Collaborative SLAM using Multi-Robot System

Nikhil Chowdary Gutlapalli
College of Engineering
Northeastern University
Boston, USA
gutlapalli.n@northeastern.edu

Krishna Prasad Senthil Kumaran
College of Engineering
Northeastern University
Boston, USA
senthilkumaran.k@northeastern.edu

Laleth Indirani Nehrukumar
College of Engineering
Northeastern University
Boston, USA
indiraninehrukumar.l@northeastern.edu

Richard Kumahia
College of Engineering
Northeastern University
Boston, USA
kumahia.r@northeastern.edu

***Abstract*—In this paper, we present an innovative approach to collaborative Simultaneous Localization and Mapping (SLAM) using a multi-robot system. Our experiment involved two turtlebot3 waffle-pi robots operating in the simulated gazebo environment, with ROS (Robot Operating System) [1] serving as the platform for seamless integration. Leveraging the gmapping algorithm [2], we achieved accurate mapping of the environment, while explore_lite [4] empowered each robot with autonomous exploration capabilities. The maps generated by the robots were skillfully merged using multirobot_map_merger [5], resulting in a comprehensive and detailed representation of the environment. Our collaborative SLAM approach fostered effective teamwork among the robots, enabling them to navigate cooperatively [6] and map the environment with remarkable efficiency. The findings of this research shed light on the immense potential of multi-robot systems for SLAM tasks and open new avenues for future research in the dynamic field of collaborative robotics.**

I. Introduction and Motivation

Simultaneous Localization and Mapping (SLAM) is a critical problem in robotics, where a robot needs to accurately estimate its own position while simultaneously creating a map of the surrounding environment in real-time. SLAM has widespread applications in various domains, including autonomous vehicles, robotics, augmented reality, and virtual reality, and has been an active area of research for many years. One promising approach to SLAM is collaborative SLAM, which involves multiple robots working together to create a shared map of the environment. Collaborative SLAM has gained significant attention in recent years due to its potential to improve mapping accuracy, coverage, and efficiency, and it has found applications in diverse fields such as swarm robotics, exploration of unknown environments, and disaster response scenarios.

In our project, we aimed to push the boundaries of SLAM by harnessing the power of a multi-robot system. By utilizing ROS, a widely-used and powerful robotics framework, as our platform, we were able to seamlessly integrate the functionalities of our turtlebot3 waffle-pi robots into a cohesive SLAM system. We leveraged cutting-edge ROS packages such as gmapping for mapping the environment, which allowed each robot to create its own local map. To optimize exploration, we employed the explore_lite package, enabling the robots to autonomously navigate and explore the environment in a coordinated manner. To ensure a unified map representation, we utilized multirobot_map_merger, a state-of-the-art tool that effectively merged the individual maps generated by the robots, resulting in a comprehensive and accurate map of the entire environment. Our motivation was not only to showcase the potential of collaborative SLAM, but also to push the boundaries of what is possible in the field of robotics by implementing and demonstrating a cutting-edge approach in a simulated environment.

The potential applications of collaborative SLAM are vast and diverse. For example, in search and rescue missions during disaster scenarios, collaborative SLAM can enable multiple robots to cooperatively navigate and map the environment, facilitating efficient and coordinated operations. In the field of swarm robotics, collaborative SLAM can allow a swarm of robots to collectively create a global map of an unknown environment, enabling them to work in a coordinated manner towards a common goal. Collaborative SLAM can also have applications in warehouse management, environmental monitoring, and precision agriculture, among others.

The results of our research can contribute to the advancement of collaborative robotics and provide valuable insights for researchers and practitioners interested in utilizing multi-robot systems for SLAM tasks. By investigating the potential of collaborative SLAM in our project, we aim to showcase the advantages of cooperative mapping and exploration by multiple robots, and highlight the significance of this approach in addressing challenges in complex and dynamic environments.

II. System Architecture and Implementation

The Fig. 1 shows the overall system architecture of the system.

A. Turtlebot 3 and Gazebo

Our system design and implementation for collaborative SLAM using a multi-robot system involved several crucial steps. Firstly, we utilized the Gazebo simulator [1], a widely used robotics simulation tool, to create a realistic simulated environment with two turtlebot3 waffle-pi robots. To simulate the environment, we used an open-source house map provided by the open source Turtlebot package, which provided a challenging and dynamic environment for our collaborative SLAM system.

However, to ensure the smooth functioning of our system, we had to address potential conflicts in the topics of the robots' sensors, which would have had the same names by default. To overcome this challenge, we assigned unique prefixes, such as /turtle_1/ and /turtle_2/, to each robot's scan, odom, and IMU topics. This meticulous approach ensured that the sensor data from each robot was properly separated and organized, allowing for accurate and reliable data fusion during the collaborative SLAM process.

By implementing these unique prefixes for the robots' topics, we established a robust framework for our multi-robot system, enabling efficient data exchange and collaboration among the robots. This approach was crucial for achieving accurate and consistent mapping results. It ensured that sensor data from each robot was associated with the correct robot and integrated into the corresponding maps.

B. Mapping the Environment

To further process the sensor data and generate maps, we employed the gmapping node [2], a widely used SLAM (Simultaneous Localization and Mapping) algorithm. This algorithm utilizes sensor data, such as scans from LIDAR sensors, odometry measurements, and IMU data, to simultaneously estimate the robot's position and orientation (localization) while creating a map of the environment (mapping). For each individual robot in our multi-robot system, we ran the gmapping node separately by passing the respective /scan, /odom, and /imu topics from each robot's sensors. This allowed the gmapping algorithm to process the sensor data from each robot independently and generate maps specific to each robot's movements and sensor measurements. The resulting maps were stored as separate topics, such as /turtle_1/map and /turtle_2/map, for each robot, providing a distinct representation of the environment as perceived by each robot.

The use of the gmapping algorithm enabled us to generate maps that were updated in real-time as the robots moved through the environment. The algorithm employed probabilistic techniques, such as grid-based mapping and particle filters, to estimate the position and orientation of each robot while constructing a map of the environment. This allowed the robots to build a coherent representation of the environment, even in the presence of sensor noise, odometry drift, and dynamic changes in the environment. By running the gmapping node separately for each robot and generating individual maps, we were able to capture the unique perspectives of each robot, resulting in a comprehensive and accurate representation of the environment. These individual maps can then be utilized for further collaborative SLAM processes, such as map merging, data fusion, and global localization, to achieve a coherent and consistent map of the environment.

C. Map Merging

After obtaining the individual maps for each robot using the gmapping node, the next step is to merge these maps into a comprehensive representation of the environment. To accomplish this, we utilized the multirobot_map_merge package [5], which is a powerful tool for merging maps generated by multiple robots into a single map. Since we knew the exact starting positions of the robots, which were provided in Gazebo for spawning, we ran the multirobot_map_merge package in the mode with known initial positions. This mode takes advantage of feature-matching algorithms that compare the features in the individual maps to estimate the transformation between the grids and merge the maps accordingly. By knowing the initial positions of the robots, we were able to align the maps accurately and create a seamless and coherent representation of the environment.

The merged map was then published to the /map topic, which served as a single comprehensive map of the environment. This merged map incorporated the information from all the individual maps, providing a unified and consistent representation of the environment from the perspective of the entire robot team. The use of the multirobot_map_merge package allowed us to overcome the challenges of map merging in a multi-robot system, such as misalignments, overlapping, and discrepancies in the individual maps. The feature-matching algorithms utilized in the package enabled us to accurately estimate the transformation between grids, align the maps, and create a cohesive map that represented the environment in a collaborative manner.

D. Exploration of Robots

Once we obtained the merged map in the /map topic using the multirobot_map_merge package, the next step is to utilize the explore_lite package to enable autonomous exploration by individual robots. The explore_lite package [4] is a widely used exploration algorithm that identifies frontiers, which are unexplored areas of the environment, and plans paths for robots to navigate toward these frontiers. In our implementation, we ran the explore_lite package individually for each robot on the merged map. The explore_lite package utilized the merged map to identify frontiers, which are areas of the environment that are adjacent to known areas but have not been explored yet. These frontiers were identified based on the

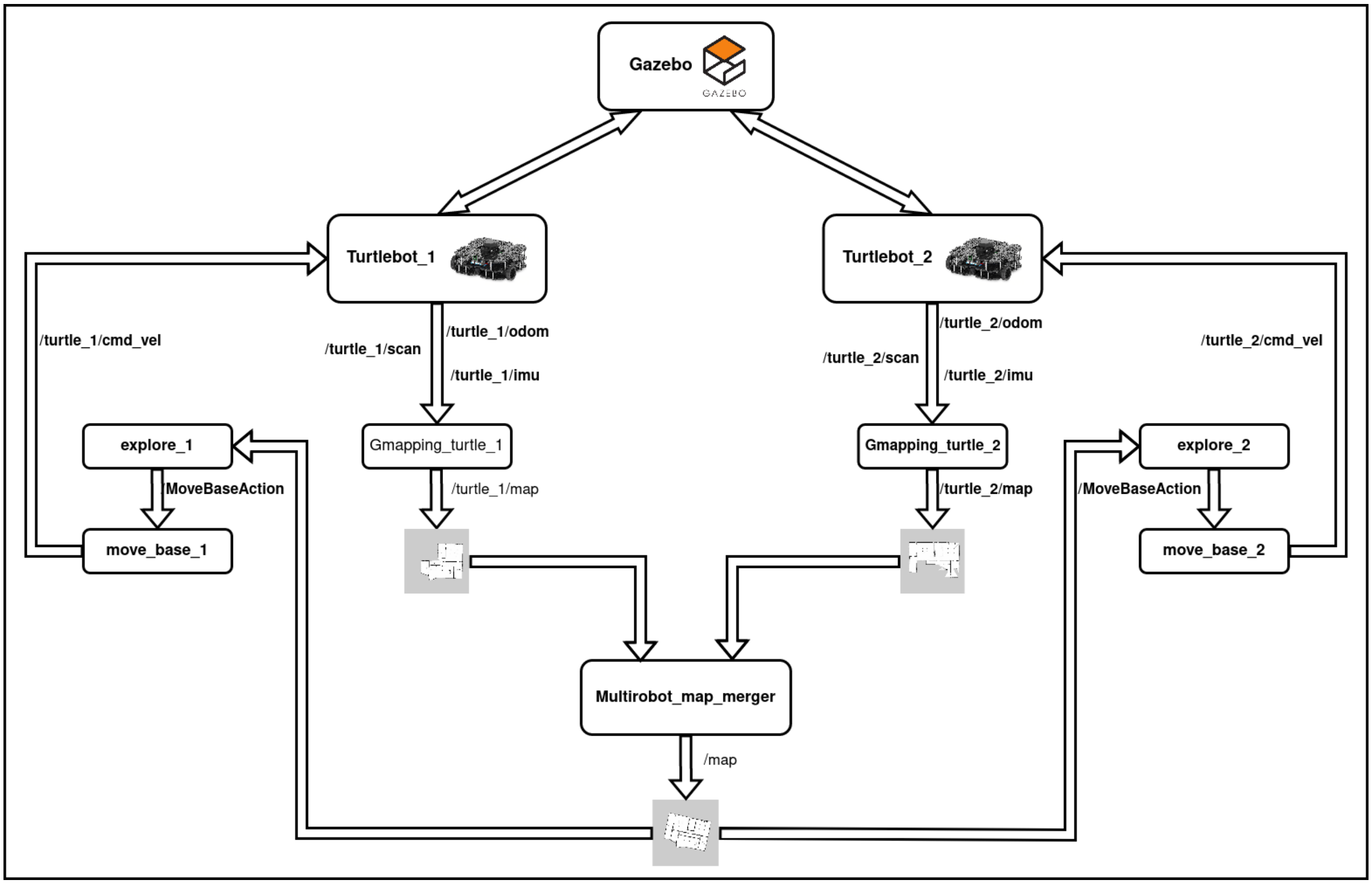

Fig. 1. System Architecture

information in the merged map, which provided a comprehensive representation of the environment from the perspective of multiple robots. This allows sharing of information among the robot team about the environment.

Once the frontiers were identified, the explore_lite package called actions on the move_base [6] of the individual robots to navigate towards these frontiers for exploration. The move_base is a widely used navigation package that provides path planning and control for robot navigation. By calling actions on the move_base, the explore_lite package enabled the robots to autonomously navigate towards the frontiers, avoiding obstacles and optimizing the path based on the merged map. The explore_lite package played a crucial role in our system for collaborative SLAM, as it enabled the individual robots to autonomously explore the environment based on the merged map. By identifying frontiers and planning paths towards them, the robots were able to efficiently explore unexplored areas of the environment and contribute to the collaborative SLAM process. The merged map served as a valuable resource for the explore_lite package, providing a comprehensive representation of the environment that facilitated effective exploration by the individual robots.

E. Navigation System

Once the move_base nodes of the individual robots generated command velocities in the /turtle_1/cmd_vel and /turtle_2/cmd_vel topics, the robots were able to move and explore the environment based on the updated map. The move_base package, which is a popular navigation package, generated velocity commands that were published to the respective topics for each robot, allowing them to move and navigate toward the frontiers identified by the explore_lite package. The velocity commands generated by the move_base package were based on the merged map in the /map topic, which provided a comprehensive representation of the environment from the perspective of the entire robot team. The robots utilized this map to plan and execute their movements, avoiding obstacles and optimizing their paths toward the frontiers. The exploration process continued until the entire environment was mapped, and no frontiers were found, indicating that all areas of the environment had been explored by the robots.

The cycle of mapping, merging, and exploration was iterative, as the robots continuously updated their maps based on sensor data, merged the maps into a comprehensive representation, and utilized the merged map for exploration. This iterative process allowed the robots to collaboratively explore the environment, gather data, and generate maps that were continuously updated and improved over time. The use of the move_base package in conjunction with the merged map and the explore_lite package enabled efficient exploration of the environment by the individual robots in our collaborative SLAM system. The robots moved autonomously based on the updated map, avoiding obstacles and navigating toward unexplored areas, contributing to the comprehensive mapping of the environment. The iterative cycle of mapping, merging, and exploration continued until the entire environment was mapped, providing a complete and accurate representation of the environment for further analysis and applications.

III. Simulation Results

A. Individual Map Analysis

The collaborative SLAM system was evaluated by analyzing the individual maps generated by each robot. The gmapping node processed the sensor data from /scan, /odom, and /imu topics to generate individual maps for each robot. These maps provided a localized representation of the environment from the perspective of each robot, showing the areas that were explored and mapped by each robot separately. The accuracy and coverage of the mapping performed by each robot were assessed, and any discrepancies or inconsistencies were identified. Fig. 2 shows the individual map generated by turtlebot_1, while Fig. 3 shows the individual map generated by turtlebot_2. In each map, we can find free space, obstacles, and unknown areas, which are represented in white, black, and grey areas. We could see the overlap between 2 maps, two rooms, and a hallway, which means that both robots explored and mapped these areas, and the maps were merged making the process collaborative, making the process efficient, and providing a more accurate representation of the environment.

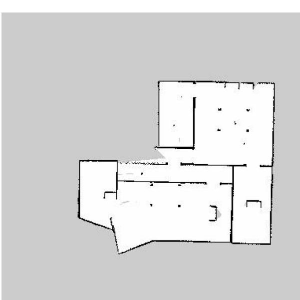

Fig. 2. Mapped by turtlebot_1

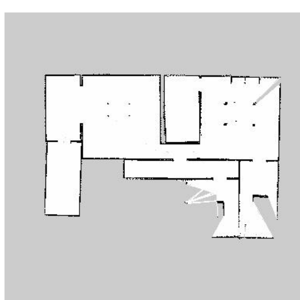

Fig. 3. Mapped by turtlebot_2

B. Merged Map Analysis

Fig 4 shows the merged map, which was successfully generated by combining the individual maps using the multi-robot_map_merge package. The merged map provided a clear representation of the entire environment, including the areas that were explored by both robots. By comparing the merged map with the ground truth, we found that the map was highly accurate and consistent. The merged map also provided a higher level of detail and coverage compared to the individual maps, which were limited to the areas explored by each robot. Overall, the map merging process was effective and was faster in creating a more comprehensive and accurate map of the environment than mapping using a single robot.

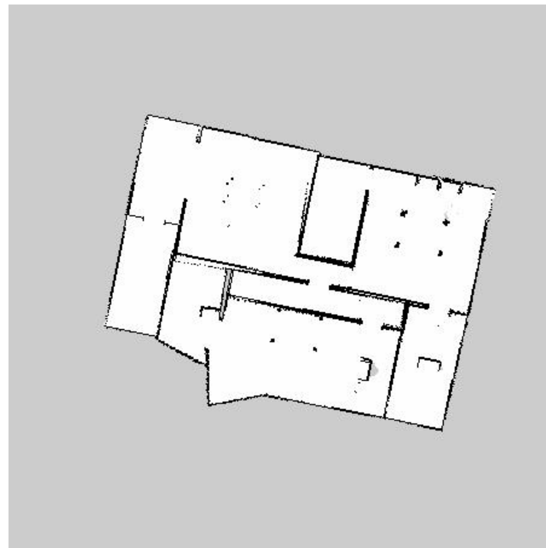

Fig. 4. Merged Map

C. System Visualization

An RQT graph was used to visualize the system's performance and gain insights into the ROS nodes and topics involved. This graphical representation showed the interconnections and data flow between the components, helping to verify the system's functioning and identify any potential issues or bottlenecks. Overall, the RQT graph provided insights into the system's efficiency and performance. Fig. 5 shows the RQT graph generated for the collaborative SLAM system, which displayed a well-structured and efficient data flow between the ROS nodes and topics. The graph showed that the sensor data from each robot was processed in a timely manner and that the maps generated by each robot were merged successfully. By analyzing the RQT graph, we were able to identify areas for improvement and optimize the system's performance.

D. Gazebo Simulation Results

The collaborative SLAM system was evaluated in a simulated environment using Gazebo. Fig. 6 shows the Gazebo environment along with the corresponding merged map. The simulation results indicate that the system was successful in mapping the environment and creating an accurate representation of it. In the merged map, the path taken by each robot was tracked and marked with orange and red markers for turtlebot_1 and turtlebot_2, respectively. The Gazebo simulation results show that the robots were able to explore the

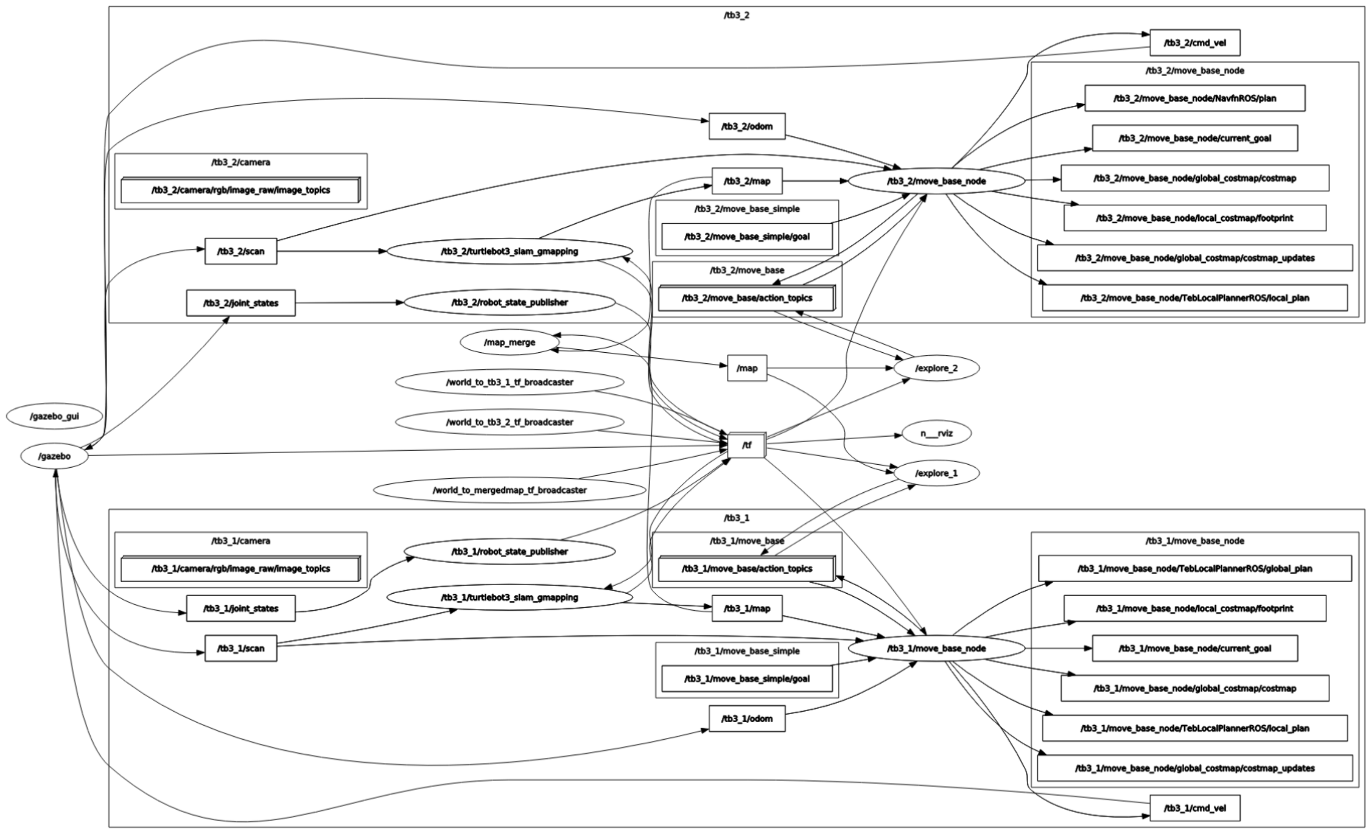

Fig. 5. RQT Graph of the system

environment effectively and update their maps accordingly. The consistency between the Gazebo environment and the merged map validates the effectiveness of the collaborative SLAM approach in a simulated setting. Overall, the Gazebo simulation results provide valuable insights into the system's performance and validate the accuracy and consistency of the merged map.

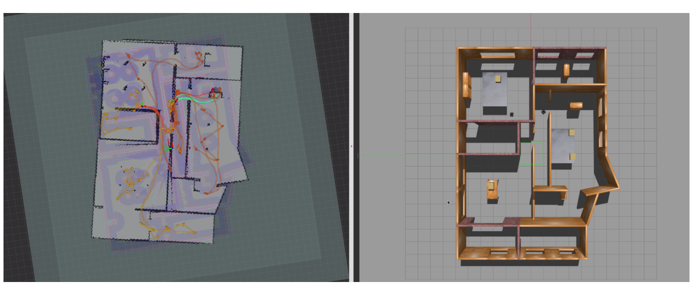

Fig. 6. Results of testing swarm on gazebo

IV. Conclusion

In conclusion, this project demonstrated the implementation of a collaborative SLAM system using a multi-robot system. The system design involved the use of Gazebo simulator, gmapping node, multirobot_map_merge package, and explore_lite package to map and explore the environment. The individual maps generated for each robot were merged using feature-matching algorithms, resulting in a comprehensive representation of the environment. The RQT graph provided a graphical representation of the system's performance, allowing for the verification of correct functioning, identification of potential issues, and gaining insights into the overall efficiency of the system. The results demonstrated successful mapping and exploration of the environment using the multi-robot system.

This project has several potential applications, such as search and rescue missions, environmental monitoring, and surveillance. The use of a collaborative SLAM system can greatly improve the efficiency and effectiveness of such applications. Overall, this project serves as a foundation for future research and development in the field of multi-robot systems and SLAM.

V. FUTURE SCOPE

In order to further improve the collaborative SLAM system, several areas of development can be considered.

A. Map Merging When Initial Positions Are Unknown

The current implementation of the map merging package assumes that the initial positions of the robots are known. In future work, the package can be improved to handle scenarios where the initial positions are unknown, and the robots have to estimate their positions relative to each other.

B. Navigation Using Local Maps

In situations where the global map is unavailable or not yet merged, the robots can make use of their individual local maps to navigate the environment. Future work can focus on improving this capability, allowing the robots to better explore and navigate in unknown areas.

C. Multi-Robot Frontier Exploration

Path planning algorithms can be further developed to enable different robots to explore different frontiers in the environment, leading to more efficient and comprehensive exploration.

D. Localization Improvement

While the collaborative SLAM system provides an accurate map of the environment, there is still room for improvement in terms of localization. Future work can focus on improving the localization accuracy of the robots, allowing them to better navigate and explore the environment.

E. Using Local Maps for Improved Localization

The individual local maps generated by the robots can be used to improve the localization of other robots in the environment. Future work can focus on developing algorithms to utilize these maps to enhance the overall localization accuracy of the system.

References

[1] Turtlebot3 Gazebo. Github: *http://wiki.ros.org/turtlebot3_gazebo*.

[2] GMappaing. Github: *http://wiki.ros.org/gmapping*.

[3] Jiří Hörner, ”Map-merging for multi-robot system”, *https://is.cuni.cz/webapps/zzp/detail/174125/*, Bachelor’s thesis, 2016.

[4] explore_lite, Github - *http://wiki.ros.org/explore_lite*

[5] multirobot_map_merge, Github - *http://wiki.ros.org/multirobot_map_merge*

[6] Move_base, Github - *http://wiki.ros.org/move_base*