Technical issues of MRL Virtual Robots Team RoboCup 2019, Sydney – Australia

Mohammad H. Shayesteh[1], Mohammad M. Raeisi[1]

[1] Islamic Azad University of Qazvin, Electrical, IT & Computer Sciences Department
, Mechatronics Research Laboratory,
Qazvin, Iran

{m.h.shayesteh, mahdrsi}@gmail.com

Abstract. This paper presents technical issues of MRL team preparation to participate in Virtual-Robot league in RoboCup competitions Sydney-Australia 2019. Due to new challenges on this league, a new software has designed for controlling four-wheeled and aerial robots in unknown environment manually and autonomous based on ROS Framework, after that some required modules like SLAM[1], navigation, exploration systems is developed to search in disaster areas without human intervention.

1 Introduction

In the virtual robot, a disaster environment is simulated which could be explored with a team of rescue robots. It based on a simulator and presented under an open source package, a high fidelity simulator on the Gazebo simulator [2]. Within this simulator research teams can setup multiple agents whose capabilities closely mirror those of real robots. This simulator currently features wheeled as well as some sensors and actuators. Moreover, teams can easily develop models of new robotic platforms, sensors and test environments.

MRL Virtual Robot have participated since 2006 in various RoboCup completions such as IranOpen, Kharazmi, WorldCup and Asia Pacific. Our major focus is on developing four-wheels and areal robots bases. We have been champion on 2013 and 2014 WorldCup competitions and best in class of tools development in Robocup Asia Pacific 2018.

This team consist of M.Sc. and BC.s students in different fields such as Artificial Intelligent, Software Engineering and Information Technology Engineering. Most of mentioned researches area are defined as thesis's topics. Mechatronic Research Laboratory is depend on Islamic Azad University of Qazvin.

[1] Simultaneous Localization and Mapping

2 Team Members

The team members and their contributions are as follows:

- Mohammad H. Shayesteh: GUI[2], SLAM[3], Map Merge
- Mohammad M. Raeisi: Navigation, Autonomous Exploration

3 GUI

To control as set of wheeled and areal robots in virtual robot environments, a flexible software needs to be design for managing all the robots concurrently to gathering their information in a centralized system and using them on multi-robot exploration and or multi robot mapping systems. Due to these changes, we presents a new software as a dependable program for use all ROS capabilities in our platform.
This new software has designed with C++ language programming and QT platform for multi-robot driving application. The software is consist of multi-robot mapping, multi-robot control, multi robot exploration and a new camera visualization system. In Addition, our new software won best in class of tools development award in Robocup Asia Pacific 2018.
In this TDP we show the most important abilities and sections of our software follows as:

Setup Environment:
This section provides a wizard form to spawn how many robots for each rounds and you can configure robot name, initial positions or robot topics with just one click setup.

Multi-Robot Control:
A software with a suitable dashboard to control many robots and switch between manual or autonomous mode simply. In this section, users can select how many robot to be spawn by setup form and considers a dynamic view based on system configuration.

Visualizer:
A dock panel in main software for visualize the robot trajectories, explored maps and marking the victims. Explored maps is prepared from our map merged package and operator can follow real time map in competition rounds.

Camera Viewer:
A new widget designed that shows RGB and Thermal cameras in multi-window. Operator can easily monitor all of robot cameras in a single form.

[2] **G**raphical **U**ser **I**nterface

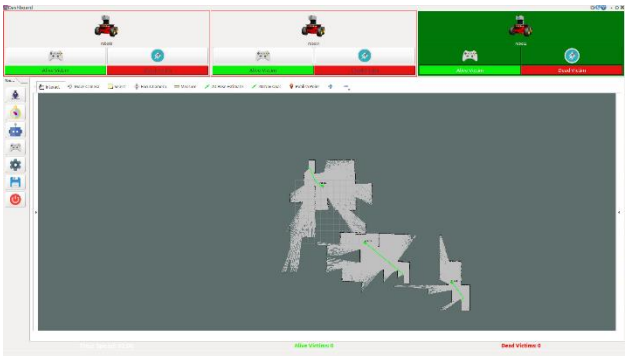

Fig 1. Multi-robots control dashboard and dock visualizer

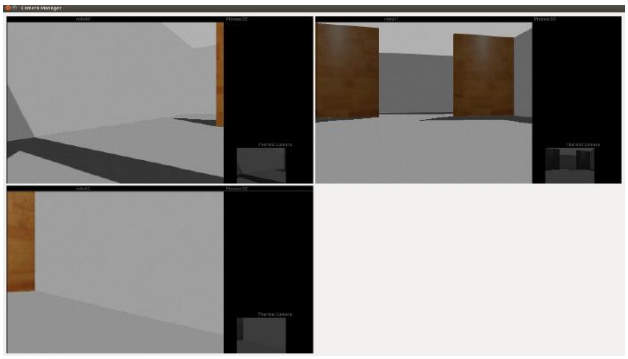

Fig 2. Multi-robots camera monitoring window

As shown in Fig 1, you can do some necessary tasks like opening cameras window, switching all robots to manual drive or autonomous exploration and saving explored map from left panel. In top panel you change robot status between manual and autonomous, mark dead/live victims on map separately and also explored map with robot trajectories is shown in button dock panel.
As shown in Fig 2, all robot RGB/Thermal cameras shows vertically/horizontally next to each other and operator can easily monitor all robot windows in a single form.

4 System Overview

Software in ROS framework is organize in packages. A package might contain ROS nodes, a ROS-independent library, configuration files, a third-party piece of software, or anything else that logically constitutes a useful module. The goal of these packages it to provide this useful functionality in an easy-to-consume manner so that software can be easily reused.

In our new system, each challenges such as SLAM, Navigation, Exploration, multi-robot control is separated from each other by standard ROS packages, and it customized and enhanced many ROS package based on our requirements.

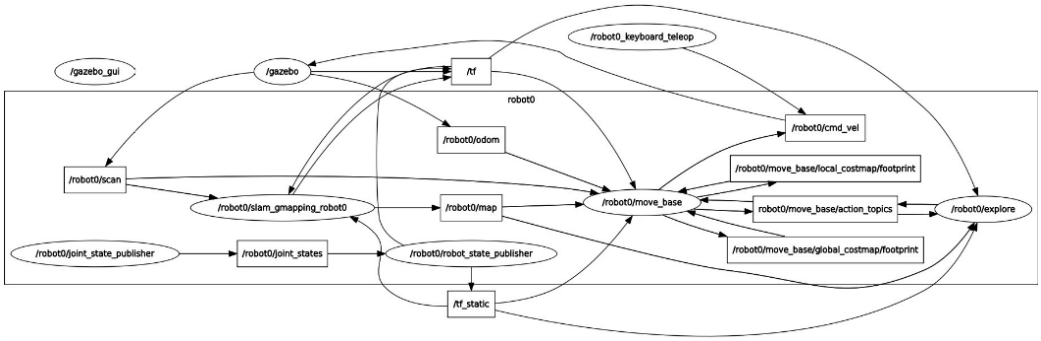

Fig 3. Robot node with its SLAM, Navigation and Exploration stacks

As shown in Fig 3, each robots acts as a node with their SLAM, Navigation and Exploration module and they share its information via multi-map module and centralized user interface application.

5 SLAM & Navigation

Scan matching as a basic part of SLAM has a key role in localization and even Mapping of mobile robots. In our previous researches, we implemented ICEG [3] as Scan matching method and Grid Mapping in previous competitions.
This year for SLAM and Navigation challenge, we uses ROS Packages. They are standard in implementation and have a good performance in real time rounds. For mapping, The gmapping package[4] provides laser-based SLAM (Simultaneous Localization and Mapping), as a ROS node called slam_gmapping and the algorithm is used with a specific packages enhanced for our team, which using a 2D Hokoyou type laser scanner which embedded on all of our Pioneer-3at Robots.
In addition, each robot have a local slam node and they use this service in many others sections like navigation and exploration. Finally, we merge all of local maps for every robot with multirobot_map_merge package [5]. This package is customized based on our requirements and able publish global positions of robots and also their trajectories.

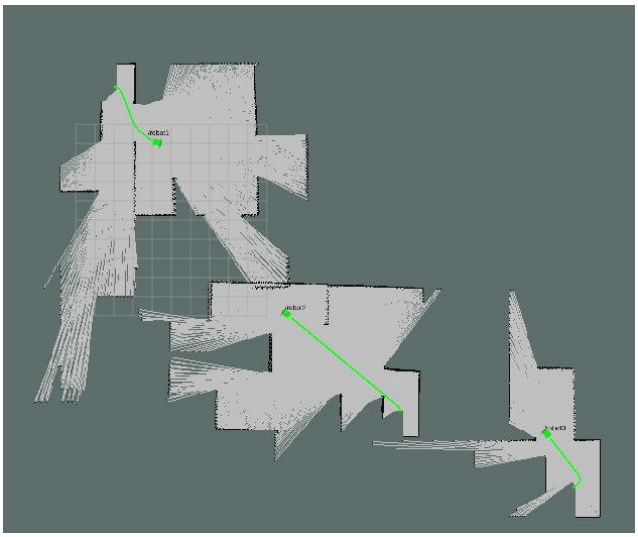

Fig 4. Merged maps of spawned robot

As shown in Fig 4, each robot sends its local map to map merge package, after that all local maps merge together then publishes by a specific topic for visualization and another purpose. In addition, robot trajectories and positions is calculated in our customized map merge package and drawing in visualizer panel.

For more precise explain core of our 2D-Navigation System, It is based on base_local planner which uses TEB local planner [6, 7, 8, 9, 10] algorithm that provides clear path from published maps and optimizes the robot's trajectory with respect to trajectory execution time, separation from obstacles and compliance with kino-dynamic constraints at runtime.

6 Autonomous Exploration

One of the main purposes in virtual robot league is autonomous exploration. Each robot should automatically explore the unknown areas based on definite rules.

For this goal, we are using explore_lite [11] package and this package provides greedy frontier-based exploration. When node is running, robot will greedily explore its environment until no frontiers could be found. Movement commands will be send to Navigation section.

7 Quadrotor Robot

There is a new type of rescue robots that fly on unknown environments to find victims in less possible time. This year we are using this new robot and they are able to search the outdoor maps and mark victims.

For this purpose, we use hector_quadrator [13] package that has been developed by the Team Hector Darmstadt of Technische Universität Darmstadt.

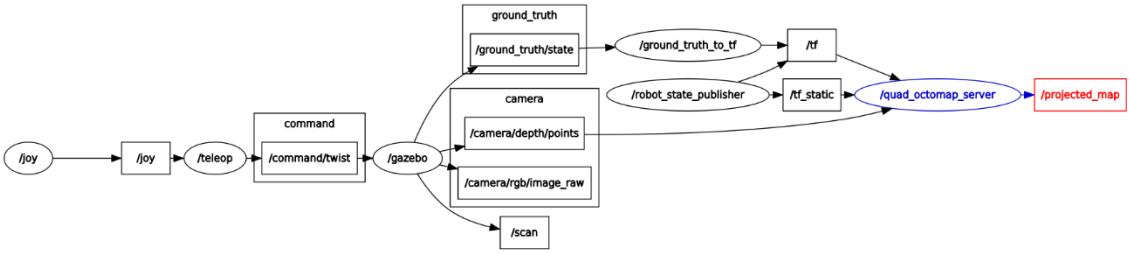

Fig 5. Quadrotor robot node with its related stacks

As shown in Fig 5, Our Quadrotor Robot model consists, a joy stack and laser scanner sensor to control robot manually via operators and scan its around environment, send visited data with scan topic to SLAM module. Finally, generated 2D map will be merged with other robots map. Mapping process is doing quickly and uses by multi-robot exploration system.

8 Conclusion

In this paper we are designed our new software based on ROS framework which are needed for autonomous systems. On the other hand, we presents a new ROS stack structure for controlling a set of robots in disaster environments autonomously based on the reliable SLAM, Navigation and Exploration issues. Some new packages such as multi-robot map merge systems customized due to our requirements. Finally, we use quadrotor robots to search unknown areas quickly and find the victims easier.

9 Future Works

Based on our research in this area, there is a stable package designed by Hector Team [12] that consists, SLAM, Navigation, and Multi-Robot Exploration systems with a good performance in mentioned challenges. We tend to immigrate on this system with a new attitude and customize them with our requirements. I addition, two new challenges has announced by technical members of virtual robot league for multi-floor map

and communication cut-off systems between the robots recently. Due to these new challenges, this team is trying to design some new modules for commutation management between robots and generate multi-floor maps.

References

1. Quigley M., Gerkey B., Conley K., Faust J., Foote T., Leibs J., Berger E., Wheeler R., Andrew N. : "ROS: an open-source Robot Operating System", "ICRA Workshop on Open Source Software"
2. https://github.com/reyhanehpahlevan/RoboCup2018RVRL_Demo
3. Taleghani, S., Sharbafi, M. A., Esmaeili, E., Haghighat, A. T., 2010, ICE matching, a robust mobile robot localization with application to SLAM. IEEE 22nd International Conference on Tools with Artificial Intelligence (ICTAI), Arras, France, Oct 27–29.
4. http://wiki.ros.org/gmapping
5. http://wiki.ros.org/multirobot_map_merge
6. C. Rösmann, F. Hoffmann and T. Bertram: Integrated online trajectory planning and optimization in distinctive topologies, Robotics and Autonomous Systems, Vol. 88, 2017, pp. 142–153.
7. C. Rösmann, W. Feiten, T. Wösch, F. Hoffmann and T. Bertram: Trajectory modification considering dynamic constraints of autonomous robots. Proc. 7th German Conference on Robotics, Germany, Munich, May 2012, pp 74–79.
8. C. Rösmann, W. Feiten, T. Wösch, F. Hoffmann and T. Bertram: Efficient trajectory optimization using a sparse model. Proc. IEEE European Conference on Mobile Robots, Spain, Barcelona, Sept. 2013, pp. 138–143.
9. C. Rösmann, F. Hoffmann and T. Bertram: Planning of Multiple Robot Trajectories in Distinctive Topologies, Proc. IEEE European Conference on Mobile Robots, UK, Lincoln, Sept. 2015.
10. C. Rösmann, F. Hoffmann and T. Bertram: Kinodynamic Trajectory Optimization and Control for Car-Like Robots, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, Sept. 2017.
11. https://github.com/hrnr/m-explore
12. http://www.teamhector.de/
13. http://wiki.ros.org/hector_quadrotor