Journal of Automation and Control, 2018, Vol. 6, No. 1, 15-18
Available online at http://pubs.sciepub.com/automation/6/1/2

DOI:10.12691/automation-6-1-2

Experimental comparison of the Simultaneous Localization and Mapping Schemes in the Practical Environments Using the Robot Operating System

Ju-Min Kim, Ha Jeong Seok, Chang-bae Moon*

Department of Mechanical Engineering, Chonnam National University, Gwanju, Korea
*Corresponding author: cbmoon@jnu.ac.kr

Abstract In recent days, mobile service robots have been widely developed for security, cleaning and guidance purpose. The use of open source code based on Robot Operating System (ROS) supports rapid testing and comparison of the previous developed schemes. Especially, the SLAM (Simultaneous Localization and Mapping) schemes are most useful because the SLAM schemes are requiring the previous recorded logging data from sensors, *e.g.*, odometry and laser scanners. In this paper, we represent the simulation method using ROS and being supported SLAM schemes by ROS. The simulations carried out by using the previous recorded odometry and laser scanning data. The experimental results are presented by using the Cartographer and KARTO SLAM schemes.

Keywords: *mobile robot, simultaneous localization and mapping, Robot Operating System (ROS)*

Cite This Article: Ju-Min Kim, Ha Jeong Seok, and Chang-bae Moon, "Experimental comparison of the Simultaneous Localization and Mapping Schemes in the Practical Environments Using the Robot Operating System." *Journal of Automation and Control*, vol. 6, no. 1 (2018): 15-18. doi: 10.12691/automation-6-1-2.

1. Introduction

Researches [1,2,3] on robot with ROS(Robot Operating System) have been wide spread recently. As ROS has provided the standard of robot software platform, it makes researches on robot accelerated and becomes the common platform for global cooperation with researchers on robot.

The main function of mobile robot consists of position-prediction and dynamic control. The position-prediction process is the comparison between data from sensors equipped and map information made previously, and the prediction for the position of robot in real time.

Path planning and dynamic control which is based on the position-data from position-prediction technology to reach the goal position from initial position is working on. There is the part of dynamic control, DWA (Dynamic Window Approach)[4], in ROS. DWA allows a robot to detect unpredicted obstacles dynamically in the range of detection of sensors and to avoid them, then it makes the robot be going to a new path as setting up a path planning generally.

The ROS supports motion control [4,5], localization [6,7], SLAM(Simultaneous Localization and Mapping), various sensors and H/W Driver, *etc*. SLAM and Localization technology have a big advantage to reuse the data which is saved previously.

Gmapping [8] using Particle filter and karto SLAM [9] based on the Graph-SLAM, cartographer SLAM package in ROS package[10] are possible to be used. This study shows qualitative analysis of the result for mapping with karto/cartographer SLAM based on graph SLAM. ROS package needed and transform filter are setup for this purpose.

The remaining sections of this paper are organized as follows. In Section 2, ROS package and experiments are explained. The qualitative analysis and experimental results are explained in Section 3. Finally, the concluding remarks are presented in Section 4.

2. ROS System Configuration

2.1. ROS System Configuration

Simple and minimum ROS package is selected for qualitative analysis of the test on this study. The minimum Selection is as the below.

- Transform publisher: Management for matrix of the coordinate transformation in the robot position
- Odometry/Scan publisher: Laser scanning and odometry data of the robot

2.2. Environmental Setup for Collecting Data

The transform publisher is shown in Figure 1. The transformation of coordinate system is based on the change from the coordinate on the map (/map) which is the global coordinate system to the coordinate which is oriented to rotation of the robot (/base). Then the transformation of coordinates from /map to /base_link is operating on each module of ROS as SLAM module is working on. The positions of laser scanner and wheels are converted into the coordinate system based on rotation of the robot(/base_link).

Odometry/Scan publisher are separated and operated each other in a different processor generally. However, it is more effective to be operating simultaneously in one processor as considering the time-synchronization with old data accumulated previously. So the data from odometry and laser scanner is managed in the uni-log file (same log file). The odometry is the predicting the position change with time by using variables such as measuring data of wheel rotation and angle.

The data collected from odometry and laser scanner is saved in storage of another PC. The laser scanner LMS100 is used and data is saved with the unit of 25Hz and 0.5º and the odometry saves data with the unit of 10Hz. Then SLAM is operated by using files saving data in a format of time/Odometry/Laser scan converted into the standard of odometry time.

3. Experimental Results

The Real environment in width, 14.5m and length, 67.9m for simulation is shown in Figure 2(a). Reference map of Environment is shown in Figure 2(c) and the test is performed in the same environment. Mobile robot for indoor equipped with SICK laser scanner LMS-100 in the test is shown in Figure 2(b) and the size is 66cm in width and 80cm in length same as the size and the scanner type of the indoor mobile robot in simulation.

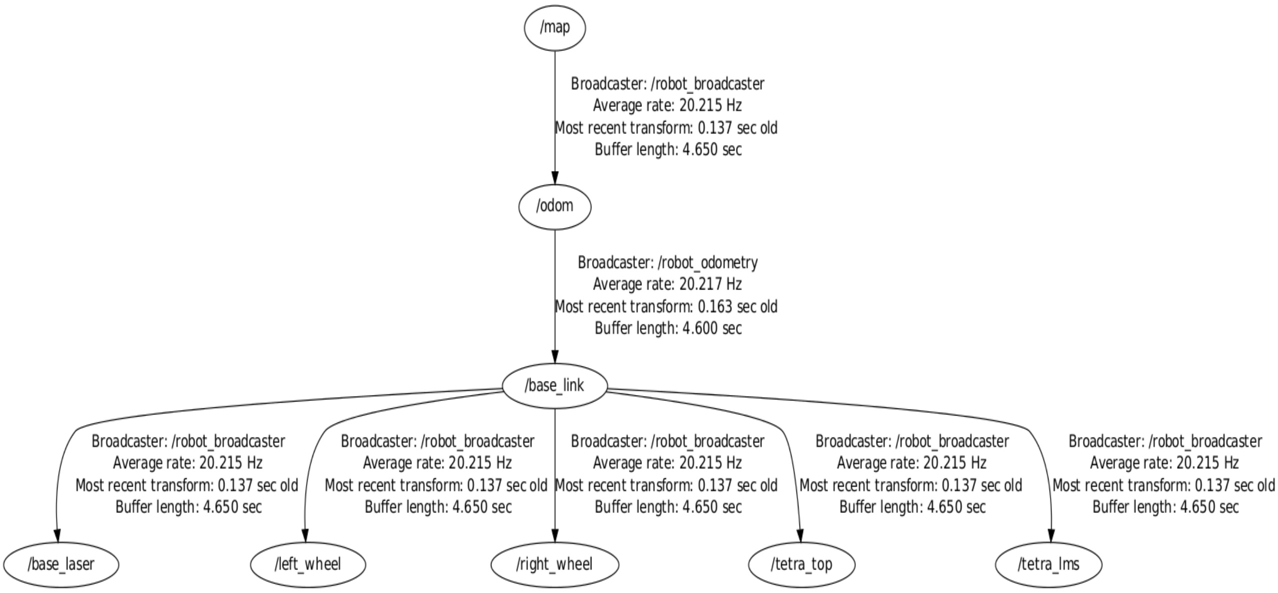


Figure 1. ROS Transform Tree

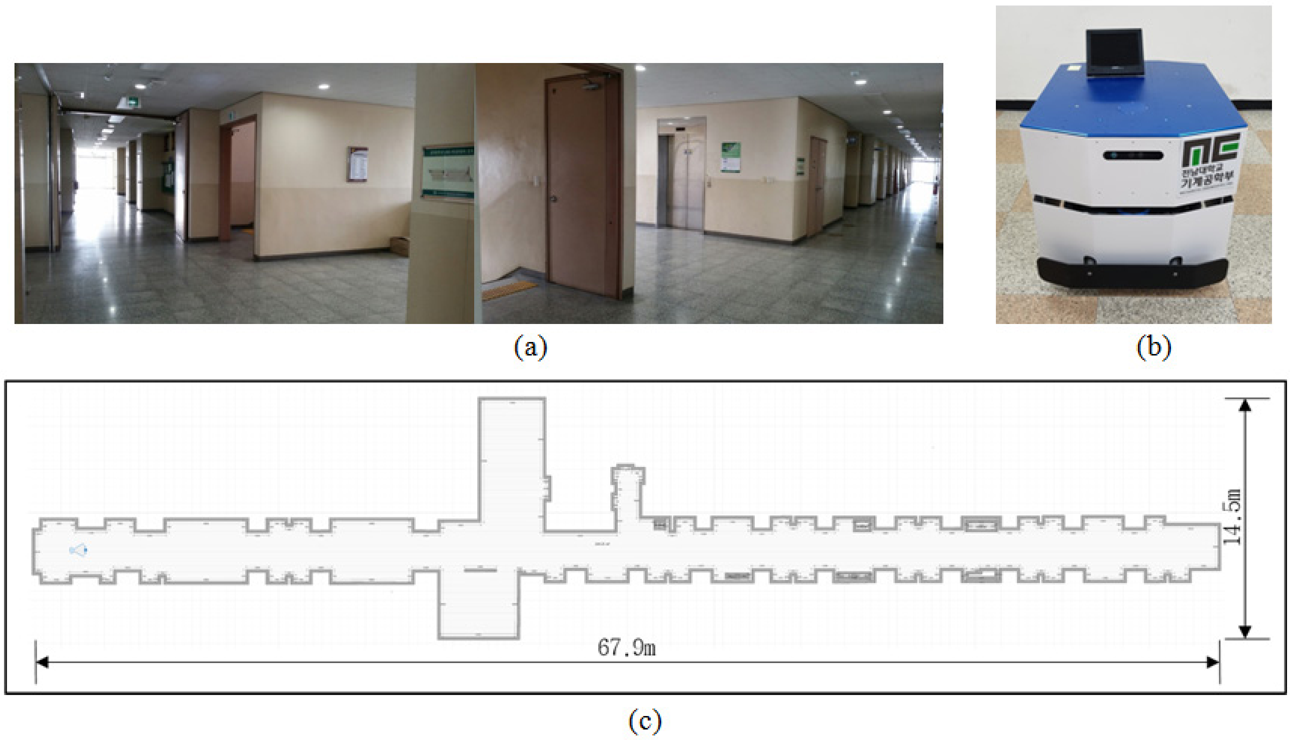
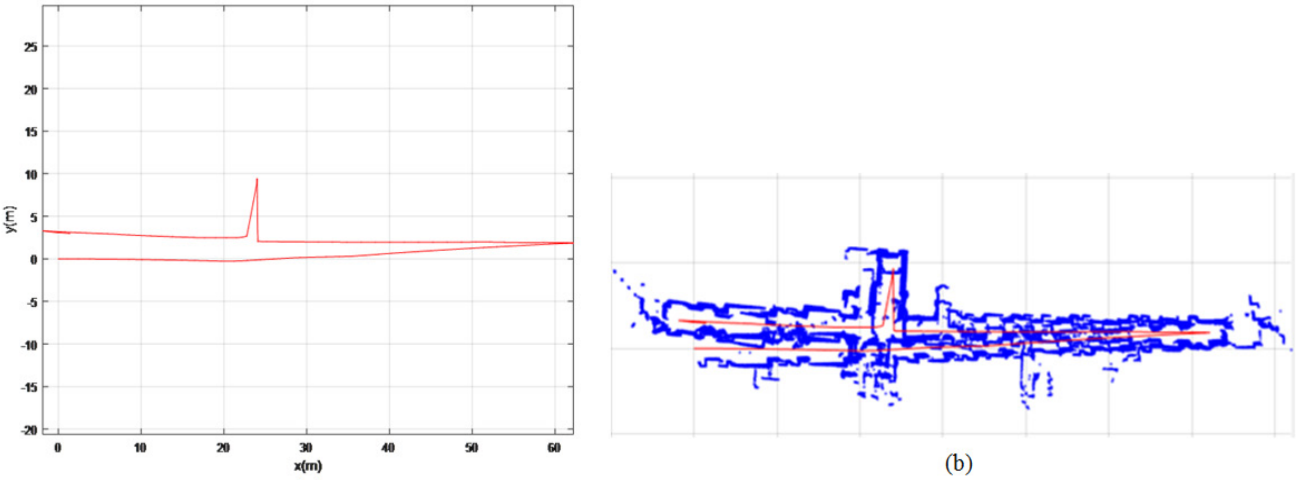


Figure 2. Simulation environment (a) A real target environment (left and right corridor), (b) SCV mobile robot, (c) Reference map of the environment

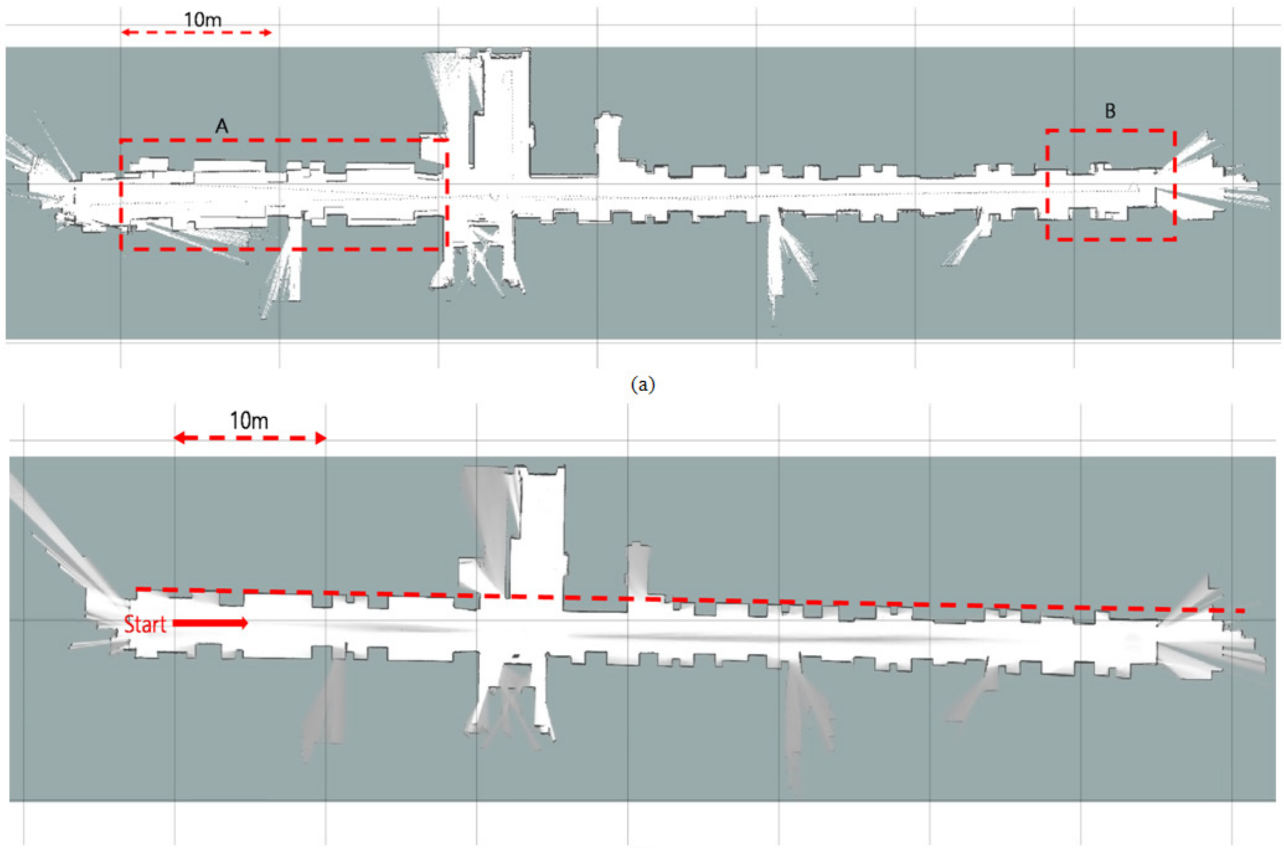
All points from odometry data accumulated for mapping have been measured by the laser scanner in the odometry coordinate in Figure 3. The starting point is set the origin, (0,0) and the graph shows the route going through the turning point and returning to the origin. And the error in odometry accumulated by deviation increases at the end point (the starting point). Therefore, the data from laser scanner is either including the error from odometry as being shown in Figure 3(b).

The results are shown about Cartographer SLAM and KARTO SLAM in Figure 4. Map matching result with loop closing of Cartographer SLAM in real time is showing the better precision than the result with KARTO SLAM. The loop closing of Cartographer SLAM is a technique for reduction of error generated in mapping the nodes on the paths which the robot has recognized the same route as it has been passing through. The deviation by 1m is shown with KARTO SLAM in section A in Figure 4 a). Loop closing process of KARTO SLAM is showing in Figure 5 and the result of the process enhances the performance and reduce the error caused by the rotation of the robot in the turning point in section B shown in Figure 4(a).



(a) (b)

Figure 3. Odometry Result (a) odometry path, (b) laser scan (blue dots)



(a)
(b)

Figure 4. SLAM Result (a)Cartographer, (b)KARTO

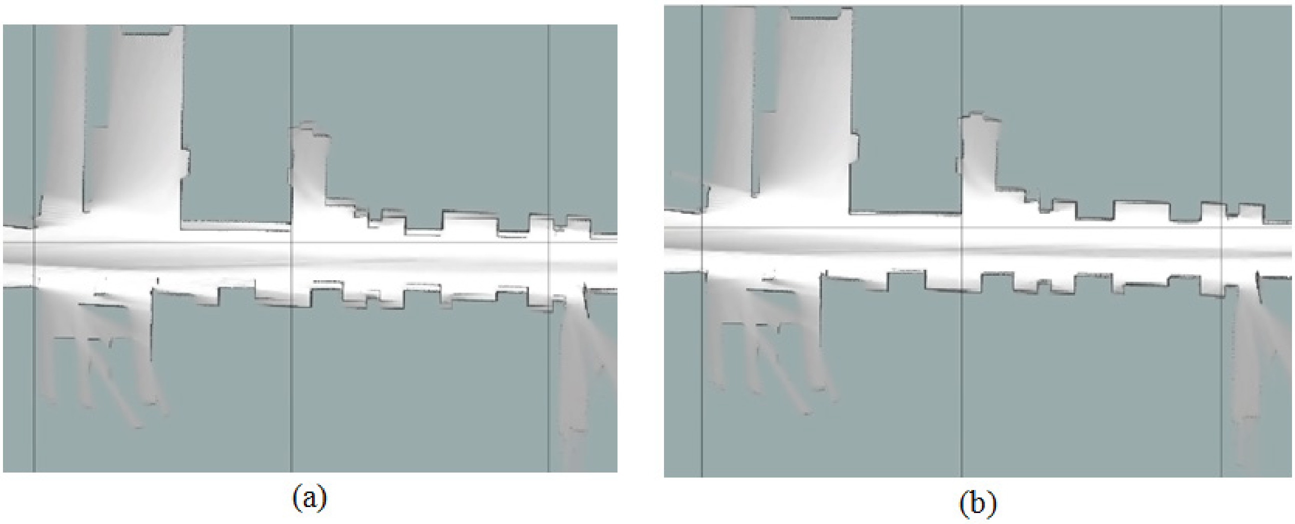


Figure 5. SLAM Result using Cartographer (a) before loop closing, (b) after loop closing

4. Conclusion

This study shows the comparison results between Cartographer SLAM and KARTO SLAM with ROS, qualitatively. The results which are shown especially confirm that the comparison can be possible by using message service in ROS without any additional program.

Acknowledgements

This work was supported by Korea Institute of Planning and Evaluation for Technology in Food, Agriculture, Forestry and Fisheries (IPET) through Advanced Production Technology Development Program, funded by Ministry of Agriculture, Food and Rural Affairs (MAFRA) (116058-03-3-HD020)

References

[1] Hardware Robot Operating system (H-ROS)., 2016, http://www.ros.org/news/2016/10/hardware-robot-operating-system-h-ros.html (accessed online: 2016-11-07).

[2] ROS Interface for Impedance/Force Control., 2016, http://www.ros.org/news/2016/09/ros-interface-for-impedanceforce-control.html (accessed online: 2016-11-07).

[3] UpDroid announces the UP1, http://www.ros.org/news/2016/01/updroid-announces-the-up1.html (accessed online: 2016-11-07).

[4] dwa_local_planner, http://wiki.ros.org/dwa_local_planner (accessed online: 2016-11-07).

[5] D. Fox, W. Burgard, and S. Thrun., 1997, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, Vol. 4, No. 1

[6] amcl, http://wiki.ros.org/amcl (accessed online: 2016-11-07).

[7] Dieter Fox, Wolfram Burgard, Frank Dellaert, and Sebastian Thrun, 1999, "Monte Carlo Localization: Efficient Position Estimation for Mobile Robot," Proc. of the sixteenth National Conference on Artificial Intelligence John Wiley & Sons Ltd.

[8] gmapping ros package link, http://wiki.ros.org/gmapping (accessed online: 2016-11-07).

[9] slam_karto ros package link, http://wiki.ros.org/slam_karto (accessed online: 2016-11-07).

[10] cartographer ros package link, https://github.com/googlecartographer (accessed online: 2016-11-07).