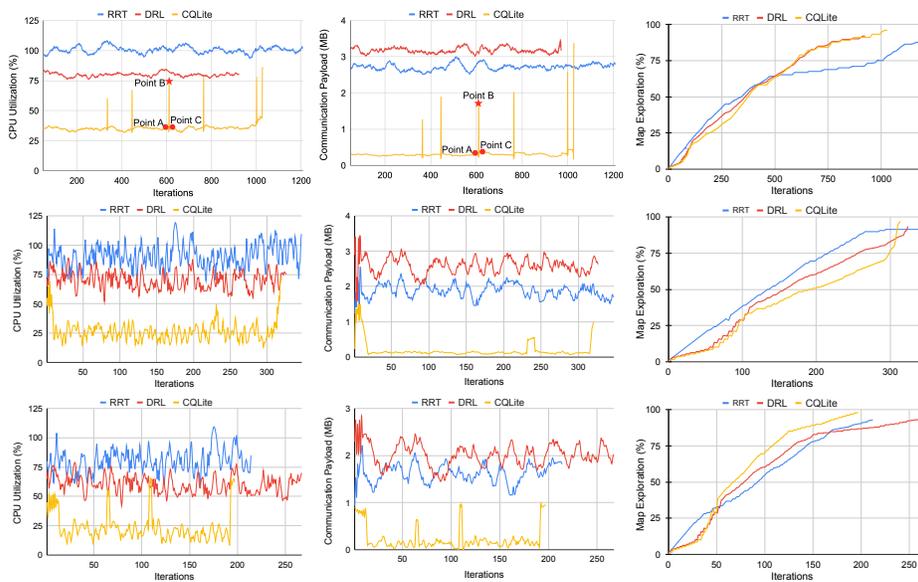
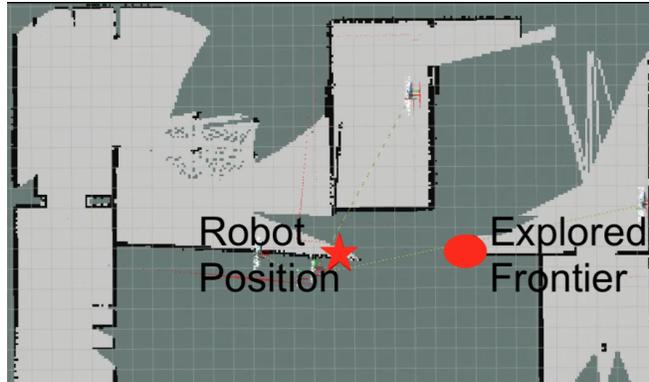
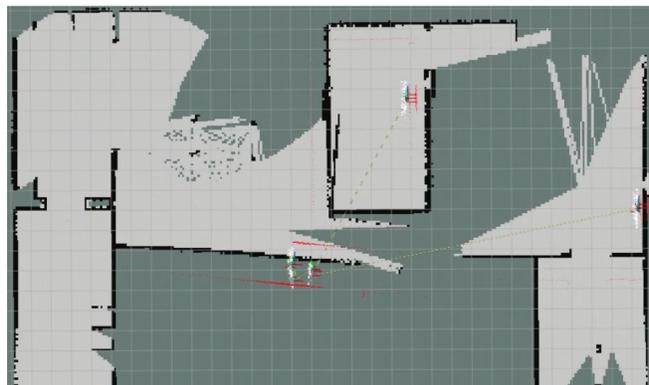


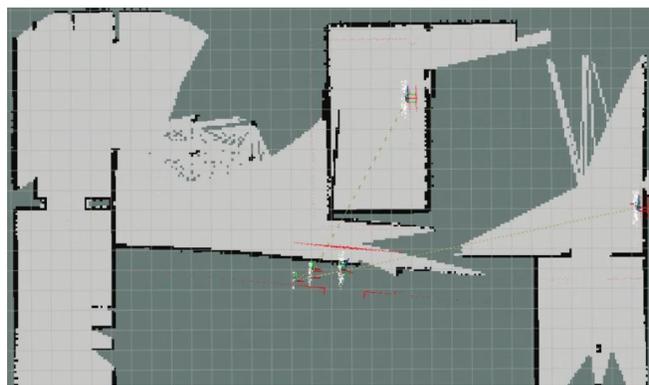
Figure 8.4: Computation (**Left**), Communication (**Center**) cost, and Exploration over time (**Right**) comparison plot of CQLite with RRT and DRL approach in three Gazebo simulated world. Row-wise: **Top** 3 robots in house world, **Middle** 3 robots in bookstore world, and **Bottom** 6 robots in bookstore world.



Point A: Explored Frontier Found



Point B: Map Sharing and Merging



Point C : After Map Merging

Figure 8.5: Exploration map of house world before, during, and after map sharing and merging corresponds to points (A, B, and C from Fig. 8.4) in Computation and Communication plots. Peaks demonstrate the request for map merging in CQLite for Computation and communication plots; RRT runs longer with persistent high communication and computational overhead but explores fewer regions than DRL and CQLite.

Table 8.1: Performance Results of RRT, DRL, and the proposed CQLite. * indicates the best performer.

Evaluation parameters	Three robots in house world			Three robots in bookstore world			Six robots in bookstore world		
	RRT (L. Zhang et al., 2020)	DRL (Hu et al., 2020)	CQLite (ours)	RRT (L. Zhang et al., 2020)	DRL (Hu et al., 2020)	CQLite (ours)	RRT (L. Zhang et al., 2020)	DRL (Hu et al., 2020)	CQLite (ours)
Mapping Time (s)	1208 ± 52	994 ± 67	1029 ± 59	337 ± 32	334 ± 21	397 ± 19	212 ± 18	267 ± 29	397 ± 13
Path Length (m)	592 ± 11	604 ± 19	545 ± 9	278 ± 26	235 ± 29	347 ± 21	223 ± 12	196 ± 17	431 ± 11
Exploration Percentage (%)	87 ± 4	91 ± 3	95 ± 3	90 ± 5	93 ± 2	97 ± 2	93 ± 4	94 ± 5	98 ± 2
Overlap Percentage (%)	31 ± 5	46 ± 6	38 ± 2	37 ± 8	31 ± 9	31 ± 6	47 ± 7	39 ± 8	32 ± 6
MAP-SSIM	0.73 ± 0.12	0.89 ± 0.08	0.92 ± 0.06	0.68 ± 0.21	0.71 ± 0.13	0.89 ± 0.08	0.71 ± 0.17	0.73 ± 0.15	0.92 ± 0.10
CPU Utilization (%)	112 ± 22	79 ± 18	42 ± 8	97 ± 18	65 ± 15	34 ± 9	68 ± 21	47 ± 16	36 ± 9
RAM (MB)	824 ± 19	1264 ± 41	665 ± 24	624 ± 16	819 ± 33	432 ± 21	452 ± 19	724 ± 38	399 ± 18
COM Payload (MB)	2.2 ± 0.08	2.4 ± 0.06	0.6 ± 0.02	1.3 ± 0.06	1.8 ± 0.04	0.4 ± 0.01	1.1 ± 0.04	1.3 ± 0.05	0.2 ± 0.01
Time Complexity Per Robot	$O(n \times \log(n))$	$O(n)$	$k \cdot \text{width of sub-map}$ $l \cdot \text{height of sub-map}$	$O(n \times \log(n))$	$O(n)$	$k \cdot \text{width of sub-map}$ $l \cdot \text{height of sub-map}$	$O(n \times \log(n))$	$O(n)$	$k \cdot \text{width of sub-map}$ $l \cdot \text{height of sub-map}$

Simulation Setup: A closed simulation environment based on the ROS Gazebo simulator with two indoor template environments is used: the Gazebo’s house world ($\approx 250m^2$ area) and the Amazon AWS bookstore world ($\approx 100m^2$ area). The robots may quickly finish the map exploration in a closed environment. Each robot has a laser scanner to gather data about its surroundings. The robot’s trajectory is determined based on the fusion of wheel odometry and laser scan information. The following parameters are used in the experiments in the simulated environment. The laser scanner’s range and $r_{i,s}$ are set to $15m$ and $1m$, respectively. Additionally, the robot’s maximum linear and angular speeds in the simulation are set to $0.5ms^{-1}$ and $\frac{\pi}{4}rads^{-1}$, respectively. The global detector’s growth factor η and the local detector’s growth factor η_1 in the RRT detector are set to $5m$ and $3m$, respectively. The weight parameters, $\alpha = 0.6$, $\gamma = 0.95$ and $\lambda_i = 2$ for $1m$ distant step. Each experiment was run for ten trials, with average observations reported. We evaluate the performance in the following three scenarios to validate the robustness and scalability of the proposed solution: 1) **3 robots** in the house world, 2) **3 robots** in the bookstore world, and 3) **6 robots** in the bookstore world.

8.4.1 Evaluation Metrics

The proposed CQLite and the methods put forward by RRT (L. Zhang et al., 2020) and DRL (Hu et al., 2020) are compared in our experiments. We use the below metrics for a comprehensive evaluation:

1. **Mapping Time:** The amount of time spent mapping is a gauge of how effectively a multi-robot map is explored;
2. **Path Length:** This term refers to the length of the robot’s trajectory as a whole, which is necessary for a multi-robot system to explore the entire map. The entire trajectory length gives an idea of the robot’s energy usage while subtly describing its investigation’s effectiveness;
3. **Exploration Percentage:** The percentage of the generated map with time elapsed;

4. **Overlap Percentage:** The percentage of the overlap of the explored map with time elapsed;
5. **Map SSIM:** Structural similarity index measure of generated maps compared with ground truth map to measure map correctness;
6. **CPU Utilization:** The maximum percentage consumption of the processor of a robot throughout the trajectory;
7. **Memory Consumption (RAM):** The maximum occupied memory by the robot throughout the trajectory;
8. **COM payload:** The size of the data communicated by a robot averaged over iterations.
9. **Time Complexity:** Empirical notation of algorithmic computational complexity.

8.4.2 Results and Discussion

We have reported each approach’s average performance after ten trials in each condition to reduce the measurement noise and analyze the statistical details. A sample of the mapping outcomes of the compared approaches with the trajectories followed by three robots in the simulated environment is shown in Fig. 8.3 and generated maps also delineate the map correctness. The outcome should be stated considering the average mapping time, distance traveled, and mapping efficiency. Mapping efficiency is determined by comparing with the original map, and reported percentages are normalized with gazebo world dimensions.

Table 8.1 provides a comparative analysis of different methods on all the performance metrics and the statistical data from the results. It also lists the theoretical (algorithmic) computational complexity. Figs. 8.4 shows the comparison of the approaches in the three key performance metrics: computation, communication, and exploration.

The proposed CQLite reliably outperforms other strategies on the key performance metrics. CQLite covers a larger area in less time, improving mapping efficiency by 10% while traveling 22 fewer meters than RRT in the experiment. In three-robot scenarios, CQLite was more effective than DRL and RRT, with 9% and 8% shorter mapping times, respectively. Its path length was also less than DRL’s by about 38%. The advantages became even more apparent when the trial involved six robots. While the mapping time was around 26% faster than DRL and 7% faster than RRT, the path length was about 38% shorter than with DRL.

CQLite had an exploration percentage that was 4% greater than DRL in the three-robot scenario. This advantage persisted in the six-robot case, where CQLite’s exploration percentage was almost 4% higher than DRL’s while maintaining the lowest overlap percentage. The stability and effectiveness of CQLite in multi-robot exploration tasks are highlighted by these results from various experiments.

Communication-wise, CQLite’s strategy is more effective. Contrary to RRT and DRL, which exchange locally explored maps continually. CQLite showed a significant reduction of more than 80% in the communication payload (average data size) shared between the robots. Notably, CQLite continues to explore at a constant rate even after reaching 60% coverage, in contrast to RRT, which slows down. This dominance carries over into a real-life three-robot bookshop scenario, where it outperformed DRL and RRT regarding reduced mapping time and shorter journey distances. Results have validated the practicality of CQLite by surpassing DRL and RRT in terms of most of the performance matrices in all scenarios. Further, demonstrating its efficacy and applicability on resource-constrained robots, CQLite maintained decreased RAM, CPU, and communication payload usage. CQLite demonstrates its power in managing a range of multi-robot exploration scenarios by offering improved map quality, as higher MAP SSIM ratings indicate. It is particularly appropriate in situations when there are significant communication and resource constraints.

The proposed CQLite can cover the largest possible area in less time than RRT and is comparable to DRL. Specifically, CQLite can achieve 10% higher mapping efficiency by traveling 22 m less than RRT. It is clear that CQLite consumes approximately half of the memory and has less than half the communication payload size as DQL. However, the memory consumption of CQLite is still comparable with RRT, as the RRT approach applies pruning techniques to reduce memory consumption. We also compared the communication and computation overhead of CQLite with RRT and DRL throughout the exploration. CQLite has a few spikes of high CPU utilization for map merging when other maps are received ad-hoc in the contract; RRT has shown continuous high CPU utilization as it keeps applying map merging and randomly exploring trees for exploration. Similarly, DRL keeps optimizing policy and merging maps during exploration.

Regarding communication, RRT and DRL keep sharing locally explored maps. However, CQLite only shared a map when requested and depicted a few peaks in the communication cost plot of Fig. 8.4. Furthermore, our result also demonstrates the exploration percentage of CQLite, which is comparable with DRL and significantly better than RRT. CQLite achieved 8% and 4%

better exploration percentages than RRT and DRL, respectively. After 60% exploration, RRT becomes slow; however, DRL and CQLite explore at the same rate, ending at 91% and 95% exploration for DRL and CQLite, respectively. Overall, the proposed CQLite shows significant potential to outperform the state-of-the-art techniques and creates promising avenues to research further. It applies well to resource-limited and communication-limited applications.

The proposed CQLite algorithm consistently beats the RRT and DRL techniques in various evaluation parameters in the scenario of three robots in a bookshop world. CQLite has speedier exploration capabilities, as evidenced by its mapping time being roughly 9% and 8% less than DRL and RRT, respectively. Additionally, CQLite travels a path length about 38% shorter than DRL, indicating higher robot movement efficiency. While the overlap percentage achieved by CQLite is about 40% less than RRT, the exploration percentage is about 4% higher than DRL, indicating a more thorough exploration with fewer overlapped areas. Additionally, CQLite shows superior performance in MAP SSIM, with a 25% improvement over RRT, and exhibits better efficiency in computational resources, with approximately 48% and 33% less CPU and RAM utilization, respectively, compared to DRL. Compared to DRL, the communication payload is decreased by around 78%.

CQLite continues outperforming the other two approaches while evaluating the performance of six robots in a bookshop environment. The reduced mapping time, which is roughly 26% faster than DRL and 7% faster than RRT, demonstrates the scalability of the suggested solution in a multi-robot scenario. The path length traveled is also reduced, about 38% less than DRL, which points to an improved exploration strategy. With a decrease of about 46% when compared to RRT, CQLite's exploration percentage is around 4% higher than DRL's while keeping the lowest overlap percentage. Better map quality is indicated by CQLite's higher MAP SSIM score, which is an improvement of about 27% over RRT. With reductions of about 45%, 56%, and 85%, respectively, compared to DRL, the CPU utilization, RAM usage, and communication payload remain the lowest among the examined approaches, further supporting the effectiveness and scalability of CQLite in the context of multi-robot exploration.

A strong proof of CQLite's robustness, efficiency, and applicability comes from the performance results of CQLite in the three and six-robot bookstore world scenarios. CQLite is resilient in managing a variety of multi-robot exploration scenarios, as evidenced by its ability to surpass RRT and DRL in terms of mapping time regularly, path length traveled, exploration percentage, and overlap percentage. The effectiveness of CQLite is further demonstrated by the large reductions in CPU use, RAM consumption, and communication pay-

load compared to competing approaches, making it a more computationally and communication-friendly solution for practical applications. CQLite’s capacity to scale to handle scenarios with three and six robots demonstrates its usefulness for tackling exploration jobs with various team sizes.

The experimental results demonstrate the generalizability and practicality of CQLite for real-world applications. While displaying a performance similar to DRL, CQLite surpasses RRT regarding mapping effectiveness, area coverage, and journey time. In addition, CQLite uses less memory and has a smaller communication payload than DQL. Although CQLite uses the same amount of RAM as RRT, it is more efficient overall because of lower communication and processing overhead. Further supporting its efficacy, CQLite’s exploration percentage is noticeably higher than RRT’s and marginally superior to DRL’s. These experimental findings and comparisons of the three main performance indicators—computation, communication, and exploration—justify the CQLite technique for practical applications, particularly when communication and resource availability are constrained. The positive results of CQLite present fresh directions for further study.

One of the limitations of the proposed CQLite is that it relies on wireless communication, which can be intermittent or harsh in specific real-world situations. In such scenarios, a communication-aware strategy can be integrated with our approach to tolerate changes in communication channels.

8.5 Summary

This dissertation proposes a novel, distributed Q-learning-based approach coined as "CQLite" for cooperative multi-robot exploration. The conventional exploration methods necessitate robots to maintain an internal global map of the environment, subsequently using localization and planning techniques for navigation. However, these methods overlook the high communication and update costs associated with map merging. CQLite employs a coverage-weighted reward function to address these challenges and minimizes communication overhead for quicker convergence and complete environment coverage. The robots only share the updated Q-value at newly explored frontiers via wireless communication and perform map merging ad-hoc, thereby reducing communication costs.

Theoretical analysis is performed to understand the convergence and efficiency of CQLite. The method is validated through simulated indoor map explorations using multiple robots. The results indicated that CQLite significantly outperformed state-of-the-art multi-robot exploration techniques such

as Rapidly-exploring Random Trees (RRT) and Deep Reinforcement Learning (DRL) regarding mapping performance with more than a twofold reduction in computation and communication. However, the limitation of CQLite lies in its reliance on wireless communication, which can be intermittent or harsh in specific real-world scenarios. Therefore, future work would explore integrating a communication-aware strategy with communication-efficient CQLite to mitigate changes in communication channels.

The chapter concludes that CQLite provides efficient exploration, quick convergence, and minimal computational cost. It is a viable solution for practical applications, particularly in situations with constrained communication and resource availability. Future work will also investigate the general applicability of the reward function for heterogeneous teams of multi-robot systems. This chapter sets the ground for using reinforcement learning for efficient map exploration under global localization; however, the ultimate objective of this dissertation is to provide a standalone solution for exploration and localization, which is discussed in the next chapter. Stay tuned and ready for the surprise!

CHAPTER 9

SEAL: SIMULTANEOUS EXPLORATION AND LOCALIZATION IN MULTI-ROBOT SYSTEMS

9.1 Introduction

This chapter discusses the integration strategy for simultaneous exploration and localization, which is the last algorithmic solution of this dissertation. The growing use of autonomous robots in various applications, including environmental monitoring, search and rescue, and mapping, has made multi-robot robotic exploration a major study field in recent years. In these scenarios, the robot must explore uncharted territory while concurrently maintaining other criteria like map accuracy, cost of travel, journey time, and energy savings. Information-based exploration methods have drawn particular attention because of their capacity to facilitate speedier exploration and efficiently expand to 3D scenes. Numerous studies have examined various methods for achieving these objectives. To build a reward function and choose the best control strategies that reduce the uncertainty for 2D and 3D maps, these techniques employ information-theoretic metrics like mutual information (MI) (Y. Xu et al., 2021).

Entropy (Botteghi et al., 2020), mutual information (Jadidi et al., 2015), and Bayesian optimization (Bai et al., 2016) based multi-robotic exploration techniques are only a few of the many information-based exploration techniques proposed in the literature. To reduce map uncertainty when exploring new areas, these techniques improve the exploration process based on several information-theoretic measures. Specifically, reducing pose uncertainty can

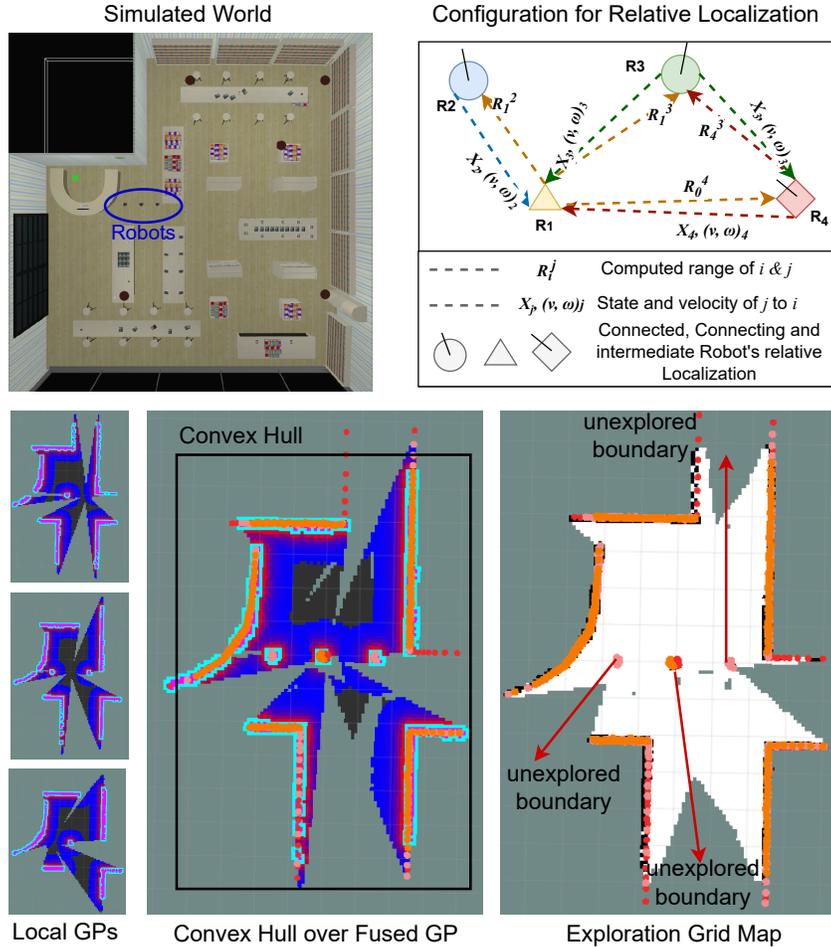


Figure 9.1: Overview of the proposed simultaneous exploration and localization in the multi-robot system, shown with the ROS Gazebo simulation, configuration for relative localization same as DGORL (Latif & Parasuraman, 2022a), and the exploration process with the convex hull

significantly reduce map uncertainty, and an information-based controller taking pose uncertainty reduction into account can direct the robot to a location where it is more likely to find a loop closure, significantly improving the overall trajectory accuracy and the resulting map (Valencia et al., 2018).

Accurate positional data is required for multi-robot exploration, which is one of the challenges posed in GPS-denied environments. Conventionally, the robot employs sensors to measure the surroundings and updates its position according to the sensors' data. This strategy can fail if the sensor is faulty or does not provide enough information. In these circumstances, simultaneous localization and mapping (SLAM) methods can be applied (Schmuck & Chli,

2019), where the robot estimates its position in the environment and simultaneously creates a map of the surrounding area. While SLAM approaches in the literature can handle dynamic scenes very well, they might not work well in contexts with few map features or without prior global position information. Furthermore, SLAM requires feature-based map merging, which is a computationally expensive procedure and causes failure while expansion of the global map. However, we aim to overcome this limitation by only applying efficient GP fusion in parallel to the relative localization, which makes the overall system computationally efficient and provides accurate positioning and mapping information.

This paper proposes a SEAL approach incorporating information-based exploration techniques to enhance the efficiency and accuracy of robotic localization and exploration designed for multi-robot systems. To choose the next-best viewpoint for exploration, we use linearized convex hull optimization, which ensures maximum coverage of the unknown region. Our proposed method uses a Gaussian process (GP) modeling based on the fusion of multiple GPs received from connected robots and relative position estimation, further improving the posterior belief of exploration and localization. Furthermore, our method is appropriate for dynamic settings with limited features or when there is no prior knowledge and does not rely on external sensors. Fig. 9.1 provides an overview of the proposed method implemented in the Robot Operating System (ROS) framework.

The main contributions of SEAL are as follows:

- We propose a new distributed exploration strategy for multi-robot systems using Gaussian process (GP) modeling over relative localization with inter-robot communication data. Here, we integrate the Rao - Blackwellization technique to improve the posterior beliefs of both exploration and localization.
- We focus on maximizing the coverage (exploration) of an unknown environment with multiple robots using the linearized convex hull optimization method.
- We extensively evaluate our SEAL approach's exploration performance in realistic ROS Gazebo simulations compared against two state-of-the-art exploration approaches: Rapidly exploring Random Tree (RRT)-based exploration (L. Zhang et al., 2020) and Deep Reinforcement Learning (DRL)-based exploration (Hu et al., 2020).
- To further analyze the performance of localization, we compare with typical Gmapping-based SLAM (used in RRT Exploration (L. Zhang et al.,

2020)) and a relative localization method, DGORL (Latif & Parasuraman, 2022a), which is based on our prior work.

¹⁶ <https://github.com/herolab-uga/ROS-SEAL>

- We open source¹⁶ our method as a ROS package for use and further development by the robotics community.

The SEAL’s motivation is utilizing data sharing through inter-robot communication to simultaneously localize and explore the environment. It continually improves the beliefs of exploration and localization using Rao-Blackwellization for environments where global position information is unknown. Furthermore, ensuring maximum exploration inspired by convex hull optimization (Pham et al., 2009) and its linearization addresses the problem of coverage in unknown and constrained environments. Our method does not rely on global information for localization and boundary information for exploration. SEAL achieved high localization, exploration accuracy, and efficiency by employing parallel computation of localization and exploration powered by relative localization (Latif & Parasuraman, 2022a), Rao-Blackwellization (Rao, 2008), and linearized convex hull optimization to ensure maximum area coverage.

9.2 Related Work

Occupancy grid maps are extensively used in 2D and 3D settings because they can be swiftly queried and updated (Hornung et al., 2013). Additionally, there are several applications where the data in OGMs can be used for navigation and collision-free path planning. By utilizing these benefits, OGMs offer a new way to quantify map uncertainty in light of newly observed data (Z. Zhang et al., 2020). Collaborative simultaneous localization and mapping (CSLAM), which is possible with a centralized system, is an alternative strategy to maximize the effectiveness of exploration. Shi et al. (Y. Shi et al., 2020) presented an adaptive informative sampling strategy that divides the environment into various sections. This strategy, however, necessitates a substantial amount of communication and computational resources. Placed and Castellanos (Placed & Castellanos, 2021) suggested a quick autonomous robotic exploration strategy using an underlying graph structure. Their approach makes it possible to explore new settings quickly and effectively. All of these approaches use OGMs as a source of mapping and exploration.

Frontier detection algorithms are built for fast navigation using rapidly expanding random trees known as RRTs. For instance, to improve the effectiveness and efficiency of multi-robot map exploration, the aim is integrated into an optimization framework that uses RRTs in (L. Zhang et al., 2020). However,

this strategy’s limitations are the cost of computing the optimization algorithms and the potential for non-optimal results due to the stochastic nature of RRTs.

OGMs are not the only mapping method; continuous occupancy mapping is one (COM). For instance, given the observed observations (training set) comprising free and occupied space, Gaussian processes-based occupancy maps (Swiler et al., 2020) employ kernel inference methods to learn distributions across continuous occupancy. GP maps may query map points at any resolution and record the spatial correlations between sample points. They are particularly useful for extrapolating the occupancy and uncertainty of unseen places.

Jadidi et al. (Jadidi et al., 2014) presented a GP-based mapping and exploration technique (GPME) to identify frontiers defined by mean occupancy from the GP map to realize COM-based information-theoretic exploration. In the extended work by Jadidi et al., (Ghaffari Jadidi et al., 2018), a high-dimensional logistic regression classifier used to produce a probabilistic frontier representation based on the uncertainty implied by GP is also noteworthy. The forward sensor model (FSM) and combined predictive distribution over the entire map are used to numerically design and compute the GPME surfaces in the present perception field.

Another method that is gaining popularity in solving exploration and mapping difficulties is reinforcement learning (RL). A deep RL technique for autonomous graph exploration with fallible sensing capabilities was presented by Chen et al. (F. Chen et al., 2020). Their technique performed well but can only be used in relatively narrow surroundings. Chen et al. (F. Chen et al., 2021) also proposed a zero-shot RL technique on graphs for autonomous exploration in uncertain scenarios. Although their method drastically reduced training time, it hasn’t been tested in real-world situations environmental changes may hamper robotic exploration in outdoor settings. Moreover, a Voronoi-based strategy that uses DRL for cooperative multi-robot exploration is suggested in (Hu et al., 2020). The study aims to improve exploration efficiency by dividing the environment into Voronoi cells and assigning a robot to each cell for study. However, this approach is constrained by the difficulty of training deep RL models and the potential for less-than-ideal results because of the complexity of the environment and the unknown mapping of the environment.

This approach is compared to frontier detection-based RRT (L. Zhang et al., 2020) and DRL-based (Hu et al., 2020) exploration strategies. Both approaches are assumed to have accurate position information to be fully functional; robots merge locally examined maps and share them between iterations, increasing computing complexity. Our proposed SEAL overcomes limitations

of existing solutions of explorations that require accurate global positioning information and works on feature matching map merging techniques by providing an optimal solution.

9.3 Problem Formulation

Problem Statement: Given that n robots operating in an uncharted region are wirelessly connected, using shared exploration grids and relative location information, robots should efficiently and accurately perform simultaneous exploration and localization.

Notations: Let r_i be the concerning robot, n be the total number of robots deployed in the unknown environment, and X_i be the robot's position. We only consider x, y coordinates of robots from the frame of reference of r_i , which is sufficient to update the exploration grid. Let $z_{i,j}$ be the scanned observation between robots i and j and g_i be the exploration grid for robot i . Additionally, w_i is the robot's weight at i^{th} exploration grid g_i , s_j^i is the observed Radio Signal Strength Indicator (RSSI) for connected robot j , and C_i is the convex hull of the robot's postures.

The difficulty lies in figuring out the robots' positions while minimizing the sensing gap between them and building an accurate grid map of the bounding box. The problem is best described as follows:

$$\min_{X_{1:n}} \sum_{i=1}^n \sum_{j \in \mathcal{N}_i} \text{dist}(X_i, X_j) \quad (9.1)$$

Each robot r_i forms a 2-D Pose Estimation Graph (PEG) $G_i = (V_i, E_i)$ using the Received Signal Strength Indicator (RSSI) from connected robots. \tilde{X}_i is the position estimation of the objective function:

$$\min_{\tilde{X}_{1:n}} \sum_{i=1}^n \sum_{j \in \mathcal{N}_i} (\tilde{X}_i - X_j - z_{i,j})^2 \quad (9.2)$$

Robot r_i forms an exploration grid considering uniform distributions over the workspace and receives observations $z_{i,j}$ from the connected robots at each position and updates its global grid as $g_{global} = \sum_{i=1}^n w_i g_i$. Further, \tilde{X}_i is updated using Rao-Blackwellization (Robert & Roberts, 2021) for the updated belief state:

$$p(\tilde{X}_i, g_i | z_{i,j}) = p(\tilde{X}_i | z_{i,j}) p(g_i | \tilde{X}_i) \quad (9.3)$$

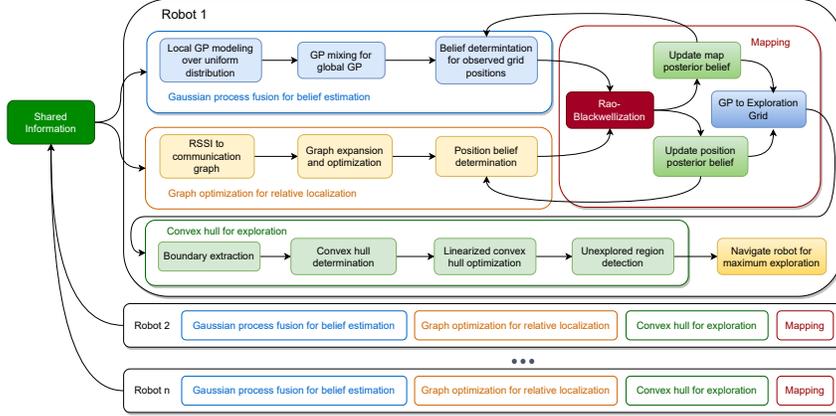


Figure 9.2: System architecture of SEAL distributed across several robots. It shows Robot 1's process of exploring, convex hull optimization, and relative localization operations using shared information from n connected robots.

Lastly, the convex hull C_i is updated using linearized convex hull optimization for the following objective :

$$\min_{x_{1:n}} \sum_{i=1}^n \sum_{j \in \mathcal{N}_i} \text{dist}(\tilde{X}_i, \tilde{X}_j) \quad (9.4)$$

As the above convex hull objective function is non-linear, it can be linearized as:

$$\text{minimize } f(\tilde{X}_i) \text{ subject to } \tilde{X}_i \in X, \quad (9.5)$$

Where $f(\tilde{X}_i)$ is the global objective function, \tilde{X}_i is the robots' positions in the region of low wireless vicinity, and X is the set of feasible positions.

$$f(\tilde{X}_i) \approx f(\tilde{X}_i) + \nabla f(\tilde{X}_i)^T (\tilde{X} - \tilde{X}_i) \quad (9.6)$$

$\nabla f(\tilde{X}_i)$ is the gradient of the objective function evaluated at \tilde{X}_i .

To maximize the coverage, the linearized objective function can be minimized to estimate the following position to navigate for r_i . Robots then position themselves to cover the closed region in the bounded area efficiently. Standard barrier certificates and navigation stacks are utilized for safe autonomous navigation. The system architecture of SEAL for the mentioned problem can be visualized in Fig. 9.2.

Algorithm 12: Distributed SEAL

```
1 Required: previous position  $X_{t-1}$ , local GP  $g$ , received GPs, RSSI
   from connected robots  $S$ , number of connected robots  $n$ ;
2 do in parallel
3    $g$  = Local predicted GP grid;
4   for  $g_j \in GPs$  from connected robots do
5     Merge received  $g_j$  into local  $g$ ;
6      $b_j^g$  = update exploration belief of  $g$  as Eq. 9.8;
7   end
8   for  $j \in connected\ robots$  do
9     Apply relative localization using DGORL (Latif &
       Parasuraman, 2022a);
10     $bel_j(\tilde{X})$  = update position belief;
11  end
12 end
13 Apply Rao-Blackwellization to update exploration as well as position
   belief;
14 Update position belief using GP belief as Eq. 9.12;
15 Update GP belief by incorporating Entropy  $H(bel())$  from Eq. 9.14;
16 Build occupancy grid map based on updated belief from GP;
17  $C$  = received linearized convex hull using Alg. 13;
18 Navigate the robot to the unexplored region based on  $C$  for max
   exploration.
```

9.4 Approach and Algorithms

SEAL is composed of four major parts:

1. Global Exploration over Distributed Gaussian Process (GP) Modeling (W. Luo & Sycara, 2018).
2. Probabilistic Extension of Relative localization using DGORL (Latif & Parasuraman, 2022a).
3. Integration of relative localization with exploration to update belief state using modified Rao-Blackwellization.
4. Multi-robot distribution for full coverage using linearized convex hull optimization.

The method proposed in this work leverages the strengths of non-parametric generalization of GP for individual robot exploration, polynomial time graph optimization for relative localization, probabilistic efficiency of Rao-Blackwellization to update belief state, and linearized convex hull optimization for complete coverage without performing Voronoi partitioning. Our proposed SEAL is a merger of relative location with exploration, along with the linearized extension of the convex hull algorithm that can leverage multi-robot teams to cover the bounded region efficiently and comprehensively.

Furthermore, SEAL relies only on shared data between robots for position estimation and exploration, without considering any centralized information sources. Alg. 12 elaborates the distributed approach of SEA utilizing GPs and relative localization belief information to generate a map and determine the unexplored region for robot to navigate to using linearized convex hull optimization.

9.4.1 Gaussian Process for Exploration

GP regression is a method that is frequently used to model spatial phenomena. Simulating the hidden mapping from training data to the target phenomenon is possible while considering the uncertainty provided by the natural non-parametric generalization of linear regression. Assume that the target phenomenon, a wireless source, satisfies a multivariate joint Gaussian distribution in this case. The Gaussian probability distribution of the phenomena $\omega(g)$ as given by the mean function $\mu(g) = \mathcal{E}[\omega(g)]$ and covariance function $\delta^2(g, g') = \mathcal{E}[(\omega(g) - \mu(g))^T \times (\omega(g') - \mu(g'))]$ is the output of the trained GP model using training data.

Formally, let $GP^{[i]} = [g_1^{[i]}, \dots, g_k^{[i]}]^T$ be the grid positions at which r_i observed k noisy RSSI readings from the source $S^{[i]} = [s_1^{[i]}, \dots, s_k^{[i]}]^T$. Since the mean function is assumed to be zero without losing generality, each observation is noisy, with the formula $y = \omega(g) + \varepsilon$. To achieve this, given a sampling location $g_k \in GP$, we have the conditional posterior means $\mu_{g_k|GP_j, y_j}$ and variance $\delta_{g_k|GP_j, y_j}^2$ as follows from the learned GP model defining the Gaussian distribution of $\omega(g_k) \sim N(\mu_{g_k|GP_j, y_j}, \delta_{g_k|GP_j, y_j}^2)$ for sample position k .

Once each robot learns its GP model for its current position, they share GPs with the connected robots for global exploration. Since each local GP is assumed to be learned by each robot, we assume that the underlying data distribution to be explored can be characterized by a combination of n fused GPs, $GP = \{GP_1, \dots, GP_n\}$ received from connected robots. We define $p(g | s_{1:n}, X_{1:n}) = b_n^g$ as the likelihood that, for each point in the environment, $g \in$

GP is best characterized by the GP that the r_i has learned. Then, we produce a combination of GP models weighted by the b_n^s at any grid $g \in GP$ as a linear combination of GP_1, \dots, GP_n . The means $\mu'_{g|GP,Y}$ and variance $\delta_{g|GP,Y}^2$ for robot i are the parameters that define the model as:

$$\begin{aligned}\mu'_{g|GP,Y} &= \sum_{j=1}^n b_j^{s[i]} \times \mu_{g|g_j,y_j}^{[i]} \\ \delta_{g|GP,Y}^2 &= \sum_{j=1}^n b_j^{s[i]} \times \left(\delta_{g|g_j,y_j}^2 + \left(\mu_{g|g_j,y_j}^{[i]} - \mu'_{g|GP,Y} \right)^2 \right)\end{aligned}\quad (9.7)$$

In this problem, the algorithm computes the weight distribution in the expectation step (E-Step). Then, the maximization step updates the parameters of the local fused GP with the estimated weight distribution (M-Step), the same as discussed in GPMix (Y. Shi et al., 2020). Finally, the weight distribution is set to the following values before the first iteration for each random wireless sample s at grid position g :

$$O_{bg} \approx \begin{cases} 1 & \text{if } g \in GP^{[i]}, b^s \geq \theta \\ 0 & \text{Otherwise} \end{cases} \quad \forall j = 1, \dots, n \quad (9.8)$$

Where θ is the threshold for exploration and O_{bg} is the value of grid position in the exploration grid map.

9.4.2 Relative Localization

Assume at a given time t , a team of robots considered as vertices of the graph contains $n \in \mathbb{N}$ connected robots can form a weighted undirected graph, denoted by $G = (V, E, A)$, of order n consists of a vertex set $V = \{v_1, \dots, v_n\}$, an undirected edge set $E \in V \times V$ is a range between connected robots and an adjacency matrix $A = \{a_{ij}\}_{n \times n}$ with non-negative element $a_{ij} > 0$, if $(v_i, v_j) \in E$ and $a_{ij} = 0$ otherwise. An undirected edge e_{ij} in the graph G is denoted by the unordered pair of robots (v_i, v_j) , which means that robots v_i and v_j can exchange information with each other.

Here, we only consider the undirected graphs, indicating that the robots' communications are all bidirectional. Then, the connection weight between robots v_i and v_j in graph G satisfies $a_{ij} = a_{ji} > 0$ if they are connected; otherwise, $a_{ij} = a_{ji} = 0$. Without loss of generality, it is noted that $a_{ii} = 0$ indicates no self-connection in the graph. The degree of robot v_i is defined by $d(v_i) = \sum_{j=1}^n a_{ij}$ where $j \neq i$ and $i = 1, 2, \dots, n$. The Laplacian matrix of the graph G is defined as $L_n = D - A$, where D is the diagonal with $D =$

$diag\{d(v_1), d(v_2), \dots, d(v_n)\}$. If the graph G is undirected, L_n is symmetric and positive semi-definite. A path between robots v_i and v_j in a graph G is a sequence of edges $\{(v_i, v_{i1}), (v_{i1}, v_{i2}), \dots, (v_{in-1}, v_{in}), (v_{in}, v_j)\}$ in the graph with distinct robots $v_{in} \in V$. An undirected graph G is connected if a path exists between any pair of distinct robots v_i and v_j where $(i, j = 1, \dots, n)$.

We begin by creating the Estimated Relative Position Measurement Graph (ERPMG): $G_E = (V_E, E_E, A_E)$, based on the Relative Position Estimation Graph (RPMG): $G = (V, E, A)$, which can be constructed using range information received from connected robots of an n -robot system and describes the relative position measurements among robots. Based on robotic motion constraints, we extend RPMG to accommodate all possible robot positions in the succeeding time step.

Since each robot also gets RSSI from other robots, we may map the predicted relative positions for each robot using the model and identify the k soft maximum out of them by locating the intersection region as an area of interest. Once we have k possible positions of each robot, we can generate $< n^k$ solvable graphs for optimization.

We consider a network with n robots for optimization of expanded graphs, labeled by $V = \{1, 2, \dots, n\}$ and k possible connections to other robots. Every robot i has a local convex objective function and a global constraint set. The network cost function is given by:

$$\begin{aligned} \text{minimize } f(\mathbf{x}) &= \sum_{i=1}^N f_i(\mathbf{x}) \\ \text{subject to } \mathbf{x} \in \mathcal{D} &= \left\{ \mathbf{x} \in \mathbf{R}^k : c(\mathbf{x}) \leq 0 \right\} \end{aligned}$$

Here, $\mathbf{x} \in \mathbf{R}^k$ is a global decision vector; $f_i : \mathbf{R}^k \rightarrow \mathbf{R}$ is the convex objective function of robot i known only by robot i ; D is a bounded convex domain, which is (without loss of generality) characterized by an inequality constraint, i.e., $c(\mathbf{x}) \leq 0$, where $c : \mathbf{R}^k \rightarrow \mathbf{R}$ is a convex constraint function known by all robots. In addition, to estimate the relative positioning of connected robots, we also calculate the probability of estimation $bel_j(\tilde{X}) = p(\tilde{X}_j | \tilde{X}_i, s_{i,j})$ for each relative position, where \tilde{X} corresponds to the position estimation by \mathbf{x} , which will be used in the improvisation of the exploration grid. Technical and theoretical details of the proposed relative localization technique are explained in our previous work DGORL (Latif & Parasuraman, 2022a).

9.4.3 Rao-Blackwellization for SEAL

In the Rao-Blackwellization update procedure, according to (Rao, 2008), the proposal distribution is suboptimal, especially when the proprioceptive odometer measurements are less accurate than the position estimates through the relative location as mentioned in Section 9.4.2. Instead, we can use the persistent exploration grid map belief in Gaussian Process as discussed in Section 9.4.1 to improve the localization accuracy and vice versa. Furthermore, in a Bayesian framework, we explicitly incorporate the exploration grid map belief into the measurement likelihood function:

$$\begin{aligned}
 & bel_j^{[i]}(\tilde{X}) \\
 &= \psi \overline{bel}_j^{[i]}(\tilde{X}) \int_g p(s_j|g, \tilde{X}_{1:j}, s_{1:j-1})^{[i]} p(g|\tilde{X}_{1:j-1}, s_{1:j-1})^{[i]} dg \quad (9.9) \\
 &= \psi \overline{bel}_j^{[i]}(\tilde{X}) \int_g p(s_j|g, \tilde{X}_{1:j}, s_{1:j-1})^{[i]} b_{j-1}^{g[i]} dg
 \end{aligned}$$

According to the definition of posterior map belief, $b_{j-1}^{g[i]}$ is a sufficient statistic for all previous positions $\tilde{X}_{1:j-1}$ and RSSI $s_{1:j-1}$, and ψ is the belief update constant, the following expression holds:

$$p(s_j|g, \tilde{X}_{1:j}, s_{1:j-1})^{[i]} \approx p(s_j|g, \tilde{X}_j, b_{j-1}^g)^{[i]} \quad (9.10)$$

Similarly, the belief $b_j^{g[i]}$ is also a sufficient statistic for the current estimated position \tilde{X}_j and the previous map belief $b_{j-1}^{g[i]}$, so we can get:

$$p(s_j|g, \tilde{X}_j, b_{j-1}^g)^{[i]} \approx p(s_j|b_j^g)^{[i]} \quad (9.11)$$

Thus Eq. 9.9 can be written as:

$$bel_j^{[i]}(\tilde{X}) \propto \overline{bel}_j^{[i]}(\tilde{X}) \int_g p(s_j|b_j^g)^{[i]} b_{j-1}^{g[i]} dg \quad (9.12)$$

Now the weight of sample j possesses the sample position for robot i can be defined as:

$$w_j^{[i]} = \int_g p(s_j|b_j^g)^{[i]} b_{j-1}^{g[i]} dg. \quad (9.13)$$

Furthermore, for a sample approximating the position belief, a straightforward method to estimate pose uncertainty is to use all normalized weights in a dis-

Algorithm 13: Distributed Linearized Convex Hull Optimization

```
1 C = convex hull ( $O_f \cup O_o$ ) #convex hull of observed
   walls and space;
2  $C_o = O_o \in H$  #identify occupied cells on convex
   hull contour;
3  $L_o = \text{Probabilistic Hough Lines}(C_o)$  #search for lines on
   convex hull;
4 for each  $l$  in  $L_o$  do
5   |  $c_{i,j} \in I_o = \text{find intersections}(l)$  #identify
   |   intersecting hull lines;
6 end
7 C = convex hull( $O_f \cup O_o \cup I_o$ ) #convex hull of
   observations and intersection points;
8 if C is linear then
9   | return C;
10 else
11   |  $L_C = \text{Linearize}(C)$  #Linearize convex hull;
12   | return  $L_C$ ;
13 end
```

cretized way as:

$$\begin{aligned} H(\text{bel}^{[i]}(\tilde{X})) &= - \int \text{bel}^{[i]}(\tilde{X}) \log \text{bel}^{[i]}(\tilde{X}) d\tilde{X} \\ &\approx - \sum_{j=1}^{N_s} \omega_j^{[i]} \log \omega_j^{[i]} \end{aligned} \quad (9.14)$$

9.4.4 Maximum Exploration with Distributed Linearized Convex Hull Optimization

The primary goal of linearizing convex hull optimization is to maximize the coverage of the explorable space and distribute robots to unknown areas.

The method, as shown in Alg. 13, is a heuristic approach that starts by figuring out the convex hull of the observed map. Observing wall cells in the exploration map on the convex hull is used to forecast potential unobserved areas of the area. The probabilistic Hough line transformation, which uses a maximum likelihood estimation of a line via sparsely connected points, is used

to find these unobserved sections of the region. It identifies unseen connections between observed wall segments. The map’s unseen areas are then expanded along the identified wall lines.

Each junction of these wall-line projections adds a projected corner (represented by a single cell, $c_{i,j} \in I_o$) to the exploration map. Limiting the distance of predicted corners from the closest observations is necessary to prevent nearly parallel lines from predicting corner locations that are unreasonably far from the observed space. The observed areas of the map and the corner points are then combined to form a new convex hull. This new convex hull provides an initial estimation of the boundaries of the exploration space. The cells inside the hull are initially anticipated to be free, while the cells along the hull are predicted to be occupied in the exploration map.

In the case of a non-linearize convex hull, projecting robots to specific unexplored regions would be non-optimal. To linearize a non-convex hull, we first need to find a convex hull that approximates the non-convex function. This can be done using techniques such as convex relaxation or piecewise linearization. Once we have a convex hull approximation, we can use the following linearization equation:

$$f(C) \geq \sum_{j=1}^n a_j \nabla f_j(C) + b, \quad (9.15)$$

where $f(C)$ is the non-convex function we want to optimize, $\nabla f_j(C)$ are the convex functions that approximate $f(C)$, a_j are non-negative coefficients, and b is a constant that ensures the approximation is valid.

Upkeep on the projected map is done before moving on to the whole exploration. To account for unseen depth in barriers, occupied cells, $O_o \cup I_o$, are inflated into I_f ; for example, when an obstacle is only visible from one side, it has no observable depth. User-specified inflation depth is determined by prior operating environment knowledge. Then, the map’s unreachable regions inside the expected perimeter are labeled as inferred obstacles.

9.4.5 Time Complexity

We examine the temporal complexity for each observation using Algs. 12 and 13. The map resolution α_M , the map size n , the maximum size of a convex hull C , the number of beams per scan n_z , the number of measurement α_z , and the one of continuous map occupancy α_m for a map of size m is used to define the complexity. We assume that the squared grid’s size is fixed, the query time is

Table 9.1: Performance results of different approaches. * indicates the best performer. L - Localization. E - Exploration.

Parameters	SLAM + RRT (L. Zhang et al., 2020) L + E	DRL (Hu et al., 2020) E Only	DGORL (Latif & Parasuraman, 2022a) L Only	SEAL (ours) L + E
Exploration Metrics				
Mapping Time (s)	302 ± 22	273 ± 33	N/A	247* ± 19
Total distance (m)	267 ± 14	198 ± 21	N/A	139* ± 11
Explored Area (%)	92 ± 2	94 ± 3	N/A	97* ± 2
Localization and Mapping Metrics				
Map SSIM	0.69 ± 0.15	0.72 ± 0.12	N/A	0.88* ± 0.09
ATE (m)	0.5 ± 0.09	N/A	0.09 ± 0.04	0.04* ± 0.02
ALE (m)	1.3 ± 0.14	N/A	0.81 ± 0.5	0.63* ± 0.2
Computing, Communication, and Efficiency Metrics				
CPU (%)	109 ± 20	88 ± 17	N/A	82* ± 11
RAM (MB)	1014 ± 22	1321 ± 59	N/A	891* ± 27
Communication (MB)	2.4 ± 0.07	2.7 ± 0.09	N/A	1.2* ± 0.04

constant, and the data structure containing the OG map is identical to the GP data structure.

The proposed SEAL has two parallel processes: Gaussian process fusion for exploration belief estimation and Graph optimization for belief estimation of relative localization. Rao-Blackwellization will use both beliefs to update overall localization and map beliefs. The resulting time complexity of two parallel processes are $O(n^2)$ and $O(\alpha_M^{-2} n_z \alpha_z^{-1} \alpha_m^{-1})$. It is worth noting that the time complexities of standard Log-Odds for generating OG mapping from GP are $O(n_z \log n)$. Note that $n_z < n$ normally because of the limited sensing range and narrow beam angle. The time complexity of the proposed SEAL is, at worst, quadratic about the cone size and map resolution for belief update and GP computation, respectively. Importantly, the time cost of map belief update is quite similar to Log-Odds approach in a large scene because the independence of map size n and the actual cone size is much smaller in cluttered environments. The time efficiency of SEAL calculation also depends on the linearization of convex hull optimization, which is $O(n \log n)$. Hence, the overall upper bound on the time complexity per robot for the proposed SEAL will be $O(n \times m)$, where m is the number of immediate (connected) neighbor robots.

This time complexity is significantly better than other methods, such as SLAM + RRT (L. Zhang et al., 2020) and DRL (Hu et al., 2020), which have complexities $O(n \times \log(n))$ and $O(n \times k)$, respectively, where k is the number of temporal states, and $m \ll k$.

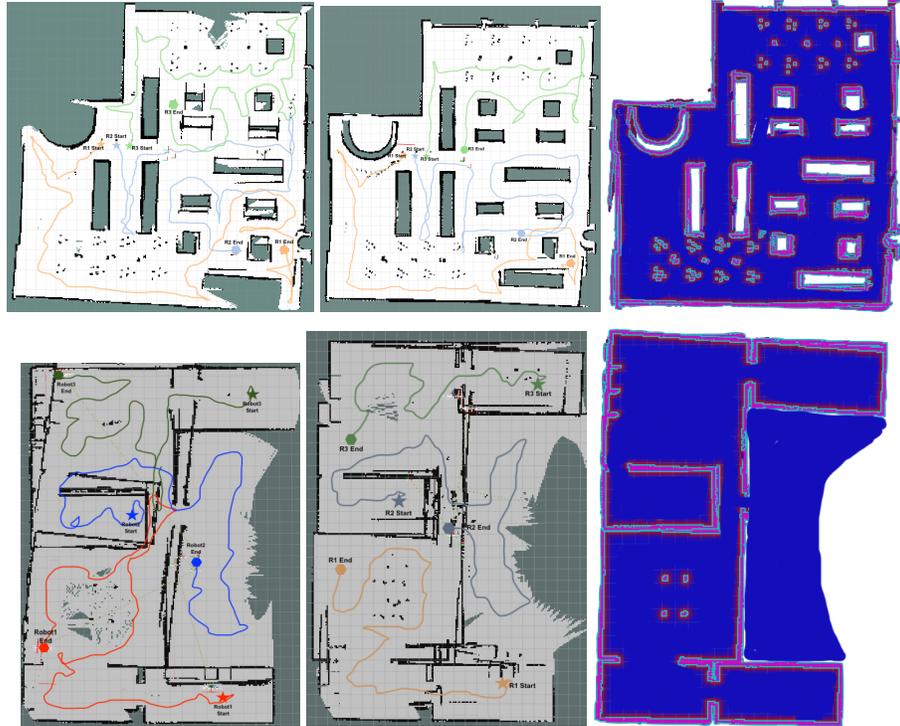


Figure 9.3: Exploration grid maps along trajectories: The top pictures show the Bookstore world and the bottom pictures show the House world. **(Top Left)** Explored map using RRT (L. Zhang et al., 2020), **(Top Center)** Explored map using proposed SEAL, **(Top Right)** Explored maps' confidence (belief map) of SEAL. **(Bottom Left)** Explored map using DRL (Hu et al., 2020), **(Bottom Center)** Explored map using proposed SEAL, **(Bottom Right)** Explored maps' confidence (belief map) of SEAL. *Confidence maps are provided only by SEAL.*

9.5 Experimental Validation

We evaluated the SEAL approach in two situations in ROS Gazebo-modified versions of the AWS bookstore and house worlds. A laser sensor with a 180° field of vision, a 5 m range, and 1500 beams per scan are installed on the robot's wheeled platform. The robot's maximum speeds in both directions are $0.2 \frac{m}{s}$ and $0.8 \frac{rad}{s}$, respectively. Each robot in the experimentation performs SEAL and generates a map based on updated belief for the GP grid using Rao-Blackwellization (discussed in Section 9.4.3). Each robot computes linearized convex hull optimization for navigation, subsequently moving to the closest boundary based on the temporal convex hull.

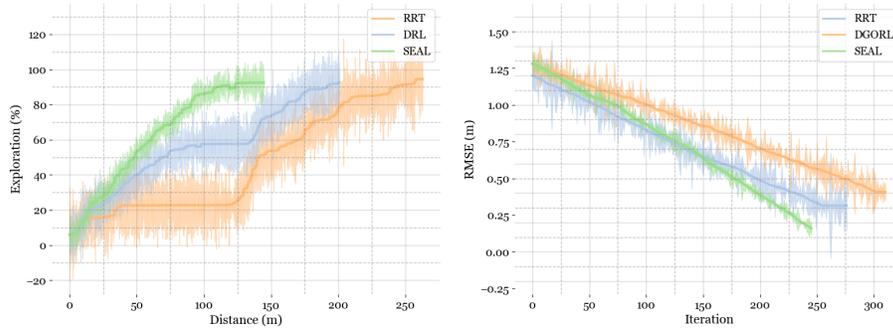


Figure 9.4: Exploration progress (**Left**) and localization error (**Right**) of the proposed SEAL, RRT (L. Zhang et al., 2020), and DRL (Hu et al., 2020) based exploration and mapping.

9.5.1 Evaluation Metrics

Exploration: The exploration approach and the method put forward by RRT (L. Zhang et al., 2020) and DRL (Hu et al., 2020) are compared with that of SEAL in the bookstore and house worlds, respectively, since they use the same environments so that we can obtain a fair comparison. Below are the used metrics for evaluation:

1. **Mapping Time:** The amount of time spent in the mapping (a measure of how effective is the multi-robot exploration);
2. **Total Distance Traveled:** This term refers to the cumulative path length of all robot's trajectory, which is necessary for a multi-robot system to explore the entire map. The entire trajectory length gives an idea of the robot's energy usage while subtly describing its investigation's effectiveness;
3. **Exploration Percentage:** The percentage of the generated map with time elapsed;
4. **Map SSIM:** Structural similarity index measure of generated maps compared with ground truth map;
5. **CPU Utilization:** The maximum percentage consumption of the processor of a robot throughout the robot's trajectory;
6. **Memory Consumption (RAM):** The maximum occupied memory by the robot throughout the trajectory;

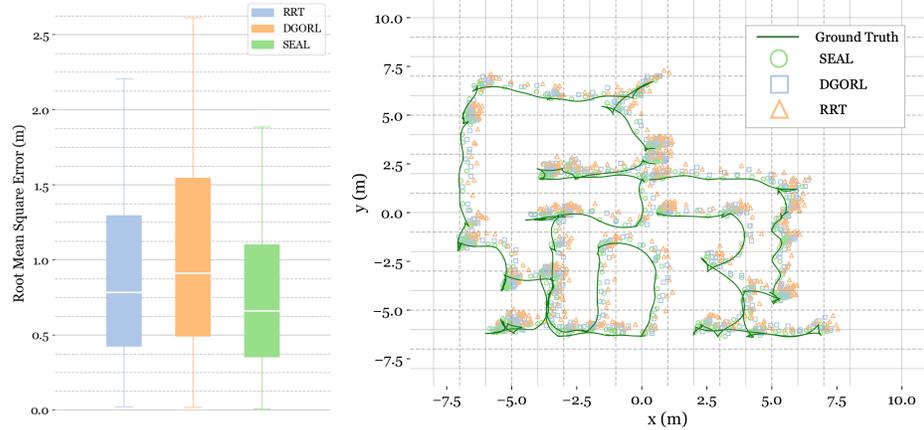


Figure 9.5: **(Left)** Localization error comparison with RRT (L. Zhang et al., 2020) and DGORL (Latif & Parasuraman, 2022a), **(Right)** Predicted trajectory of the three robots for different approaches.

7. **Communication payload (Data Size):** The maximum size of shared data by a robot.

Localization: The localization approach used RRT-Exploration (L. Zhang et al., 2020) (which used GMapping for SLAM and Map merging to merge maps and find relative localization between robots) and DGORL (Latif & Parasuraman, 2022a) (which only does relative localization) are compared in our experiments. We use the below metrics for evaluation:

1. **ATE (m):** The absolute trajectory error computed for the whole navigated trajectory by each approach, which measured the deviation from the ground truth;
2. **ALE (m):** The absolute localization error for the predicted location of each approach.

9.5.2 Exploration Results

Comprehensive listing of all results are provided in Table 9.1. To reduce measurement noise, we examined the exploration performance of each strategy after several trials. There are several simulations run in a three-robot system. Fig. 9.3 displays the mapping results of each approach made by the three robots in the simulated area. Fig. 9.3 also delineates the belief maps generated by SEAL, which are not provided by typical SLAM (or map merging) algorithms or other exploration approaches.

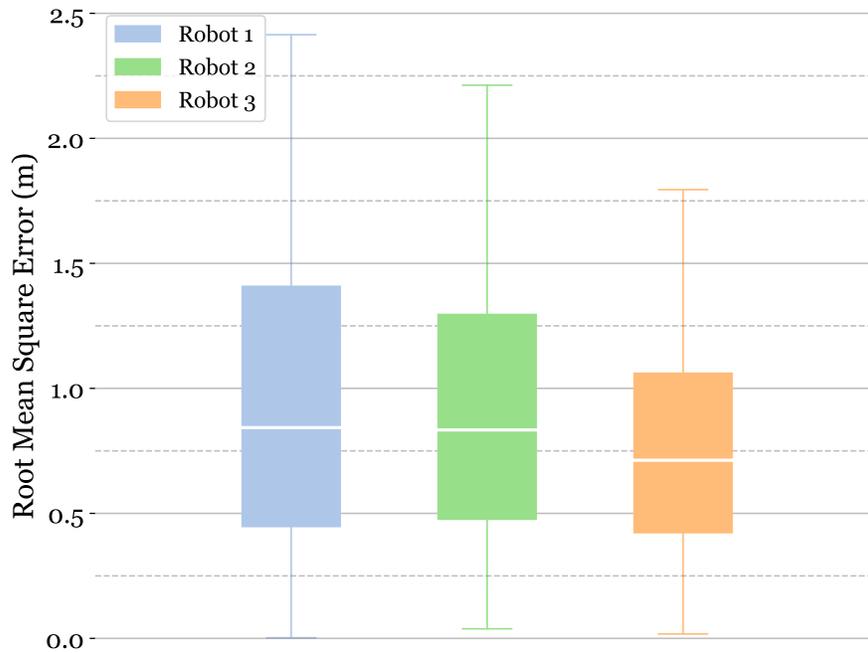


Figure 9.6: Localization performance of SEAL for the individual robot.

Note that the results consider the typical mapping time, the distance traveled, and the effectiveness of the mapping. Comparison with the original map is used to gauge mapping effectiveness and reported percentages are adjusted using gazebo world dimensions. Fig. 9.4 delineates the reduced traveled distance and achieves a high exploration percentage. It can be seen that the proposed SEAL traveled approximately half the distance and achieved a comparable exploration percentage. Furthermore, The exploration capabilities of RRT, DRL, and SEAL are thoroughly compared in Table 9.1. These results suggest that SEAL is a more effective method for exploration activities, producing superior outcomes.

9.5.3 Localization Results

We also have analyzed the localization performance of SOTA RRT and our previous approach DGORL for localization and compared results with the proposed SEAL. Fig. 9.5 has shown the trajectory by each approach and localization error in terms of RMSE. Results show that our SEAL algorithm outperformed the RRT Exploration and DGORL, as it incorporates the map belief for location correction and achieved 25% higher localization accuracy than DGORL

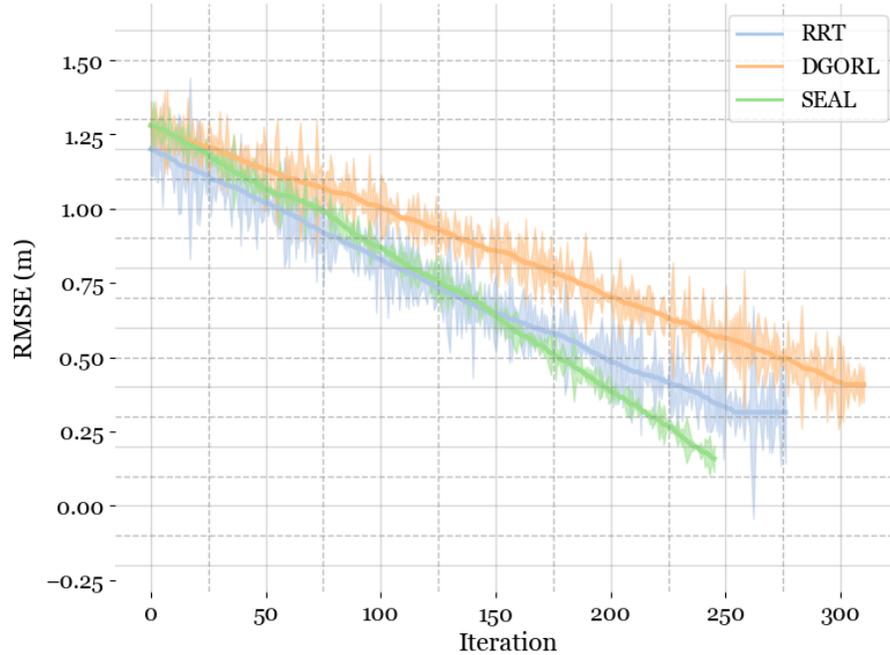


Figure 9.7: Overall localization error comparison with RRT (L. Zhang et al., 2020) and DGORL (Latif & Parasuraman, 2022a) for complete exploration.

and 10% higher than RRT. We have plotted individual robots' localization performance to better visualize the accuracy of relative localization independently in Figure 9.6. Results have shown an insignificant difference between each robot's localization accuracy, which validates the performance of the proposed SEAL even on individual robot levels. To analyze the localization performance over iterations, Fig. 9.4 provides evidence for a gradual reduction in RMSE over robotic progression towards the maximum exploration. Additionally, Table 9.1 provided exact data showing the significant improvement of the absolute trajectory and localization errors by SEAL compared to the state-of-the-art approaches. Specifically, the proposed SEAL has 25% higher localization accuracy than RRT and DGORL.

9.6 Summary

This chapter presents a novel method called simultaneous exploration and localization (SEAL) for distributed multi-robot systems to improve localization accuracy and exploration efficiency. SEAL integrates Gaussian Processes (GP) and graph optimization techniques, resulting in high localization accuracy and fast

exploration without relying on global localization information. The methodology involves choosing the next-best unexplored area for distributed exploration through linearized convex hull optimization and enhancing the posterior belief of exploration and localization through GP modeling over relative localization via the Rao-Blackwellization technique.

SEAL was compared against two state-of-the-art exploration methods: Rapidly exploring Random Tree (RRT)-based exploration and Deep Reinforcement Learning (DRL)-based exploration. For localization, it was compared to the typical Gmapping-based SLAM method and a relative localization method known as DGORL. The SEAL approach outperformed these methods' exploration and localization performances in various simulations, demonstrating its practicality in real-world applications. Specifically, it significantly reduced time complexity per robot compared to RRT and DRL while maintaining improved localization accuracy and mapping efficiency.

Notably, SEAL does not rely on boundary information for localization or external sensors, making it suitable for dynamic settings with limited features or no prior knowledge. These qualities make SEAL a potentially powerful tool for multi-robot exploration and localization in mapping, search and rescue applications, and environmental monitoring. The code for this methodology has been open-sourced as a ROS package for the robotics community to utilize and further develop. This chapter concludes the dissertation proposed solutions into an integrated solution for simultaneous exploration and localization.

CHAPTER 10

CONCLUSION

In this dissertation, we propose and explore various novel algorithms to tackle crucial problems in multi-robot systems, particularly those related to localization, exploration, and optimization challenges in ad hoc wireless networks. Our first contribution, the collaborative direction of arrival (CDOA) based instantaneous localization method, optimally utilizes wireless signals for efficient position estimation of robotic nodes in GPS-denied environments. This method demonstrated a significant improvement in localization accuracy through rigorous experimentation over traditional non-sampling-based methods and existing sampling-based techniques, offering high efficiency with less computational overhead.

We further extend our research by proposing the multi-robot synergistic localization (MRSL) algorithm. This innovative algorithm enables individual robots within a multi-robot system to cooperatively update their position estimations using pose and sensor data. The computational efficiency of this Bayesian rule-based integration method, scalability, robustness, and fault tolerance, marks it as an advance over existing multi-robot collaboration algorithms. Our third contribution is the exploration of graph theory to solve multi-robot relative localization problems. By estimating a relative pose measurement graph and utilizing graph optimization techniques, our method yielded high localization accuracy while maintaining competitive optimization times. Our fourth contribution, the GPRL approach, ingeniously combines Gaussian Process Regression with hierarchical inferencing for relative localization using RSSI data. This method demonstrated superior performance in accuracy and efficiency compared to state-of-the-art methods, offering a reliable solution for diverse robots.

In addition, we proposed a novel reinforcement learning-aided coordination algorithm for swarm robotic exploration problems. This approach, offer-

ing local information transfer in a distributed manner, demonstrated significant improvement in maze coverage performance in low communication range and high packet loss environments over conventional search-based and heuristic methods. To meet the demands of practical scenarios, our sixth contribution, CQLite, adopts a distributed Q-learning approach with reduced communication and computation overhead. The high efficiency of CQLite, as supported by empirical data, suggests its potential for applications in real-world multi-robot exploration. Finally, we developed a simultaneous exploration and localization (SEAL) approach for distributed multi-robot systems, integrating Gaussian Processes and graph optimization techniques. SEAL’s superior performance in exploration and localization tasks and its ability to function effectively in dynamic settings with no prior knowledge underscore its potential for various robotic applications.

These contributions enhance our understanding and handling of multi-robot systems’ localization and exploration challenges in ad hoc wireless networks.

10.1 Future Work

Our future work primarily focuses on expanding and enhancing the proposed algorithms to achieve better performance in real-world environments. Future work can include further enhancements to the accuracy and efficiency of these methods, overcoming their limitations by integrating additional techniques and broadening their applicability in real-world settings. Our work has laid a solid foundation and opened new avenues for continued research in collaborative algorithms for ad hoc multi-robot networks.

Firstly, for the SEAL approach discussed in Chapter 9, we intend to combine it with the Gaussian Process-based relative Localization method explored in Chapter 6. Integrating these two methods will yield more efficient solutions for multi-robotic exploration and localization tasks. Such a combination would take advantage of SEAL’s superior exploration capability and GPRL’s efficient relative localization, potentially resulting in a more effective algorithm that could tackle the exploration and localization problem synergistically.

Secondly, we plan to examine the robustness and resilience of these algorithms to various perturbations that commonly occur in real-world scenarios. To achieve this, we aim to integrate Byzantine fault detection into our multi-robot systems to handle malevolent nodes or data corruptions using a graph-theoretic approach. In this way, our multi-robot systems could still function effectively even in the face of unexpected faults or adversarial attacks, provid-

ing a more reliable solution for practical applications. This augmentation will extend our existing research on robustness and fault tolerance in multi-robot systems.

Thirdly, we plan to enhance the reinforcement learning-aided coordination algorithm for swarm robotic exploration by incorporating more complex learning mechanisms, such as deep reinforcement learning. We believe that this augmentation could potentially address the constraints faced by conventional reinforcement learning methods and further improve the maze coverage performance in low communication range and high packet loss environments. This also opens up an avenue for exploring the potential synergies between deep learning and reinforcement learning techniques in the context of swarm robotic exploration.

Lastly, future work could also include empirical validation of the proposed algorithms in different scenarios and their comparison with state-of-the-art solutions. We aim to conduct rigorous experimentation in diverse environments to showcase the practical effectiveness of our algorithms and their adaptability in various real-world settings. These investigations will not only enable us to further validate our research contributions but will also broaden their applicability by making necessary adjustments and adaptations based on the findings.

Overall, the future direction of this research lies in refining the proposed algorithms and integrating advanced techniques to ensure their robustness, efficiency, and practical applicability in handling the complex challenges associated with multi-robot systems in ad hoc wireless networks.

APPENDIX A

Below are the specific details of the methods implemented from the state-of-the-art fingerprint-less localization techniques so as to compare with the proposed CDOA methods.

A.o.1 Trilateration (Manolakis, 1996)

Trilateration is a historical model-based technique (Manolakis, 1996) that uses distances to determine the receiver's location numerically. To calculate with trilateration, we need three transmitting devices to obtain a 2-D position and four to find a 3-D position. The distances between the transmitter and the receivers and the right number of transmitting devices are necessary. A frequent method for calculating the distance between devices is to use the RSSI of a signal. For 2-D space, with three anchor nodes N_1, N_2, N_3 and positions in space be $(a_1, b_1), (a_2, b_2), (a_3, b_3)$ respectively. We can find the unknown position (x, y) of the receiver as:

$$\begin{cases} (a_1 - x)^2 + (b_1 - y)^2 = d_1^2 \\ (a_2 - x)^2 + (b_2 - y)^2 = d_2^2 \\ (a_3 - x)^2 + (b_3 - y)^2 = d_3^2 \end{cases}$$

To minimize the posting error, we need to minimize the following objective function using a non-linear least squares technique:

$$f(x, y) = \sum_{i=1}^3 \left[\sqrt{(x - a_i)^2 + (y - b_i)^2} - d_i \right]^2$$

A.o.2 Weighted Centroid (Z.-M. Wang & Zheng, 2014)

The basic idea of a weighted centroid localization algorithm (e.g., (Z.-M. Wang & Zheng, 2014)) based on RSSI is that unknown nodes gather RSSI information from the beacon nodes around them. Assuming there are n anchor nodes in the wireless sensor network, with coordinates $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, respectively, the location of the unknown node can be obtained by using the improved centroid algorithm estimating the coordinates of n nodes as:

$$\begin{cases} x = \frac{w_1 * x_1 + w_2 * x_2 + w_3 * x_3 + \dots + w_n * x_n}{w_1 + w_2 + w_3 + \dots + w_n} \\ y = \frac{w_1 * y_1 + w_2 * y_2 + w_3 * y_3 + \dots + w_n * y_n}{w_1 + w_2 + w_3 + \dots + w_n} \end{cases}$$

$$w_i = \frac{RSSI_i}{RSSI_1 + RSSI_2 + RSSI_3 + \dots + RSSI_n}$$

$$i \in (1, 2, 3, \dots, n)$$

A.o.3 Differential RSS (Podevijn et al., 2018)

The Differential RSS method in (Podevijn et al., 2018) works without knowing to transmit power beforehand. There are two phases in this technique; offline and online phases. During the offline phase, received RSS values are generated using the representative (measured) RSS model for each grid point. During the online phase, the measured DRSS values for each grid point are compared to the theoretical ones. The estimated location (X, Y) is determined as the grid point with the theoretical RSS values closest (the least squares) to the ones measured:

$$(X, Y) = \min_{x,y} \sum_{i=1}^N (DRSS_{(x,y),i,T} - DRSS_{i,M})^2$$

Here $DRSS_{(x,y),i,T}$ denotes the actual differential RSS value at position (x,y) from or at anchor $i, i = 1 \dots N$ with N is the number of anchors. $RSS_{i,M}$ is the measured RSS value from or at anchor i and is therefore required. $DRSS_0$ is obtained from a measurement at a reference point using the below equation.

$$DRSS_i = RSS_i - RSS_1$$

Here, RSS_1 denotes the most significant received signal strength. In the algorithm, n is considered constant and known.

A.o.4 Improved RSSI (W. Xue et al., 2017)

The mean or probability of the RSSI signal uses to determine the RSSI signal characteristics. However, because of multipath and non-line-of-sight propagation in a complex and dynamic interior environment, RSSI characteristics can vary significantly in time and space. For example, if multipath interference is higher than the signal, the probability distribution is right-skewed. As a result, the RSSI signal's mean does not adequately represent RSSI's dynamic nature. Therefore, the improved RSSI algorithm (W. Xue et al., 2017) works to bag RSSIs for a particular time interval to maintain the temporal correlation between observations, then extract top k values from decidedly sorted observations (value of k determined to be 13 for indoor localization). Later on, average out the extracted observation to find the finest RSSI will then be used to calculate the differential distance from the previous observation using the differential of standard signal intensity attenuation model as:

$$\Delta d = d_o 10^{\frac{A-R_t}{10\eta}} - d_o 10^{\frac{A-R_{t-1}}{10\eta}},$$

where d_o set to one meter, A is received RSSI at d_o , R_t and R_{t-1} are the processed RSSI at time t and $t - 1$ respectively. For localization, an initial position is (o, o) , and subsequent positions are estimated using the intersection of Δd measured from each access point using standard rules of trilateration.

A.o.5 Particle Filter Extended Kalman Filter (PF-EKF) (Zafari et al., 2018)

For indoor localization, Bayesian filtering is an appealing solution. Which, on the other hand, requires the system (depicts how the state changes over time.) and measurement (relates the noisy measurements (RSSI for PF and the user position for EKF) with the state/position) models. Using a recursive filter, the PF-EKF algorithm in (Zafari et al., 2018) calculates the posterior Probability Density Function (PDF). The prediction and update stages of recursive filters are where the state predicts and then updates once the measurements are available. Then, using the Bayes theorem, the updated state has gathered measurements to adjust the prediction PDF. The algorithm first uses a particle filter to find the estimated PDF in weighted random samples as:

$$p(y_i|x_{1:i}) \approx \sum_{k=1}^{N_s} w_i^k \delta(y_i - y_i^k)$$

Furthermore, we used this PDF in an extended Kalman filter to get the prediction of position and update the measurement model for future estimation as:
Predict :

$$\begin{aligned}\bar{Y}_{i-1} &= FY_{i-1} \\ \bar{P}_{i-1} &= FP_{i-1}F^T + Q\end{aligned}$$

Update :

$$\begin{aligned}K_i &= \bar{P}_{i-1}H^T(H\bar{P}_{i-1}H^T + R)^{-1} \\ \bar{Y}_i &= \bar{Y}_{i-1} + K_i(X_i - H\bar{Y}_{i-1}) \\ P_i &= \bar{P}_{i-1}(1 - KH)\end{aligned}$$

A.o.6 SBL-DOA (H. Wang et al., 2019).

The proposed method in (H. Wang et al., 2019) focuses on assisted vehicle localization based on three collaborative base stations and SBL-based robust DOA estimation. To accurately estimate the direction of arrival (DOA) for vehicle localization, the authors create the sparse Bayesian learning (SBL) technique. The main concept is to use three base stations for cooperative localization and to take advantage of the previously known sparsity in the angular space. The SBL-based robust DOA estimation problem is formulated as:

$$\begin{aligned}\underset{\theta, \alpha}{\text{minimize}} \quad & \frac{1}{2}|\mathbf{y} - \mathbf{A}(\theta)\mathbf{x}|^2 + \frac{\gamma}{2}|\mathbf{x}|_1 \\ \text{subject to} \quad & \alpha_i = \frac{1}{\sigma_i^2}, \quad i = 1, \dots, N,\end{aligned}$$

where \mathbf{y} represents the received signal, $\mathbf{A}(\theta)$ denotes the steering matrix, \mathbf{x} is the sparse vector of interest, α is the hyperparameter vector, and γ is the regularization parameter.

After obtaining the DOA estimates, the proposed method triangulates the robot's location using the base stations. For a system with three base stations, the vehicle's position (x, y) is calculated using the following set of equations:

$$\begin{aligned}d_1 &= \sqrt{(x - x_1)^2 + (y - y_1)^2} \\ d_2 &= \sqrt{(x - x_2)^2 + (y - y_2)^2} \\ d_3 &= \sqrt{(x - x_3)^2 + (y - y_3)^2},\end{aligned}$$

where (x_i, y_i) denotes the position of the i -th base station and d_i represents the distance between the vehicle and the i -th base station.

By applying the SBL-based robust DOA estimation and triangulation, the proposed approach in (H. Wang et al., 2019) achieves accurate and reliable vehicle localization in the presence of multipath and shadowing effects commonly encountered in urban environments.

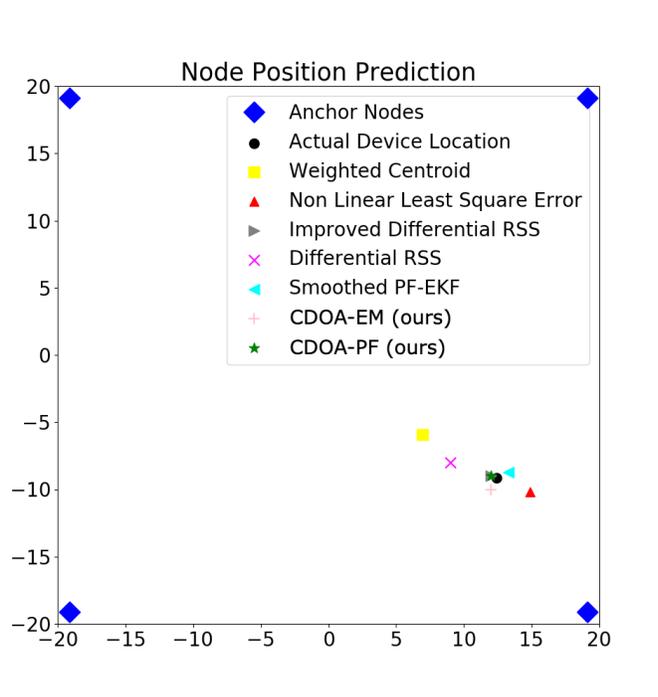


Figure A.1: A sample output comparison of different localization algorithms in a 40 x 40 m bounded region.

BIBLIOGRAPHY

- Abbas, M., Elhamshary, M., Rizk, H., Torki, M., & Youssef, M. (2019). Wideep: Wifi-based accurate and robust indoor localization system using deep learning. *2019 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 1–10.
- Alamri, S., Alshehri, S., Alshehri, W., Alamri, H., Alaklabi, A., & Alhmiedat, T. (2021). Autonomous maze solving robotics: Algorithms and systems. *International Journal of Mechanical Engineering and Robotics Research*, 10(12).
- Alarifi, A., Al-Salman, A., Alsaleh, M., Alnafessah, A., Al-Hadhrami, S., Al-Ammar, M. A., & Al-Khalifa, H. S. (2016). Ultra wideband indoor positioning technologies: Analysis and recent advances. *Sensors*, 16(5), 707.
- Alfurati, I. S., & Rashid, A. T. (2018). Performance comparison of three types of sensor matrices for indoor multi-robot localization. *International Journal of Computer Applications*, 975, 8887.
- Arbula, D., & Ljubic, S. (2020). Indoor localization based on infrared angle of arrival sensor network. *Sensors*, 20(21). <https://doi.org/10.3390/s20216278>
- Arshad, S., & Kim, G.-W. (2021). Role of deep learning in loop closure detection for visual and lidar slam: A survey. *Sensors*, 21(4), 1243.
- Aurangzeb, M., Lewis, F. L., & Huber, M. (2013). Efficient, swarm-based path finding in unknown graphs using reinforcement learning. *2013 10th IEEE International Conference on Control and Automation (ICCA)*, 870–877. <https://doi.org/10.1109/ICCA.2013.6564940>
- Bahr, A., Leonard, J. J., & Fallon, M. F. (2009). Cooperative localization for autonomous underwater vehicles. *The International Journal of Robotics Research*, 28(6), 714–728. <https://doi.org/10.1177/0278364908100561>
- Bai, S., Wang, J., Chen, F., & Englot, B. (2016). Information-theoretic exploration with bayesian optimization. *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1816–1822.

- Bar-Shalom, Y., Li, X. R., & Kirubarajan, T. (2004). *Estimation with applications to tracking and navigation: Theory algorithms and software*. John Wiley & Sons.
- Bernreiter, L., Khattak, S., Ott, L., Siegwart, R., Hutter, M., & Cadena, C. (2022). Collaborative robot mapping using spectral graph analysis. *2022 International Conference on Robotics and Automation (ICRA)*, 3662–3668.
- Bono, A., Fedele, G., & Franzè, G. (2021). A swarm-based distributed model predictive control scheme for autonomous vehicle formations in uncertain environments. *IEEE Transactions on Cybernetics*, 52(9), 8876–8886.
- Botteghi, M., Khaled, M., Sirmacek, B., & Poel, M. (2020). Entropy-based exploration for mobile robot navigation: A learning-based approach. *Planning and robotics workshop, PlanRob*.
- Bouman, A., Ginting, M. F., Alatur, N., Palieri, M., Fan, D. D., Touma, T., Pailevanian, T., Kim, S.-K., Otsu, K., Burdick, J., et al. (2020). Autonomous spot: Long-range autonomous exploration of extreme environments with legged locomotion. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2518–2525.
- Burgard, W., Moors, M., Stachniss, C., & Schneider, F. E. (2005). Coordinated multi-robot exploration. *IEEE Transactions on robotics*, 21(3), 376–386.
- Caccamo, S., Parasuraman, R., Båberg, F., & Ogren, P. (2015). Extending a ugv teleoperation flc interface with wireless network connectivity information. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4305–4312.
- Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., & Leonard, J. J. (2016). Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32(6), 1309–1332.
- Canedo-Rodriguez, A., Alvarez-Santos, V., Regueiro, C. V., Iglesias, R., Barro, S., & Presedo, J. (2016). Particle filter robot localisation through robust fusion of laser, wifi, compass, and a network of external cameras. *Information Fusion*, 27, 170–188.
- Carlevaris-Bianco, N., & Eustice, R. M. (2013). Generic factor-based node marginalization and edge sparsification for pose-graph slam. *2013 IEEE International Conference on Robotics and Automation*, 5748–5755.
- Celaya-Echarri, M., Azpilicueta, L., Lopez-Iturri, P., Picallo, I., Aguirre, E., Astrain, J. J., Villadangos, J., & Falcone, F. (2020). Radio wave prop-

- agation and wsn deployment in complex utility tunnel environments. *Sensors*, 20(23), 6710.
- Chang, T.-K., Chen, K., & Mehta, A. (2021). Resilient and consistent multi-robot cooperative localization with covariance intersection. *IEEE Transactions on Robotics*, 38(1), 197–208.
- Chen, F., Martin, J. D., Huang, Y., Wang, J., & Englot, B. (2020). Autonomous exploration under uncertainty via deep reinforcement learning on graphs. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 6140–6147.
- Chen, F., Szenher, P., Huang, Y., Wang, J., Shan, T., Bai, S., & Englot, B. (2021). Zero-shot reinforcement learning on graphs for autonomous exploration under uncertainty. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 5193–5199.
- Chen, S., Yin, D., & Niu, Y. (2022). A survey of robot swarms' relative localization method. *Sensors*, 22(12), 4424.
- Chen, Y.-H., & Wu, C.-M. (2020). An improved algorithm for searching maze based on depth-first search. *2020 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-Taiwan)*, 1–2.
- Choi, J., Park, K., Kim, M., & Seok, S. (2019). Deep reinforcement learning of navigation in a complex and crowded environment with a limited field of view. *2019 International Conference on Robotics and Automation (ICRA)*, 5993–6000.
- Corah, M., O'Meadhra, C., Goel, K., & Michael, N. (2019). Communication-efficient planning and mapping for multi-robot exploration in large environments. *IEEE Robotics and Automation Letters*, 4(2), 1715–1721.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Introduction to algorithms*. MIT press.
- Dai, A., Papatheodorou, S., Funk, N., Tzoumanikas, D., & Leutenegger, S. (2020). Fast frontier-based information-driven autonomous exploration with an mav. *2020 IEEE international conference on robotics and automation (ICRA)*, 9570–9576.
- Dong, Q., & Dargie, W. (2012a). Evaluation of the reliability of rssi for indoor localization. *2012 International Conference on Wireless Communications in Underground and Confined Areas, ICWCUCA 2012*, 1–6. <https://doi.org/10.1109/ICWCUCA.2012.6402492>
- Dong, Q., & Dargie, W. (2012b). Evaluation of the reliability of rssi for indoor localization. *2012 International Conference on Wireless Communications in Underground and Confined Areas*, 1–6.

- Dube, R., Gawel, A., Sommer, H., Nieto, J., Siegwart, R., & Cadena, C. (2017). An online multi-robot slam system for 3d lidars. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1004–1011.
- Duda, P., Jaworski, M., Pietruczuk, L., & Rutkowski, L. (2014). A novel application of hoeffding’s inequality to decision trees construction for data streams. *2014 International Joint Conference on Neural Networks (IJCNN)*, 3324–3330.
- Elgui, K., Bianchi, P., Portier, F., & Isson, O. (2020). Learning methods for rssi-based geolocation: A comparative study. *Pervasive and Mobile Computing*, 67, 101199.
- Faigl, J., Krajník, T., Chudoba, J., Přeučil, L., & Saska, M. (2013). Low-cost embedded system for relative localization in robotic swarms. *2013 IEEE International Conference on Robotics and Automation*, 993–998.
- Fang, B., Ding, J., & Wang, Z. (2019). Autonomous robotic exploration based on frontier point optimization and multistep path planning. *IEEE Access*, 7, 46104–46113.
- Fang, W., Zhang, W., Yang, W., Li, Z., Gao, W., & Yang, Y. (2021). Trust management-based and energy efficient hierarchical routing protocol in wireless sensor networks. *Digital Communications and Networks*, 7(4), 470–478.
- Fink, J., Michael, N., Kim, S., & Kumar, V. (2012). Distributed pursuitevasion without global localization via local frontiers. *Autonomous Robots*, 32(1), 81–95.
- Fink, J., & Kumar, V. (2010). Online methods for radio signal mapping with mobile robots. *2010 IEEE International Conference on Robotics and Automation*, 1940–1945.
- Fox, D., Burgard, W., Kruppa, H., & Thrun, S. (1999). Collaborative multi-robot localization. In *Mustererkennung 1999* (pp. 15–26). Springer. https://doi.org/10.1007/3-540-48238-5_21
- Gao, W., Booker, M., Adiwahono, A., Yuan, M., Wang, J., & Yun, Y. W. (2018). An improved frontier-based approach for autonomous exploration. *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 292–297.
- Gao, Y., Wang, Y., Zhong, X., Yang, T., Wang, M., Xu, Z., Wang, Y., Lin, Y., Xu, C., & Gao, F. (2022). Meeting-merging-mission: A multi-robot coordinate framework for large-scale communication-limited exploration. *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 13700–13707.

- Ghaffari Jadidi, M., Valls Miro, J., & Dissanayake, G. (2018). Gaussian processes autonomous mapping and exploration for range-sensing mobile robots. *Autonomous Robots*, 42, 273–290.
- Ghazal, T. (2022). Data fusion-based machine learning architecture for intrusion detection. *Computers, Materials & Continua*, 70(2), 3399–3413.
- Gielis, J., Shankar, A., & Prorok, A. (2022). A critical review of communications in multi-robot systems. *Current Robotics Reports*, 1–13.
- Grisetti, G., Stachniss, C., & Burgard, W. (2007). Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE transactions on Robotics*, 23(1), 34–46.
- Gu, S., & Mao, G. (2021). An improved q-learning algorithm for path planning in maze environments. *Intelligent Systems and Applications: Proceedings of the 2020 Intelligent Systems Conference (IntelliSys) Volume 2*, 547–557.
- Gui, Y., Latif, E., & Parasuraman, R. (2022). Message expiration-based distributed multi-robot task management. *6th International Symposium on Swarm Behavior and Bio-Inspired Robotics*.
- Guo, K., Qiu, Z., Meng, W., Xie, L., & Teo, R. (2017). Ultra-wideband based cooperative relative localization algorithm and experiments for multiple unmanned aerial vehicles in gps denied environments. *International Journal of Micro Air Vehicles*, 9(3), 169–186.
- Hamer, M., & D'Andrea, R. (2018). Self-calibrating ultra-wideband network supporting multi-robot localization. *IEEE Access*, 6, 22292–22304. <https://doi.org/10.1109/ACCESS.2018.2829020>
- Han, D., Andersen, D. G., Kaminsky, M., Papagiannaki, K., & Seshan, S. (2009). Access point localization using local signal strength gradient. *International Conference on Passive and active network measurement*, 99–108.
- Han, R., Chen, S., & Hao, Q. (2020). Cooperative multi-robot navigation in dynamic environment with deep reinforcement learning. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 448–454.
- Hao, N., He, F., Hou, Y., & Yao, Y. (2022). Graph-based observability analysis for mutual localization in multi-robot systems. *Systems & Control Letters*, 161, 105152. <https://doi.org/https://doi.org/10.1016/j.sysconle.2022.105152>
- Hassani, A., Bertrand, A., & Moonen, M. (2015). Cooperative integrated noise reduction and node-specific direction-of-arrival estimation in a fully connected wireless acoustic sensor network. *Signal Processing*, 107, 68–81.

- He, G., Yuan, X., Zhuang, Y., & Hu, H. (2020). An integrated gnss/lidar-slam pose estimation framework for large-scale map building in partially gnss-denied environments. *IEEE Transactions on Instrumentation and Measurement*, 70, 1–9.
- Hoang, M. T., Yuen, B., Dong, X., Lu, T., Westendorp, R., & Reddy, K. (2019). Recurrent neural networks for accurate rssi indoor localization. *IEEE Internet of Things Journal*, 6(6), 10639–10651.
- Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., & Burgard, W. (2013). Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous robots*, 34, 189–206.
- Hsieh, C.-H., Chen, J.-Y., & Nien, B.-H. (2019). Deep learning-based indoor localization using received signal strength and channel state information. *IEEE access*, 7, 33256–33267.
- Hu, J., Niu, H., Carrasco, J., Lennox, B., & Arvin, F. (2020). Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning. *IEEE Transactions on Vehicular Technology*, 69(12), 14413–14423.
- Huang, G., Kaess, M., & Leonard, J. J. (2013). Consistent sparsification for graph optimization. *2013 European Conference on Mobile Robots*, 150–157. <https://doi.org/10.1109/ECMR.2013.6698835>
- Indelman, V., Nelson, E., Michael, N., & Dellaert, F. (2014). Multi-robot pose graph localization and data association from unknown initial relative poses via expectation maximization. *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 593–600.
- Islam, M. J., Mo, J., & Sattar, J. (2021). Robot-to-robot relative pose estimation using humans as markers. *Autonomous Robots*, 45(4), 579–593.
- Jadhav, N., Wang, W., Zhang, D., Kumar, S., & Gil, S. (2021). Toolbox release: A wifi-based relative bearing sensor for robotics. *arXiv preprint arXiv:2109.12205*.
- Jadhav, N., Wang, W., Zhang, D., Kumar, S., & Gil, S. (2022). Toolbox release: A wifi-based relative bearing framework for robotics. *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 13714–13721.
- Jadidi, M. G., Miro, J. V., & Dissanayake, G. (2015). Mutual information-based exploration on continuous occupancy maps. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 6086–6092.

- Jadidi, M. G., Miro, J. V., Valencia, R., & Andrade-Cetto, J. (2014). Exploration on continuous gaussian process frontier maps. *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 6077–6082.
- Jia, Y., Luo, H., Zhao, F., Jiang, G., Li, Y., Yan, J., Jiang, Z., & Wang, Z. (2021). Lvio-fusion: A self-adaptive multi-sensor fusion slam framework using actor-critic method. *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 286–293.
- Jiang, X., Zeng, X., Sun, J., & Chen, J. (2022). Distributed solver for discrete-time lyapunov equations over dynamic networks with linear convergence rate. *IEEE Transactions on Cybernetics*, 52(2), 937–946. <https://doi.org/10.1109/TCYB.2020.2989835>
- Johannsson, H., Kaess, M., Fallon, M., & Leonard, J. J. (2013). Temporally scalable visual slam using a reduced pose graph. *2013 IEEE International Conference on Robotics and Automation*, 54–61.
- Kahn, G., Villafior, A., Ding, B., Abbeel, P., & Levine, S. (2018). Self-supervised deep reinforcement learning with generalized computation graphs for robot navigation. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 5129–5136.
- Kalinowska, A., Davoodi, E., Strub, F., Mathewson, K. W., Kajic, I., Bowling, M., Murphey, T. D., & Pilarski, P. M. (2022). Over-communicate no more: Situated rl agents learn concise communication protocols. *arXiv preprint arXiv:2211.01480*.
- Kantasewi, N., Marukatat, S., Thainimit, S., & Manabu, O. (2019). Multi q-table q-learning. *2019 10th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES)*, 1–7.
- Keidar, M., & Kaminka, G. A. (2014). Efficient frontier detection for robot exploration. *The International Journal of Robotics Research*, 33(2), 215–236.
- Khan, T., Ramuhalli, P., & Dass, S. C. (2011). Particle-filter-based multisensor fusion for solving low-frequency electromagnetic nde inverse problems. *IEEE Transactions on Instrumentation and Measurement*, 60(6), 2142–2153. <https://doi.org/10.1109/TIM.2011.2117170>
- Kshirsagar, J., Shue, S., & Conrad, J. M. (2018). A survey of implementation of multi-robot simultaneous localization and mapping. *SoutheastCon 2018*, 1–7. <https://doi.org/10.1109/SECON.2018.8478985>
- Kummerle, R., Grisetti, G., Strasdat, H., Konolige, K., & Burgard, W. (2011). G2o: A general framework for graph optimization. *2011 IEEE International Conference on Robotics and Automation*, 3607–3613.

- Latif, E., Gui, Y., Munir, A., & Parasuraman, R. (2022). Energy-aware multi-robot task allocation in persistent tasks. *6th International Symposium on Swarm Behavior and Bio-Inspired Robotics*.
- Latif, E., & Parasuraman, R. (2022a). Dgorl: Distributed graph optimization based relative localization of multi-robot systems. <https://doi.org/10.48550/arXiv.2210.01662>
- Latif, E., & Parasuraman, R. (2022b). Multi-robot synergistic localization in dynamic environments. *ISR Europe 2022; 54th International Symposium on Robotics*, 1–8.
- Latif, E., & Parasuraman, R. (2023a). Cqlite: Communication-efficient multi-robot exploration using coverage-biased distributed q-learning. *arXiv preprint arXiv:2307.00500*.
- Latif, E., & Parasuraman, R. (2023b). Instantaneous wireless robotic node localization using collaborative direction of arrival. *arXiv preprint arXiv:2307.01956*.
- Latif, E., & Parasuraman, R. (2023c). Seal: Simultaneous exploration and localization in multi-robot systems. *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. <https://doi.org/10.48550/arXiv.2306.12623>
- Latif, E., Song, W., & Parasuraman, R. (2023). Communication-efficient reinforcement learning in swarm robotic networks for maze exploration. *Proceedings of 16th International Workshop on Wireless Sensing and Actuating Robotic Networks (WISARN 2023)*. <https://doi.org/10.48550/arXiv.2305.17087>
- Li, H., Zhang, Q., & Zhao, D. (2019). Deep reinforcement learning-based automatic exploration for navigation in unknown environment. *IEEE transactions on neural networks and learning systems*, 31(6), 2064–2076.
- Li, J. (2020). Faster parallel algorithm for approximate shortest path. *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, 308–321.
- Li, Q., Gama, F., Ribeiro, A., & Prorok, A. (2020). Graph neural networks for decentralized multi-robot path planning. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 11785–11792.
- Li, S., De Wagter, C., & De Croon, G. C. (2022). Self-supervised monocular multi-robot relative localization with efficient deep neural networks. *2022 International Conference on Robotics and Automation (ICRA)*, 9689–9695.
- Li, X., Sun, H., Jiang, L., Shi, Y., & Wu, Y. (2015). Modified particle filtering algorithm for single acoustic vector sensor doa tracking. *Sensors*, 15(10), 26198–26211.

- Liu, F., Liu, J., Yin, Y., Wang, W., Hu, D., Chen, P., & Niu, Q. (2020a). Survey on wifi-based indoor positioning techniques. *IET Communications*, *14*(9), 1372–1383. <https://doi.org/https://doi.org/10.1049/iet-com.2019.1059>
- Liu, F., Liu, J., Yin, Y., Wang, W., Hu, D., Chen, P., & Niu, Q. (2020b). Survey on wifi-based indoor positioning techniques. *IET Communications*, *14*(9), 1372–1383. <https://doi.org/https://doi.org/10.1049/iet-com.2019.1059>
- Liu, J., Chen, K., Liu, R., Yang, Y., Wang, Z., & Zhang, J. (2022). Robust and accurate multi-agent slam with efficient communication for smart mobiles. *2022 International Conference on Robotics and Automation (ICRA)*, 2782–2788.
- Luis, N., Pereira, T., Fernandez, S., Moreira, A., Borrajo, D., & Veloso, M. (2020). Using pre-computed knowledge for goal allocation in multi-agent planning. *Journal of Intelligent & Robotic Systems*, *98*(1), 165–190.
- Luo, S., Kim, J., Parasuraman, R., Bae, J. H., Matson, E. T., & Min, B.-C. (2019). Multi-robot rendezvous based on bearing-aided hierarchical tracking of network topology. *Ad Hoc Networks*, *86*, 131–143.
- Luo, W., & Sycara, K. (2018). Adaptive sampling and online learning in multi-robot sensor coverage with mixture of gaussian processes. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 6359–6364.
- Mackey, A., Spachos, P., Song, L., & Plataniotis, K. N. (2020). Improving ble beacon proximity estimation accuracy through bayesian filtering. *IEEE Internet of Things Journal*, *7*(4), 3160–3169.
- Madhavan, R., Fregene, K., & Parker, L. (2002). Distributed heterogeneous outdoor multi-robot localization. *Proceedings 2002 IEEE International Conference on Robotics and Automation*, *1*, 374–381 vol.1. <https://doi.org/10.1109/ROBOT.2002.1013389>
- Mangelson, J. G., Dominic, D., Eustice, R. M., & Vasudevan, R. (2018). Pairwise consistent measurement set maximization for robust multi-robot map merging. *2018 IEEE international conference on robotics and automation (ICRA)*, 2916–2923.
- Manolakis, D. (1996). Efficient solution and performance analysis of 3-d position estimation by trilateration. *IEEE Transactions on Aerospace and Electronic Systems*, *32*(4), 1239–1248. <https://doi.org/10.1109/7.543845>
- Martinelli, A., Pont, F., & Siegwart, R. (2005). Multi-robot localization using relative observations. *Proceedings of the 2005 IEEE international conference on robotics and automation*, 2797–2802.

- Masaba, K., & Li, A. Q. (2021). Gvgexp: Communication-constrained multi-robot exploration system based on generalized voronoi graphs. *2021 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, 146–154.
- Mokhtarzadeh, H., & Gebre-Egziabher, D. (2014). Cooperative inertial navigation. *Navigation, Journal of the Institute of Navigation*, 61(2), 77–94. <https://doi.org/10.1002/navi.60>
- Morales, J., & Kassas, Z. M. (2018). A low communication rate distributed inertial navigation architecture with cellular signal aiding. *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, 1–6.
- Motroni, A., Buffi, A., & Nepa, P. (2021). A survey on indoor vehicle localization through rfid technology. *IEEE Access*, 9. <https://doi.org/10.1109/ACCESS.2021.3052316>
- Najafi, M., Nadealian, Z., Rahmanian, S., & Ghafarinia, V. (2019). An adaptive distributed approach for the real-time vision-based navigation system. *Measurement*, 145, 14–21.
- Olcay, E., Schuhmann, F., & Lohmann, B. (2020). Collective navigation of a multi-robot system in an unknown environment. *Robotics and Autonomous Systems*, 132, 103604.
- Owen-Hill, A., Parasuraman, R., & Ferre, M. (2013). Haptic teleoperation of mobile robots for augmentation of operator perception in environments with low-wireless signal. *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 1–7.
- Ozdemir, A., Gauci, M., Kolling, A., Hall, M. D., & Groß, R. (2019). Spatial coverage without computation. *2019 International Conference on Robotics and Automation (ICRA)*, 9674–9680.
- Pajovic, M., Orlik, P., Koike-Akino, T., Kim, K. J., Aikawa, H., & Hori, T. (2015a). An unsupervised indoor localization method based on received signal strength (rss) measurements. *2015 IEEE Global Communications Conference (GLOBECOM)*.
- Pajovic, M., Orlik, P., Koike-Akino, T., Kim, K. J., Aikawa, H., & Hori, T. (2015b). An unsupervised indoor localization method based on received signal strength (rss) measurements. *2015 IEEE Global Communications Conference (GLOBECOM)*, 1–6.
- Parashar, R., & Parasuraman, R. (2020). Particle filter based localization of access points using direction of arrival on mobile robots. *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*, 1–6.
- Parasuraman, R., Fabry, T., Kershaw, K., & Ferre, M. (2013). Spatial sampling methods for improved communication for wireless relay robots. *2013 In-*

- ternational Conference on Connected Vehicles and Expo (ICCVE), 874–880.
- Parasuraman, R., Kershaw, K., & Ferre, M. (2013). Experimental investigation of radio signal propagation in scientific facilities for telerobotic applications. *International Journal of Advanced Robotic Systems*, 10(10), 364.
- Parasuraman, R., & Min, B.-C. (2019). Consensus control of distributed robots using direction of arrival of wireless signals. In *Distributed autonomous robotic systems* (pp. 17–34). Springer.
- Parasuraman, R., Ogren, P., & Min, B.-C. (2018). Kalman filter based spatial prediction of wireless connectivity for autonomous robots and connected vehicles. *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, 1–5.
- Passafiume, M., Maddio, S., Lucarelli, M., & Cidronali, A. (2016). An enhanced triangulation algorithm for a distributed rssi-doa positioning system. *2016 European Radar Conference (EuRAD)*, 185–188.
- Penna, F., & Cabric, D. (2011). Bounds and tradeoffs for cooperative doa-only localization of primary users. *2011 IEEE Global Telecommunications Conference-GLOBECOM 2011*, 1–5.
- Pham, V., Laird, C., & El-Halwagi, M. (2009). Convex hull discretization approach to the global optimization of pooling problems. *Industrial & Engineering Chemistry Research*, 48(4), 1973–1979.
- Pinto, B., Barreto, R., Souto, E., & Oliveira, H. (2021). Robust rssi-based indoor positioning system using k-means clustering and bayesian estimation. *IEEE Sensors Journal*, 21(21), 24462–24470.
- Placed, J. A., & Castellanos, J. A. (2021). Fast autonomous robotic exploration using the underlying graph structure. *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 6672–6679.
- Podevijn, N., Plets, D., Trogh, J., Karaagac, A., Haxhibcqiri, J., Hoebeke, J., Martens, L., Suanet, P., & Joseph, W. (2018). Performance comparison of rssi algorithms for indoor localization in large open environments. *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 1–6. <https://doi.org/10.1109/IPIN.2018.8533695>
- Pozza, N. D., Buffoni, L., Martina, S., & Caruso, F. (2021). Quantum reinforcement learning: The maze problem. *arXiv preprint arXiv:2108.04490*.
- Prorok, A., Bahr, A., & Martinoli, A. (2012). Low-cost collaborative localization for large-scale multi-robot systems. *2012 IEEE International Conference on Robotics and Automation*, 4236–4241.

- Prorok, A., & Martinoli, A. (2011). A reciprocal sampling algorithm for lightweight distributed multi-robot localization. *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3241–3247.
- Qin, H., Meng, Z., Meng, W., Chen, X., Sun, H., Lin, F., & Ang, M. H. (2019). Autonomous exploration and mapping system using heterogeneous uavs and ugvs in gps-denied environments. *IEEE Transactions on Vehicular Technology*, 68(2), 1339–1350.
- Queralt, J. P., Taipalmaa, J., Pullinen, B. C., Sarker, V. K., Gia, T. N., Tenhunen, H., Gabbouj, M., Raitoharju, J., & Westerlund, T. (2020). Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision. *Ieee Access*, 8, 191617–191643.
- Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A. Y., et al. (2009). Ros: An open-source robot operating system. *ICRA workshop on open source software*, 3(3.2), 5.
- Quinonero-Candela, J., & Rasmussen, C. E. (2005). A unifying view of sparse approximate gaussian process regression. *The Journal of Machine Learning Research*, 6, 1939–1959.
- Ranganathan, P., & Nygard, K. (2010). Time synchronization in wireless sensor networks: A survey. *Int. J. UbiComp*, 1(2), 92–102.
- Rao, C. R. (2008). Rao-blackwell theorem. *Scholarpedia*, 3(8), 7039.
- Richter, C., & Roy, N. (2017). Safe visual navigation via deep learning and novelty detection. *Proceedings of Robotics: Science and Systems (RSS)*.
- Ridolfi, M., Fontaine, J., Herbruggen, B. V., Joseph, W., Hoebeke, J., & Poorter, E. D. (2021). Uwb anchor nodes self-calibration in nlos conditions: A machine learning and adaptive phy error correction approach. *Wireless Networks*, 27(4), 3007–3023.
- Rizzo, C., Seco, T., Espelosin, J., Lera, F., & Villarroel, J. L. (2021). An alternative approach for robot localization inside pipes using rf spatial fadings. *Robotics and Autonomous Systems*, 136, 103702.
- Robert, C. P., & Roberts, G. O. (2021). Rao-blackwellization in the mcmc era. *arXiv preprint arXiv:2101.01011*.
- Rone, W., & Ben-Tzvi, P. (2013). Mapping, localization and motion planning in mobile multi-robotic systems. *Robotica*, 31(1), 1–23.
- Roumeliotis, S., & Bekey, G. (2002). Distributed multirobot localization. *IEEE Transactions on Robotics and Automation*, 18(5), 781–795. <https://doi.org/10.1109/TRA.2002.803461>
- Sadhu, A. K., & Konar, A. (2018). An efficient computing of correlated equilibrium for cooperative Q-learning-based multi-robot planning. *IEEE*

Transactions on Systems, Man, and Cybernetics: Systems, 50(8), 2779–2794.

- Sadowski, S., Spachos, P., & Plataniotis, K. N. (2020). Memoryless techniques and wireless technologies for indoor localization with the internet of things. *IEEE Internet of Things Journal*, 7(11), 10996–11005. <https://doi.org/10.1109/JIOT.2020.2992651>
- Sahawneh, L. R., & Brink, K. M. (2017). Factor graphs-based multi-robot cooperative localization: A study of shared information influence on optimization accuracy and consistency. *Proceedings of the 2017 international technical meeting of the institute of navigation (ION ITM 2017)*, Monterey, California, 819–838.
- Schmuck, P., & Chli, M. (2019). Ccm-slam: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams. *Journal of Field Robotics*, 36(4), 763–781.
- Serra-Gomez, A., Brito, B., Zhu, H., Chung, J. J., & Alonso-Mora, J. (2020). With whom to communicate: Learning efficient communication for multi-robot collision avoidance. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 11770–11776.
- Shi, L., Li, S., Zheng, Q., Yao, M., & Pan, G. (2020). Efficient novelty search through deep reinforcement learning. *IEEE Access*, 8, 128809–128818.
- Shi, W., Li, J., Cheng, N., Lyu, F., Zhang, S., Zhou, H., & Shen, X. (2019). Multi-drone 3-d trajectory planning and scheduling in drone-assisted radio access networks. *IEEE Transactions on Vehicular Technology*, 68(8), 8145–8158.
- Shi, Y., Wang, N., Zheng, J., Zhang, Y., Yi, S., Luo, W., & Sycara, K. (2020). Adaptive informative sampling with environment partitioning for heterogeneous multi-robot systems. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 11718–11723.
- Shorinwa, O., Yu, J., Halsted, T., Koufos, A., & Schwager, M. (2020). Distributed multi-target tracking for autonomous vehicle fleets. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 3495–3501. <https://doi.org/10.1109/ICRA40945.2020.9197241>
- Shrestha, R., Tian, F.-P., Feng, W., Tan, P., & Vaughan, R. (2019). Learned map prediction for enhanced mobile robot exploration. *2019 International Conference on Robotics and Automation (ICRA)*, 1197–1204.
- Song, Q., Guo, S., Liu, X., & Yang, Y. (2017). Csi amplitude fingerprinting-based nb-iot indoor localization. *IEEE Internet of Things Journal*, 5(3), 1494–1504.

- Starks, M., Gupta, A., OV, S. S., & Parasuraman, R. (2023). Heroswarm: Fully-capable miniature swarm robot hardware design with open-source ros support. *2023 IEEE/SICE International Symposium on System Integration (SII)*, 1–7.
- Subedi, S., & Pyun, J.-Y. (2020). A survey of smartphone-based indoor positioning system using rf-based wireless technologies. *Sensors*, *20*(24). <https://doi.org/10.3390/s20247230>
- Sun, D., Geißer, F., & Nebel, B. (2016a). Towards effective localization in dynamic environments. *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4517–4523. <https://doi.org/10.1109/IROS.2016.7759665>
- Sun, D., Geißer, F., & Nebel, B. (2016b). Towards effective localization in dynamic environments. *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4517–4523. <https://doi.org/10.1109/IROS.2016.7759665>
- Surynek, P., Felner, A., Stern, R., & Boyarski, E. (2016). Efficient sat approach to multi-agent path finding under the sum of costs objective. *Proceedings of the twenty-second european conference on artificial intelligence*, 810–818.
- Swiler, L. P., Gulian, M., Frankel, A. L., Safta, C., & Jakeman, J. D. (2020). A survey of constrained gaussian process regression: Approaches and implementation challenges. *Journal of Machine Learning for Modeling and Computing*, *1*(2).
- Tao, Y., & Zhao, L. (2018). A novel system for wifi radio map automatic adaptation and indoor positioning. *IEEE Transactions on Vehicular Technology*, *67*(11), 10683–10692.
- Tardioli, D., Mosteo, A. R., Riazuelo, L., Villarroel, J. L., & Montano, L. (2010). Enforcing network connectivity in robot team missions. *The International Journal of Robotics Research*, *29*(4), 460–480.
- Tian, Y., Chang, Y., Arias, F. H., Nieto-Granda, C., How, J. P., & Carlone, L. (2022). Kimera-multi: Robust, distributed, dense metric-semantic slam for multi-robot systems. *IEEE Transactions on Robotics*.
- Tjiharjadi, S., Razali, S., & Sulaiman, H. A. (2022). A systematic literature review of multi-agent pathfinding for maze research. *Journal of Advances in Information Technology Vol*, *13*(4).
- Tjiharjadi, S., Wijaya, M., & Setiawan, E. (2017). Optimization maze robot using a* and flood fill algorithm. *International Journal of Mechanical Engineering and Robotics Research*, *6*(5), 366–372.

- Tolstaya, E., Paulos, J., Kumar, V., & Ribeiro, A. (2021). Multi-robot coverage and exploration using spatial graph neural networks. *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 8944–8950.
- Twigg, J. N., Fink, J. R., Paul, L. Y., & Sadler, B. M. (2012). Rss gradient-assisted frontier exploration and radio source localization. *2012 IEEE International Conference on Robotics and Automation*, 889–895.
- Ullah, I., Su, X., Zhu, J., Zhang, X., Choi, D., & Hou, Z. (2020). Evaluation of localization by extended kalman filter, unscented kalman filter, and particle filter-based techniques. *Wireless Communications and Mobile Computing*, 2020.
- Uwano, F., & Takadama, K. (2017). Communication-less cooperative q-learning agents in maze problem. In *Intelligent and evolutionary systems* (pp. 453–467). Springer.
- Uykan, Z. (2019). On the working principle of the hopfield neural networks and its equivalence to the gadia in optimization. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9), 3294–3304.
- Valencia, R., Andrade-Cetto, J., Valencia, R., & Andrade-Cetto, J. (2018). Active pose slam. *Mapping, Planning and Exploration with Pose SLAM*, 89–108.
- Verma, G., Dagefu, F. T., Sadler, B. M., & Twigg, J. (2018). Direction of arrival estimation with the received signal strength gradient. *IEEE Transactions on Vehicular Technology*, 67(11), 10856–10870.
- Viseras, A., Losada, R. O., & Merino, L. (2016). Planning with ants: Efficient path planning with rapidly exploring random trees and ant colony optimization. *International Journal of Advanced Robotic Systems*, 13(5), 1729881416664078.
- Wai, H.-T., Yang, Z., Wang, Z., & Hong, M. (2018). Multi-agent reinforcement learning via double averaging primal-dual optimization. *Advances in Neural Information Processing Systems*, 31.
- Wanasinghe, T. R., I. Mann, G. K., & Gosine, R. G. (2015). Relative localization approach for combined aerial and ground robotic system. *Journal of Intelligent & Robotic Systems*, 77, 113–133.
- Wanasinghe, T. R., Mann, G. K., & Gosine, R. G. (2014a). Distributed collaborative localization for a heterogeneous multi-robot system. *2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE)*, 1–6.

- Wanasinghe, T. R., Mann, G. K., & Gosine, R. G. (2014b). A jacobian free approach for multi-robot relative localization. *2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE)*, 1–6.
- Wang, C., Xu, A., Kuang, J., Sui, X., Hao, Y., & Niu, X. (2021). A high-accuracy indoor localization system and applications based on tightly coupled uwb/ins/floor map integration. *IEEE Sensors Journal*, 21(16), 18166–18177.
- Wang, C., Luo, J., Liu, X., & He, X. (2021). Secure and reliable indoor localization based on multitask collaborative learning for large-scale buildings. *IEEE Internet of Things Journal*, 9(22), 22291–22303.
- Wang, D., Devin, C., Cai, Q.-Z., Yu, F., & Darrell, T. (2019). Deep object-centric policies for autonomous driving. *2019 International Conference on Robotics and Automation (ICRA)*, 8853–8859.
- Wang, H., Wan, L., Dong, M., Ota, K., & Wang, X. (2019). Assistant vehicle localization based on three collaborative base stations via sbl-based robust doa estimation. *IEEE Internet of Things Journal*, 6(3), 5766–5777.
- Wang, J., Hwang, J. G., Peng, J., Park, J., & Park, J. G. (2021). Gaussian filtered rssi-based indoor localization in wlan using bootstrap filter. *2021 International Conference on Electronics, Information, and Communication (ICEIC)*, 1–4.
- Wang, J., & Olson, E. (2016). Apriltag 2: Efficient and robust fiducial detection. *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4193–4198.
- Wang, L., & Guo, Q. (2019). Coordination of multiple autonomous agents using naturally generated languages in task planning. *Applied Sciences*, 9(17), 3571.
- Wang, S., Wang, Y., Li, D., & Zhao, Q. (2023). Distributed relative localization algorithms for multi-robot networks: A survey. *Sensors*, 23(5), 2399.
- Wang, W., Jadhav, N., Vohs, P., Hughes, N., Mazumder, M., & Gil, S. (2019). Active rendezvous for multi-robot pose graph optimization using sensing over wi-fi. *The International Symposium of Robotics Research*, 832–849.
- Wang, X., Gao, L., & Mao, S. (2016). Csi phase fingerprinting for indoor localization with a deep learning approach. *IEEE Internet of Things Journal*, 3(6), 1113–1123.
- Wang, Z.-M., & Zheng, Y. (2014). The study of the weighted centroid localization algorithm based on rssi. *2014 International Conference on Wireless Communication and Sensor Network*, 276–279. <https://doi.org/10.1109/WCSN.2014.63>

- Wiktor, A., & Rock, S. (2020). Collaborative multi-robot localization in natural terrain*. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 4529–4535. <https://doi.org/10.1109/ICRA40945.2020.9197576>
- Wojcicki, P., Zientarski, T., Charytanowicz, M., & Lukasik, E. (2021). Estimation of the path-loss exponent by bayesian filtering method. *Sensors*, 21(6), 1934.
- Xianjia, Y., Qingqing, L., Queralta, J. P., Heikkonen, J., & Westerlund, T. (2021). Applications of uwb networks and positioning to autonomous robots and industrial systems. *2021 10th Mediterranean Conference on Embedded Computing (MECO)*, 1–6.
- Xu, J., Ma, M., & Law, C. L. (2015). Cooperative angle-of-arrival position localization. *Measurement*, 59, 302–313.
- Xu, N., Low, K. H., Chen, J., Lim, K. K., & Ozgul, E. (2014). Gp-localize: Persistent mobile robot localization using online sparse gaussian process observation model. *Proceedings of the AAAI Conference on Artificial Intelligence*, 28(1).
- Xu, Y., Zheng, R., Liu, M., & Zhang, S. (2021). Crmi: Confidence-rich mutual information for information-theoretic mapping. *IEEE Robotics and Automation Letters*, 6(4), 6434–6441.
- Xue, M., Sun, W., Yu, H., Tang, H., Lin, A., Zhang, X., & Zimmermann, R. (2019). Locate the mobile device by enhancing the wifi-based indoor localization model. *IEEE Internet of Things Journal*, 6(5), 8792–8803.
- Xue, W., qiu weining, w., Hua, X., & Yu, K. (2017). Improved wi-fi rssi measurement for indoor localization. *IEEE Sensors Journal*, PP, 1–1. <https://doi.org/10.1109/JSEN.2017.2660522>
- Yang, B., Guo, L., Guo, R., Zhao, M., & Zhao, T. (2020a). A novel trilateration algorithm for rssi-based indoor localization. *IEEE Sensors Journal*, 20(14), 8164–8172. <https://doi.org/10.1109/JSEN.2020.2980966>
- Yang, B., Guo, L., Guo, R., Zhao, M., & Zhao, T. (2020b). A novel trilateration algorithm for rssi-based indoor localization. *IEEE Sensors Journal*, 20(14), 8164–8172.
- Yang, B., Li, J., Shao, Z., & Zhang, H. (2022). Robust uwb indoor localization for nlos scenes via learning spatial-temporal features. *IEEE Sensors Journal*, 22(8), 7990–8000.
- Yang, Q., & Parasuraman, R. (2021). How can robots trust each other for better cooperation? a relative needs entropy based robot-robot trust assessment model. *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2656–2663.

- Yin, P., Srivatsan, R. A., Chen, Y., Li, X., Zhang, H., Xu, L., Li, L., Jia, Z., Ji, J., & He, Y. (2019). Mrs-vpr: A multi-resolution sampling based global visual place recognition method. *2019 International conference on robotics and automation (ICRA)*, 7137–7142.
- Yiu, S., Dashti, M., Claussen, H., & Perez-Cruz, F. (2017). Wireless rssi fingerprinting localization. *Signal Processing*, 131, 235–244.
- Youssefi, K. A.-R., & Rouhani, M. (2021). Swarm intelligence based robotic search in unknown maze-like environments. *Expert Systems with Applications*, 178, 114907.
- Yu, X., Wu, Y., & Sun, X.-M. (2019). A navigation scheme for a random maze using reinforcement learning with quadrotor vision. *2019 18th European Control Conference (ECC)*, 518–523.
- Yu, X., Wu, Y., Sun, X.-M., & Zhou, W. (2021). A memory-greedy policy with guaranteed convergence for accelerating reinforcement learning. *Journal of Autonomous Vehicles and Systems*, 1(1), 011005.
- Zafari, F., Gkelias, A., & Leung, K. K. (2019a). A survey of indoor localization systems and technologies. *IEEE Communications Surveys Tutorials*, 21(3), 2568–2599.
- Zafari, F., Gkelias, A., & Leung, K. K. (2019b). A survey of indoor localization systems and technologies. *IEEE Communications Surveys Tutorials*, 21(3), 2568–2599. <https://doi.org/10.1109/COMST.2019.2911558>
- Zafari, F., Papapanagiotou, I., & Hacker, T. J. (2018). A novel bayesian filtering based algorithm for rssi-based indoor localization. *2018 IEEE International Conference on Communications (ICC)*, 1–7.
- Zhang, L., Lin, Z., Wang, J., & He, B. (2020). Rapidly-exploring random trees multi-robot map exploration under optimization framework. *Robotics and Autonomous Systems*, 131, 103565.
- Zhang, Z., Yu, J., Tang, J., Xu, Y., & Wang, Y. (2022). Mr-topomap: Multi-robot exploration based on topological map in communication restricted environment. *IEEE Robotics and Automation Letters*, 7(4), 10794–10801.
- Zhang, Z., Wang, X., Zhang, Q., & Hu, T. (2022). Multi-robot cooperative pursuit via potential field-enhanced reinforcement learning. *2022 International Conference on Robotics and Automation (ICRA)*, 8808–8814.
- Zhang, Z., Henderson, T., Karaman, S., & Sze, V. (2020). Fsmi: Fast computation of shannon mutual information for information-theoretic mapping. *The International Journal of Robotics Research*, 39(9), 1155–1177.
- Zhao, M., Guo, X., Song, L., Qin, B., Shi, X., Lee, G. H., & Sun, G. (2021). A general framework for lifelong localization and mapping in changing

- environment. *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3305–3312.
- Zheng, S., Li, Z., Yin, Y., Liu, Y., Zhang, H., Zheng, P., & Zou, X. (2022). Multi-robot relative positioning and orientation system based on uwb range and graph optimization. *Measurement*, *195*, 11068.
- Zhu, D., Li, T., Ho, D., Wang, C., & Meng, M. Q.-H. (2018). Deep reinforcement learning supervised autonomous exploration in office environments. *2018 IEEE international conference on robotics and automation (ICRA)*, 7548–7555.
- Zhu, Y., Mottaghi, R., Kolve, E., Lim, J. J., Gupta, A., Fei-Fei, L., & Farhadi, A. (2017). Target-driven visual navigation in indoor scenes using deep reinforcement learning. *2017 IEEE international conference on robotics and automation (ICRA)*, 3357–3364.