Human-Robot Interaction Using ROS Framework for Indoor Mapping Missions

M.S. Hendriyawan Achmad[1], Mohd Razali Daud[2], Saifudin Razali[3], Dwi Pebrianti[4]

[1]Department of Electrical Engineering, Universitas Teknologi Yogyakarta, Yogyakarta, Indonesia
[1-4]Robotics and Unmanned Research Group (RUS), Instrument & Control Engineering (ICE) Cluster,
Faculty of Electrical and Electronics Engineering, Universiti Malaysia Pahang, 26600, Pekan, Pahang, Malaysia

[1]hendriyawanachmad@uty.ac.id, [2]mrazali@ump.edu.my, [3]saifudin@ump.edu.my, [4]dwipebrianti@ump.edu.my

Abstract –The necessity of effective integration tools in robotics research play an important scientific contribution and highly recommended. However, robotics researchers have limited time during the experiments with definite targets and ultimately there is no choice but to use integration tools as software framework to avoid re-inventing the wheels. This research examines how humans work with teleoperated unmanned mobile robots to perform interaction in order to explore and build an indoor map by utilizing robotic software framework known as Robot Operating System (ROS). ROS infrastructure tools are involving together from the file system level to the community level, enables independent decisions about development and implementation. These experiments focus on two major area; the way human delivers the targets coordinate to the robot using ROS framework, and the way robot conducts SLAM as a feedback to the human related to current location and occupied map within ROS platform. Hector SLAM plays an important role in 2D localization using 2D LIDAR sensor which is needed by Octomap in order to construct 3D mapping simultaneously using Kinect sensor. The results showed that human-robot interaction using the ROS-based teleoperated system for mapping task is easy to configure and propose an effective tool in term of large-scale service robot development.

Keywords – ROS; Kinect; 2D LIDAR; Hector SLAM; Octomap; Human-Robot Interaction

1. INTRODUCTION

The robotics technology has developed rapidly and applied to many fields, such as industry, education, health, household, entertainment, and the military [1]. An interaction between humans and robots in completing the task is one of the benefits of robots to make people's lives better. In some cases, robots will share the same workspace and work closely with humans to accomplish collaboration tasks as part of their day-to-day work [2]. The aim of Human-Robot interaction research is to determine models of interaction in term of hardware & software development to obtain effective collaboration between humans as masters and robots as slaves [3]. Robots with many sensors used for exploration and mapping tasks without software framework will get a higher level of complexity compared to utilizing of software framework. ROS was developed to address a set of big challenges of complexity being faced when developing large-scale of service robots [4]. There are many studies discussing ROS-related to humans – robots interaction to accomplish collaboration tasks [5].

The Simultaneous Localization and Mapping (SLAM) technique is a common issue in mobile robots discussion. The basic idea of SLAM is to use the robot for explorations mission at an unknown location and construct the map base on obstacles surrounding it. In reality, the non-ideal behavior of sensors and actuators produce very complex problem and need to be addressed [6]. Researchers have been studying localization and mapping techniques by using many types of sensor devices, such as sonar depth sensors [7], visual sensors [8], and laser depth scanners [9]. Nevertheless, the techniques tend to devices-oriented due to the differences of work of the sensors and the characteristics of each device as well [6]. The idea to puts SLAM technique on the mobile robot equipped with multiple sensors is a challenge due to software integration complexity as has been done in [10].

This paper proposed the methods how to implement the ROS framework to accommodate human-robot interaction in order to carry out indoor exploration and mapping tasks (SLAM) utilizing the differential mobile robot. For localization function, we used Hector SLAM algorithm. And for 3D mapping, we used Octomap algorithm. Both of techniques are open-source and widely used together with ROS.

2. SYSTEMS DESCRIPTION

A. Components Configuration

The main components of the system are a mobile robot, remote station, and network provider. The remote station is a computer that runs ROS-master to facilitate human during interaction with the remote mobile robot. Since ROS is designed with distributed computing, single ROS-master is able to serve multiple ROS-slaves in the same network. Network provider takes in charge to ensure network availability during runtime, fast network and low latency are indispensable. The mobile robot is equipped with a computer which runs ROS-slave to accommodate sensors reading (laser and Kinect) and robot base controlling. PC ROS-slave contains three major ROS package, such as Arduino package for base controller interface handler, Hector package for 2D localization using laser sensor, and Octomap package based on Octree method for 3D mapping using Kinect sensor incorporates with Hector 2D localization which is obtained. The two-way interaction between human-robot with teleoperated mode are human set the goal position based on obtained map from a remote area and robot send the occupied map back to the human as an exploration result. Fig. 1 illustrates overall components configuration of the system based on robot operating system (ROS) framework which was used in the experiment.

B. Non-holonomic Differential Mobile Robot

The type of robot used in this experiment is a non-holonomic differential mobile robot. Right wheel and the left wheel has an independent driver of each. For each driver is handled by a PID controller to maintains a speed of wheel which is desired by PC ROS-slave. An integrated mobile robot in this experiment is equipped with some type of sensors that is LIDAR sensor: Hokuyo URG-UTM30LX, RGB-D sensor: Kinect, odometry sensor: wheels encoder.

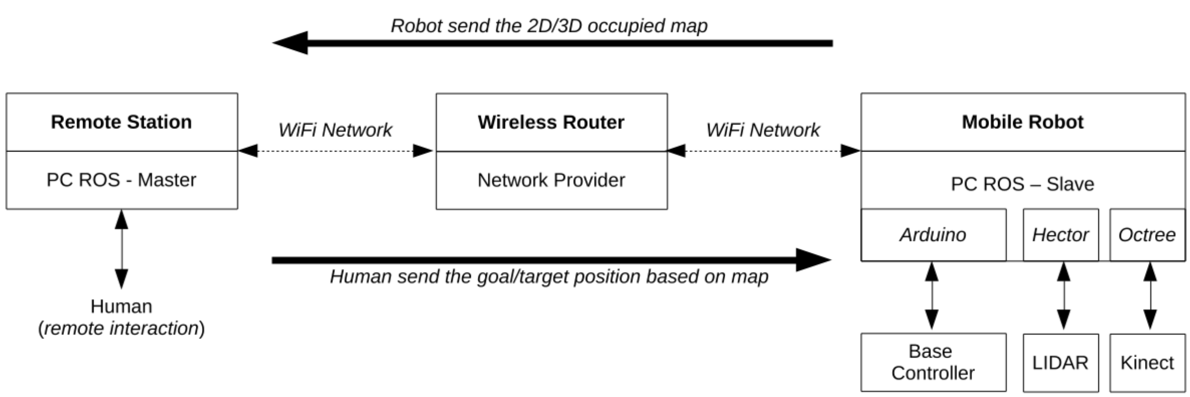

Figure 1. System configuration based on ROS framework

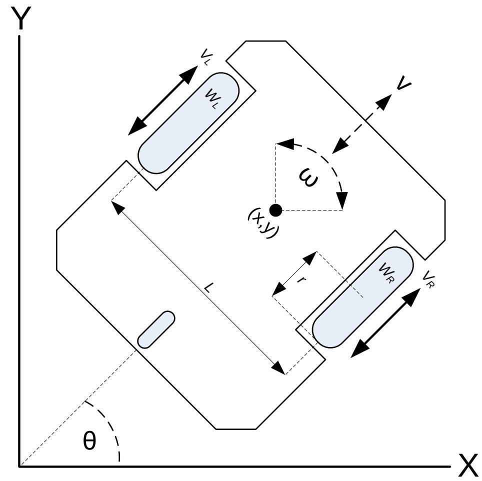

Figure 2. Kinematics model of non-holonomic differential mobile robot

Based on kinematics model of a mobile robot as seen in Fig. 2, and then we can calculate the desired velocity for both of right wheel (V_R) and left wheel (V_L), and estimate both of relative position ($\dot{x}, \dot{y}$) and relative heading $\left(\dot{\theta}\right)$ of the robot as well using (1-5).

$$V_R = \frac{\left(v - \frac{L*\omega}{2}\right)}{r} \quad (1)$$

$$V_L = \frac{\left(v + \frac{L*\omega}{2}\right)}{r} \quad (2)$$

$$\dot{x} = \frac{r}{2}(V_R + V_L)\cos\theta \quad (3)$$

$$\dot{y} = \frac{r}{2}(V_R + V_L)\sin\theta \quad (4)$$

$$\dot{\theta} = \frac{r}{L}(V_R - V_L) \quad (5)$$

where v is targeted linear velocity and ω is targeted angular velocity of mobile robot, L is distance between left and right wheel (wheelbase), and r is radius of wheel [11].

Based on Fig. 3, PC ROS-slave receives messages from Kinect and LIDAR and transmits a message to the base controller. Kinect sensor produces RGB-D image data, laser sensor produces 2D depth scanning data, and base controller receives two types of input velocity, 3-axis linear velocity $\left(v_x, v_y, v_z\right)$ and 3-axis angular velocity $\left(\omega_x, \omega_y, \omega_z\right)$. Base controller receive velocity messages were interpreted in two correspondence values of speed (left wheel & right wheel) based on robot kinematics model as seen in (1)(2). Set points for two PID controllers are Vleft_wheel and Vright_wheel which correspondence with targeted linear velocity and angular velocity given by PC ROS-slave. Afterward, PID controllers maintain linear velocity and angular velocity of the mobile robot toward the target position. Fig. 4 shows the real mobile robot which is completely designed based on model and component configuration as seen in Fig. 2 and Fig. 3.

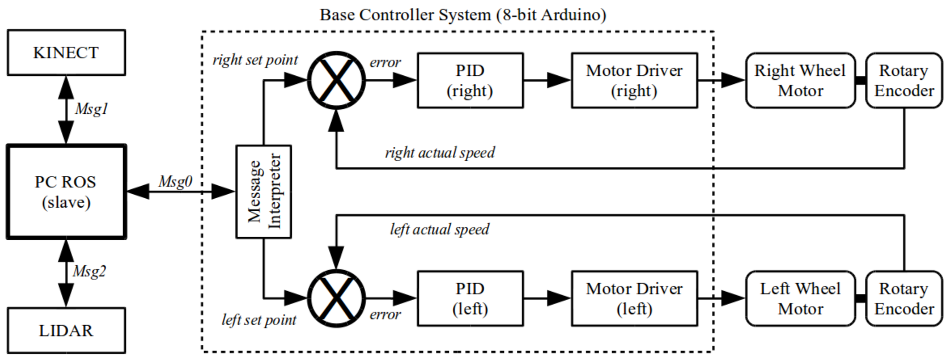

Figure 3. Integrated differential mobile robot configuration

Figure 4. Experimental differential mobile robot

C. Robot Operating System (ROS)

Robot Operating System (ROS) is an open source software framework primarily based on UNIX platform for operating robots. It is widely conducted for robotics research in last decade and an obvious overview of ROS has been presented by [4]. ROS has three levels of concepts: the File system level, the Computation Graph level, and the Community level. ROS provides the services user would expect from an operating system, including hardware layer abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management. It also provides tools and libraries for obtaining, building, writing, and running code across multiple platforms. The primary goal of ROS is to support code reuse in robotics research and development. ROS is a distributed framework of processes (nodes) that enables executables to be individually designed and flexible at runtime. These processes can be grouped into Stacks and Packages, which can be easily shared and distributed. ROS also supports a federated system of code Repositories that enable collaboration to be distributed as well. Fig. 5 shows the basic concept of ROS with distributed system capabilities.

D. Hector SLAM & Octomap 3D Mapping

Hector SLAM is an open source implementation tool of the 2D SLAM technique proposed in [12]. The technique relies on laser 2D scanning data to detect landmarks and build a grid map of the surroundings. Since wheel odometer has a notoriously unreliable in wheeled mobile robot due to slip factor, Hector SLAM is designed to not utilize the odometry data. Instead, fully relies on fast LIDAR data scan-matching at full LIDAR update rate. Scan matching is playing an important role to align current laser scans with other before or with an existing map. Modern LIDAR come up with high update rate and accuracy has significance contributions for the scan matching process that is fast and accurate pose estimation. This technique is based on a Gauss-Newton approach, which no require for a data association search between beam endpoints nor a complete pose search. Equation (6) shows the algorithm try to find the rigid transformation $\xi = (p_x, p_y, \psi)^T$ to minimize value.

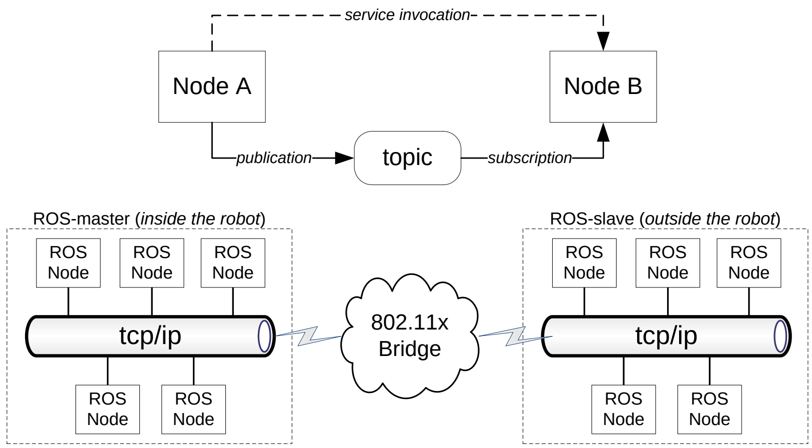

Figure 5. Basic concept of distributed system in ROS

$$\xi^* = \underset{\xi}{\operatorname{argmin}} \sum_{i=1}^{n} \left[1\text{-}M(S_i(\xi))\right]^2 \quad (6)$$

That is minimizing ξ* gives the best alignment of the laser scan with the map. Here, S_i (ξ) is become a function of ξ and represent the world coordinates of scan endpoint $S_i = \left(S_{ix}, S_{iy}\right)^T$. Robot pose in world coordinates are given by (7).

$$S_i(\xi) = \begin{pmatrix} cos(\psi) & -sin(\psi) \\ sin(\psi) & cos(\psi) \end{pmatrix} \begin{pmatrix} S_{ix} \\ S_{iy} \end{pmatrix} + \begin{pmatrix} P_x \\ P_y \end{pmatrix} \quad (7)$$

The function $M(S_i(\xi))$ returns the map value at the coordinates given by world coordinates of scan endpoint (S_i). Once a starting estimate of ξ is given, then the step transformation Δξ can be estimated by optimizing the error calculation according to (8) as seen below.

$$\sum_{i=1}^{n} \left[1\text{-}M(S_i(\xi+\Delta\xi))\right]^2 \rightarrow 0 \quad (8)$$

Implementing first order Taylor expansion of $M(S_i(\xi+\Delta\xi))$ then setting the partial derivative with respect to Δξ to zero delivers the Gauss-Newton equation for the minimization problem as seen in (9).

$$\Delta\xi = H^{-1} \sum_{i=1}^{n} \left[\nabla M\left(S_i(\xi)\right) \frac{\partial S_i(\xi)}{\partial \xi}\right]^T \left[1\text{-}M(S_i(\xi))\right] \quad (9)$$

With

$$H = \left[\nabla M\left(S_i(\xi)\right) \frac{\partial S_i(\xi)}{\partial \xi}\right]^T \left[\nabla M\left(S_i(\xi)\right) \frac{\partial S_i(\xi)}{\partial \xi}\right] \quad (10)$$

Octomap is an open-source implementation tool of the probabilistic 3D mapping technique proposed in [13]. This technique uses a tree-based representation (octree) to provide maximum flexibility related to the mapped area and resolution. It performs a probabilistic occupancy estimation to guarantee repeatability and to deal with sensor noise. Furthermore, compression methods are applied to ensure compactness of memory usage of resulting 3D map. An octree technique is using a hierarchical data structure for spatial subdivision in 3D space where each node represents the occupied space in a cubic volume, commonly known as a voxel. Every single volume is recursively divided into eight sub-volumes until reaching the minimum voxel size. However, the minimum voxel size determines the resolution of the octree. Sensor readings are performed using occupancy grid mapping technique as introduced by [14]. Thus, the occupancy probability $P(n|z_{1:t})$ of a leaf node-n provided by the sensor measurements $z_{1:t}$ is estimated according to (11).

$$P(n|z_{1:t}) = \left[1 + \frac{1-P(n|z_t)}{P(n|z_t)} \frac{1-P(n|z_{1:t-1})}{P(n|z_{1:t-1})} \frac{P(n)}{1-P(n)}\right]^{-1} \tag{11}$$

The current measurement z_t, a prior probability $P(n)$, and the previous estimate $P(n|z_{1:t-1})$ determine the update of occupancy probability $P(n|z_{1:t})$. Nevertheless, this value is subject to the sensor that generated z_t for sure.

3. RESULTS AND DISCUSSIONS

A. ROS Node Configuration

Based on the system configuration as seen in Fig. 1 which is used in this experiment, there are two main sections; remote station section and mobile robot section. All of the nodes in the entire system communicate over Transmission Control Protocol (TCP). Beside generates a 2D map, Hector node provides the laser odometry values as well for Octomap node in order to generate 3D map. Fig. 6 shows the proposed configuration of ROS node which is used in this experiment.

A remote station (left side) act as ROS-slave where the humans are able to operate mobile robot remotely, while a Mobile robot (right side) act as ROS-master where all sensors are connected to do measurements. Every node either in the ROS-slave or ROS-master is using certain topics as subscriber or publisher in order to complete the mission like what operator wants. The human operator set the goal position to be explored by the robot using Joystick device based on current visual information and location. The base controller on the mobile robot will maintain the linear and angular velocity of the robot using PID controller time after time to achieve the goal as seen in Fig. 3 and current location (position and heading) is estimated by using Hector SLAM which is use laser scan match and wheel odometry. Beside generates the odometry function, Hector SLAM generates a 2D map as well, and the odometry topic (/odometry) will be used by Octomap as localization input in order to generate 3D map instead of estimate localization using depth cloud data from Kinect which will increase the computation cost.

B. Constructed 2D & 3D Map

A 3D map obtained from Kinect sensor using Octomap was exactly located on the top of 2D map, shows that localization conducted by Hector SLAM was well performed. RVIZ runs on ROS platform which provides attractive visual interaction facilities to accommodate human-robot interaction impressively. Fig. 7 shows the occupied map from explorations in our experiment and visualizes using RVIZ (GUI). All of the topics from any kind of messages which belongs to nodes are could be obtained by RVIZ. As seen in Fig. 7 operator can use RGB image from Kinect sensor, kind of nodes contain the topics to be visualized and 2D/3D map to be analyzed as the main purpose in term of human-robot interaction that has been done.

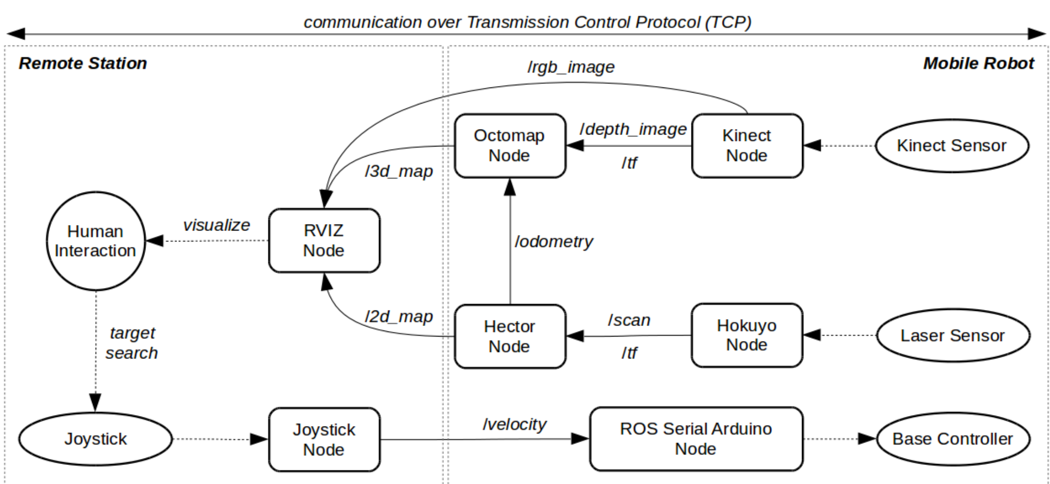

Figure 6. Proposed ROS node configuration of the system

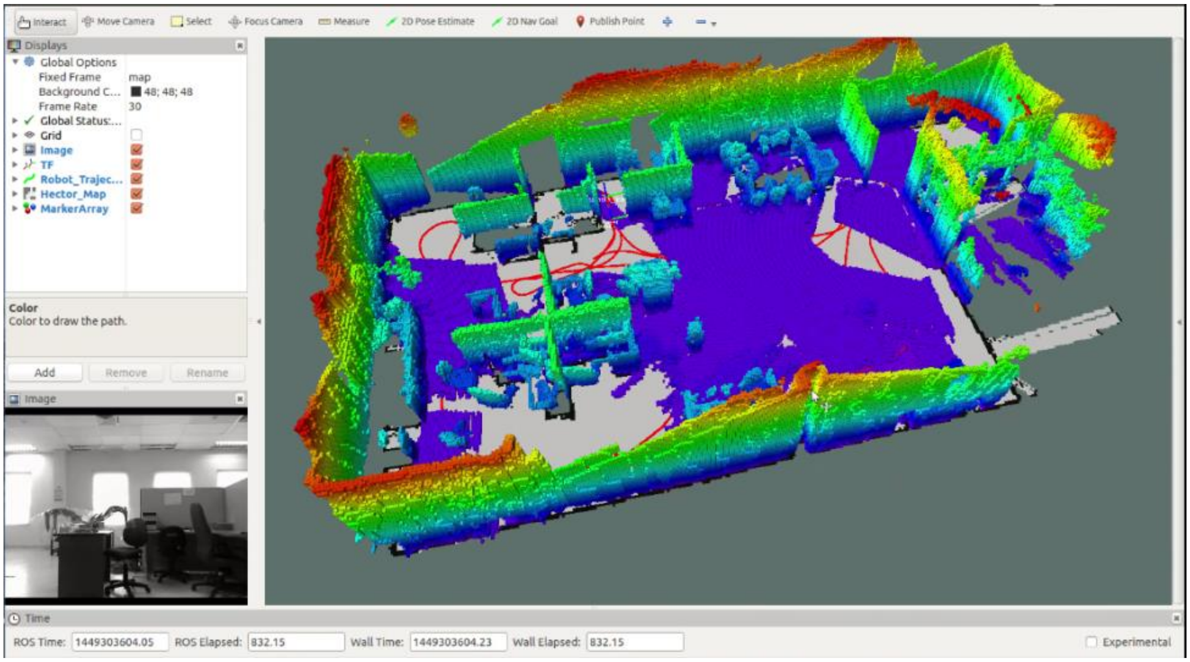

Figure 7. Visualize 2D & 3D occupied map using RVIZ

4. Conclusion

The Implementation of ROS framework in large-scale robot service development is leveraging the growth of robotics technology significantly. In this research, we proposed the application of ROS framework for human-robot interaction to accommodate indoor mapping missions. In this experiment, localization methods become a big issue in term of accuracy and precision. Hector SLAM which comes up with laser scan match and wheel odometry fusing method provides the best performance so far. 3D mapping using Octomap together with Hector SLAM to do localization instead of use depth cloud localization, which provide low memory consumption and low computation cost. The proposed methods of ROS framework accommodate the design of human-robot mission effectively over the complexities using nodes and messages concept. The ROS framework subject to distributed system and the team works oriented which will simplify the development of complex robotic control system such as the experiments which already described in this paper.

Acknowledgement

This project is supported by Malaysian Ministry of Higher Education under Fundamental Research Grant Scheme (RDU130142) and Exploration Research Grant Scheme (RDU130604).

References

[1] R. A. Brooks, “New approaches to robotics.,” *Science*, vol. 253, no. 5025, pp. 1227–1232, 1991.

[2] P. Hinds, T. Roberts, and H. Jones, “Whose Job Is It Anyway? A Study of Human-Robot Interaction in a Collaborative Task,” *Human-Computer Interact.*, vol. 19, no. 1, pp. 151–181, 2004.

[3] P. Salvini, M. Nicolescu, and H. Ishiguro, “Benefits of Human – Robot Interaction,” *IEEE Robotics & Automation Magazine*, pp. 98–99, 2011.

[4] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Mg, “ROS: an open-source Robot Operating System,” in *IEEE International Conference on Robotics and Automation*, 2009.

[5] E. Voisan, B. Paulis, R. Precup, and F. Dragan, “ROS-Based Robot Navigation and Human Interaction in Indoor Environment,” in *10th Jubilee IEEE International Symposium on Applied Computational Intelligence and Informatics*, 2015, pp. 31–36.

[6] K. Kamarudin, S. Mamduh, A. Shakaff, and A. Zakaria, “Performance Analysis of the Microsoft Kinect Sensor for 2D Simultaneous Localization and Mapping (SLAM) Techniques,” *Sensors*, vol. 14, no. 12, pp.

23365–23387, 2014.

[7] J. Choit, S. Ahnt, and W. K. Chungt, “Robust sonar feature detection for the slam of mobile robot,” *2005 IEEE/RSJ Int. Conf. Intell. Robot. Syst. IROS*, pp. 2083–2088, 2005.

[8] J. McDonald, M. Kaess, C. Cadena, J. Neira, and J. J. Leonard, “Real-time 6-DOF multi-session visual SLAM over large-scale environments,” *Rob. Auton. Syst.*, vol. 61, no. 10, pp. 1144–1158, 2013.

[9] M. Pinto, a. P. Moreira, A. Matos, H. Sobreira, and F. Santos, “Fast 3D Map Matching Localisation Algorithm,” *J. Autom. Control Eng.*, vol. 1, no. 2, pp. 110–114, 2013.

[10] S. Cousins, “ROS on the PR2,” *IEEE Robot. Autom. Mag.*, vol. 17, no. September, pp. 23–25, 2010.

[11] S. F. R. Alves, J. M. Rosário, H. F. Filho, L. K. a Rincón, and R. a T. Yamasaki, “Conceptual Bases of Robot Navigation Modeling , Control and Applications,” *Adv. Robot Navig.*, pp. 3–28, 2011.

[12] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf, “A flexible and scalable SLAM system with full 3D motion estimation,” *9th IEEE Int. Symp. Safety, Secur. Rescue Robot. SSRR 2011*, pp. 155–160, 2011.

[13] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, “OctoMap: An efficient probabilistic 3D mapping framework based on octrees,” *Auton. Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[14] H. Moravec and A. Elfes, “High resolution maps from wide angle sonar,” *IEEE Int. Conf. Robot. Autom.*, vol. 2, pp. 116–121, 1985.