JURNAL INFOTEL
Informatics - Telecommunication - Electronics
Website Jurnal : http://ejournal.st3telkom.ac.id/index.php/infotel
ISSN : 2085-3688; e-ISSN : 2460-0997

ROS-based 2-D Mapping Using Non-holonomic Differential Mobile Robot

M.S. Hendriyawan Achmad[1*], Satyo Nuryadi[2], Wira Fadlun[3], Mohd Razali Daud[4]
[1,2,3]Center for Robotics Research, Dept. Electrical Engineering, FTIE, Universitas Teknologi Yogyakarta
[4]Robotic Unmanned System Research Group, FKEE, Universiti Malaysia Pahang
[1,2,3]Jalan Ringroad Utara Jombor Sendangadi Mlati Kabupaten Sleman, Yogyakarta 55285, Indonesia
[4]Universiti Malaysia Pahang, 26600 Pekan, Pahang, Malaysia
*Corresponding email : hendriyawanachmad@uty.ac.id

Received 25 May 2018, Revised 25 June 2018, Accepted 03 July 2018

Abstract - This research proposes a 2-D mapping method by a mobile robot using LIDAR sensor. The mobile robot used is a non-holonomic type with a differential driver designed to operate in an indoor area. The robot applies an occupancy grid map method that uses a probability rule to handle the uncertainties of the sensor. The quality of 2-D occupied map relies on the accuracy of distance measurements by the LIDAR sensor and the accuracy of position estimation. Position estimation is obtained by using the 2-D LIDAR odometry which is based on the laser scan matching technique. This research uses simulation model which has characteristics like real nature. All the robotic software operations are managed by the Robot Operating System (ROS) as one of the most popular software frameworks currently used by robot researchers. The experimental results show that the robot can arrange a 2-D map well which is indicated by the similarity between the reference ground truth and the resulting 2-D map.

Keywords – mobile robot, non-holonomic, LIDAR, occupancy grid map, robot operating system.

I. INTRODUCTION

Mobile Robot technology in the last decade has experienced very significant development. A wide range of functions can be applied to any terrain, either on the surface, in the air, or underwater which has unstructured environmental characteristics [1]. Some critical areas, including industry, military, and households have applied mobile robot technology that interacts with humans to complete tasks more quickly, accurately and tirelessly. Mapping an unknown area is essential for people on a particular mission to know the obstacle model of the area to be explored. A landmark map of an area will assist the robot in planning navigation from a particular location to its final destination through the mapped obstacles. The mobile robot can represent humans to explore the areas at risk, e.g., search and rescue missions. Thus, the robot will quickly determine the navigation path to move across the already mapped areas based on detected landmarks. Thus, the map helps the robot to construct an active navigation path to the location desired by humans as a remote operator.

Currently, the idea of building a map by a moving robot is a matter of issue that remains a hot topic among researchers. The main problem is related to localization and mapping simultaneously by robots. Moravec and Elves have successfully proven the mapping method by knowing the previous location of robots more than three decades ago [2].

The method offered by Moravec and Elves became the initial idea of a mapping method for an unknown region. On the contrary, if the region map is well known, then to get information about the robot location in the area is not an easy thing, and there are many methods offered, among others offered by Borenstein et al. [3] dan Thrun et al. [4]. In 1993, Sebastian Thrun developed a mapping method using neural network techniques based on occupancy grid maps. Map updating is done dynamically using probability model [5]. Then in 2001, Sebastian Thrun also developed a new technique for solving mapping problems using Forward Sensor Models and Expectation Maximization (EM) algorithms to build the 2-D Occupancy Grid Map [6]. In the last two decades, there is a mapping technique studied by researchers intensively, namely SLAM (simultaneous localization and mapping).

The SLAM method allows the robot to know its new position on the map, and the robot also updates the map based on the latest position simultaneously [7], [8]. Some previous studies have proposed many SLAM methods. In general, two types of methods are widely used, namely Extended Kalman Filter [9] dan *Rao-Blackwellized Particle Filter* [10].

Some of the methods mentioned above require input odometry data, to determine the most recent relative position. Kohlbrecher et al. offer SLAM method without relying on odometry data input by using IMU (Inertial Measurement Unit) sensors and LIDAR [11]. The method offered is to combine the matching of 2-D laser scan results by LIDAR with 3-D position estimation by IMU using EKF (Extended Kalman Filter) algorithm. However, this study does not involve the localization on the map that has been formed because the robot runs exploration mode in an unknown region to produce a new 2-D map. The robot exploration mode applies the method offered by Kohlbrecher et al. which relies on a 2-D laser sensor without IMU, assuming that the area used has a flat surface texture and a patterned room.

In this study, we propose a 2-D mapping concept from an unknown environment by a non-holonomic differential moving robot using LIDAR (light detection and ranging) sensors. The proposed method uses the SLAM technique (simultaneous localization and mapping), which is estimating the local position based on the LIDAR scans matching technique and building map simultaneously.

II. RESEARCH METHOD

A. 2-D LIDAR Sensor

The 2-D LIDAR sensor (light detection and ranging) is a distance-measuring sensor using a laser beam covering a two-dimensional area in cartesian coordinates (X and Y). Figure 2 shows LIDAR sensors commonly used in the robotic fields. In general, the output of the 2-D-LIDAR sensor has the polar coordinates, so the polar coordinate output must be converted to Cartesian coordinate output to create a map. Figure 1 shows the projection concept of polar coordinates to cartesian from 2-D LIDAR sensor output.

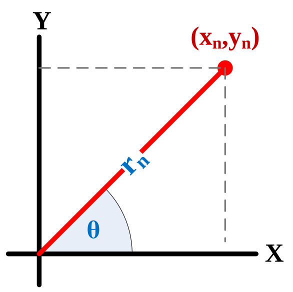

Fig.1. Projection of Polar Coordinates to Cartesian Coordinates on LIDAR Sensors.

The projection of the polar coordinates to cartesian coordinates uses simple calculations using distance information input r_n dan angle of LIDAR scanning resolution θ [12]. Equations (1) and (2) represent the projection formula of the polar coordinates into cartesian coordinates.

$$x_n = r_n \cos\theta \quad (1)$$

$$y_n = r_n \sin\theta \quad (2)$$

With n is the amount of distance information obtained in a single scan period by the LIDAR sensor. The amount n varies depending on the type of sensor used. One type of 2-D LiDAR sensor commonly used in the field of robotics is the production of HOKUYO and SICK as shown in Fig.2.

Fig.2. HOKUYO and SICK Manufacture The 2-D LIDAR Sensors Commonly Used in The Field of Robotics.

B. Pose Estimation

Exploration of the unknown territory by moving robots to obtain accurate maps is primarily determined by the robotic position estimation of each motion in the 2-D coordinates. Pose estimation P such as translation $t \in \mathbb{R}^2$ and rotation $R \in \mathbb{R}^1$. The translation estimation in cartesian coordinates involves the x and y-axes, whereas rotation estimation in cartesian coordinates only involves the z-axis. Figure 3 shows the estimated mobile robot position of DMR type (differential mobile robot) using wheel odometry technique.

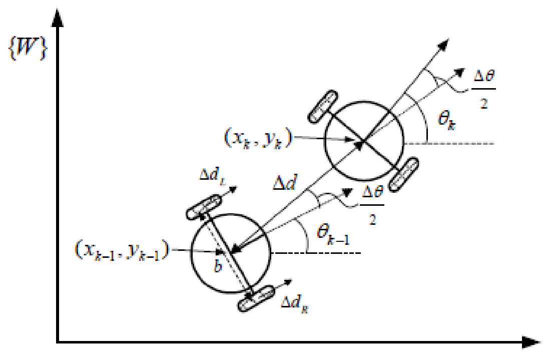

Fig.3. Relative Pose Estimation of The Differential Type Robot in The 2-D Cartesian Coordinates by Using The Wheel Odometry Technique.

Estimation of position in cartesian coordinates using wheel odometry technique relies on a robotic kinematic model. Differential type mobile robot as shown in Fig.3 has a kinematic model that involves the characteristics of the robot wheel to predict the position of either the translational t or the rotation R. The equations (3)-(5) express the robotic position estimation formula based on the kinematics of the differential type mobile robot.

$$x' = \frac{r}{2}(v_R + v_L)\cos\theta \tag{3}$$

$$y' = \frac{r}{2}(v_R + v_L)\sin\theta \tag{4}$$

$$\theta' = \frac{r}{b}(v_R - v_L) \tag{5}$$

The parameter r is the radius of the robot wheel, b is the distance between the right wheel and the left wheel, v_R is the velocity of the right wheel, and v_L is the velocity of the left wheel. These parameters are used by the DMR to estimate the current position of translational movement (x',y') and rotation angle (θ') by using wheels odometry function.

However, position estimation using the robot wheel odometry approach is prone to errors due to the robot wheel slide factor. Those errors will accumulate as global errors and produce inaccurate maps. So that required other solution options to reduce global minimal error as small as possible.

In addition to the wheeled odometry method, position estimation can also be obtained by using consecutive laser scans matching of the 2-D LIDAR sensor as proposed by Kohlbrecher et al. [11]. This method compares two sets of sequential end-points scans (P and Q) to obtain rigid transformation estimation $\xi = (x, y, \theta)$. Estimations of rigid transformation are evaluated using the minimization function (cost function). Ideally, the minimization function produces a value close to zero, indicating that the rigid transformation estimation ξ is optimal. Equation (6) denotes the minimization function between the end-points scan of the p_i source and the q_i target using the Euclidian distance method.

$$\xi = \arg\min_{\xi} \sum_{i=1}^{Np}\sum_{j=1}^{Nq} w_{ij} \left\| p_i - (Rq_j + t) \right\|^2 \tag{6}$$

N is the total number of end-points scans used by LIDAR sensors. $w_{ij} = 1$ is the weight assigned to the pair scan end-points between the source data sets and the matching targets (*correspondence*), otherwise $w_{ij} = 0$ is a weighting for pairs who do not have a match. The rigid transformation of the target q_i data set against the p_i source data set can be obtained using non-linear transformation method, i.e., SVD (single value decomposition) to obtain a minimum Euclidian distance value [13]. The minimization function based on Euclidean distance combined with the SVD transformation method allows for the achievement of position estimation that has a minimum error value.

The ICP (iterative closest point) method introduced by Besl et al. runs the combined method over and over until an acceptable error limit is reached [14]. The ICP point-to-point method estimates a robot positions in 2-D cartesian coordinates (x, y, θ) without relying on information from the robot wheel odometry system.

This study uses laser scans matching approach based on ICP method instead of wheels odometry system to estimate positions. The laser scan matching method has good accuracy on position estimation and is unaffected by the robot wheel sliding factor. However, the laser scan matching method is highly dependent on the number of features for each set of scan data. For areas with little texture, such as elongated aisles, it is relatively difficult to estimate positions compared to areas with many textures accurately. The odometry system is not affected by the texture characteristics of the area to be scanned, but vulnerable to the problem of slipping on the wheel of the robot. Based on the advantages of each estimator, the merging of the two methods is a good option to eliminate each other's weaknesses. The technique which combines two proposed estimators is the Kalman filter method [15]. The input measurement function is obtained from the ICP point-to-point estimator, and the input estimation function is obtained from the wheel odometry estimator. However, the combination method based on the Kalman prediction yet to be implemented in this study.

C. 2-D Mapping

Mapping by the mobile robot relies on two inputs, i.e., distance measurement data by LIDAR sensors and position estimation. Both inputs have uncertainty characteristics due to innate error factor of each sensor. Given the uncertainty factor, the probability approach can provide an effective solution during constructs the 2-D map [4]. One commonly used mapping technique with a probability approach is the occupancy grid mapping method, which is a probabilistic-based mapping technique and able to handle the issue of uncertainty [16]. Equation (7) states the probability of occupying the cell on the map of the grid m model.

$$p\left(m \mid x^t, z^t\right) = \prod_n p\left(m_n \mid x^t, z^t\right) \tag{7}$$

The resulting map using the occupancy grid method is constructed from the data set of position estimation $x_t = \{x_0, x_1, x_2, \ldots, x_t\}$ over time and distance measurement data by LIDAR 2-D sensors $z_t = \{z_0, z_1, z_2, \ldots, z_t\}$ for each position estimated.

III. RESULT

A. Proposed Research Model

In general, the proposed research model is shown in Fig.4 which illustrates the details of a single research flow. The grid-lined mapping requires two inputs, i.e., accurate position estimation and 2-D LIDAR sensor to obtain range measurement information. Human is the operator who operates the robot moves remotely. This concept can be applied to situations that require the creation of maps without direct human presence, e.g., on search and rescue missions in hazardous areas. All parts of the research model run on a software framework known as the Robot Operating System (ROS).

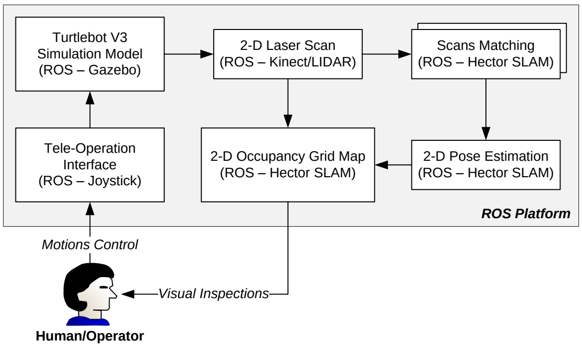

Fig.4. The Proposed Research Model.

B. *Robot Operating System (ROS)*

Software framework plays an essential role in the proposed research model. The entire operation of the system can be adequately controlled and facilitate the development of robot software without starting from scratch. Robot Operating System (ROS) is one of the software frameworks that has been known among robotics researchers because of its reliability and is open source [17]. ROS runs multi-node, multi-master, and multi-slave networking concepts that differentiate ROS from traditional peer-to-peer robotic networking systems dedicated exclusively to specific devices. Figure 4 describes the concept of ROS as a robot framework software.

Multi-node ROS capabilities can control many robotic operational functions. Robots that have a big task run many functions and sophisticated, the implementation of traditional ways will take up the time and energy of researchers. With multi-node ROS properties as shown in Fig.5 will provide efficient flow control of robots. Furthermore, the process of building robot software does not take long because it can be done on a team basis.

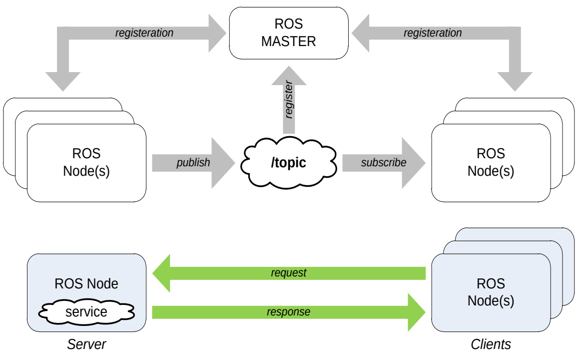

Fig.5. The Working Principle of ROS-Based Software Framework to Control The Operation of All Multi-Node Robot Systems.

IV. DISCUSSION

A. *The Differential Type Robot Kinematics*

Robotic kinematics describes the nature of robot motion (locomotion) based on the differential wheel drive system. Meanwhile, the electronic-mechanical structure discusses the solid platforms that support robotic functions and control systems. The mobile robot technology used in this study is a non-holonomic differential type with the wheeled system. The simulation process uses TurtleBot robot which developed by Willow Garage, as shown in Fig.6.

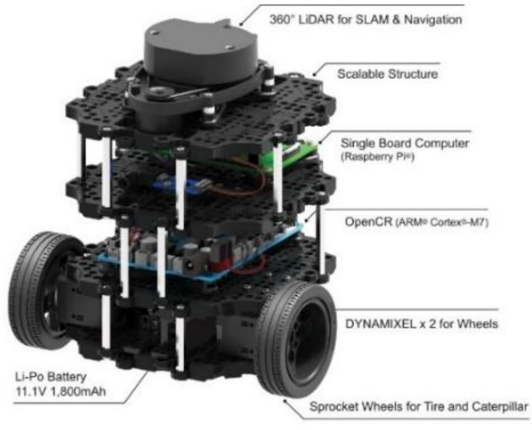

Fig.6. The TurtleBot V.3 Used During Research Simulation Produced by Willow Garage.

The TurtleBot robot used during the simulation experiments has a non-holonomic robot kinematic model and refers to the input of linear velocity (v) and angular velocity (ω). Figure 7 shows the kinematic model of the differential type mobile robot used by TurtleBot. Based on the kinematic model shown in Fig.7, we can express the mathematical model used to accurately calculate the differential velocity values of the left and right wheels to achieve the desired linear and angular velocities. Equation (8) describes the forward kinematic model of the TurtleBot robot that changes the value of speed to the position.

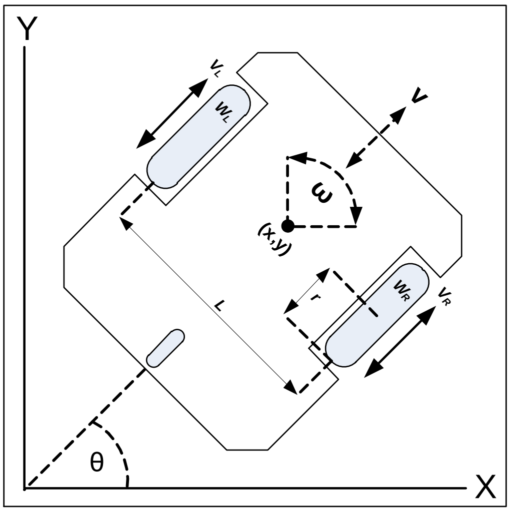

Fig.7. The Robotic Kinematic Model of Non-Holonomic Type Differential Mobile Robot.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (8)$$

Outputs $(\dot{x}, \dot{y}, \dot{\theta})$ shows the position differential over time relative to the initial coordinates based on linear velocity input (v) and angular velocity (ω) supplied by the principal controller based on the relative distance of the robot to the desired position (goal).

In addition to calculating the forward kinematic to obtain an estimated position, TurtleBot also converts the input speed values (linear and angular) into differential speeds for right wheel drive (v_R) and left wheel (v_L) as described by (9) and (10).

$$v_R = \frac{2v + \omega L}{2r} \quad (9)$$

$$v_L = \frac{2v - \omega L}{2r} \quad (10)$$

r and L are the radius of the robot wheels and the distance between two wheels respectively, as shown in Fig.7. If the right and left wheel drive systems can move by the setpoint values v_R and v_L then the position estimated $\left[\dot{x}, \dot{y}, \dot{\theta}\right]^T$ can provide accurate estimations over time.

Setpoint v_R and v_L values must be maintained by the motion control system, to obtain stable robot motion speed. One method to maintain the stability of the differential wheel drive system is to use the PID (proportional, derivative, integral) method. Figure 8 shows the concept of wheel drive control system in a differential type robot using the PID method as a low-level layer that integrates with the Robot Operating System (ROS) as an application platform at a high-level layer.

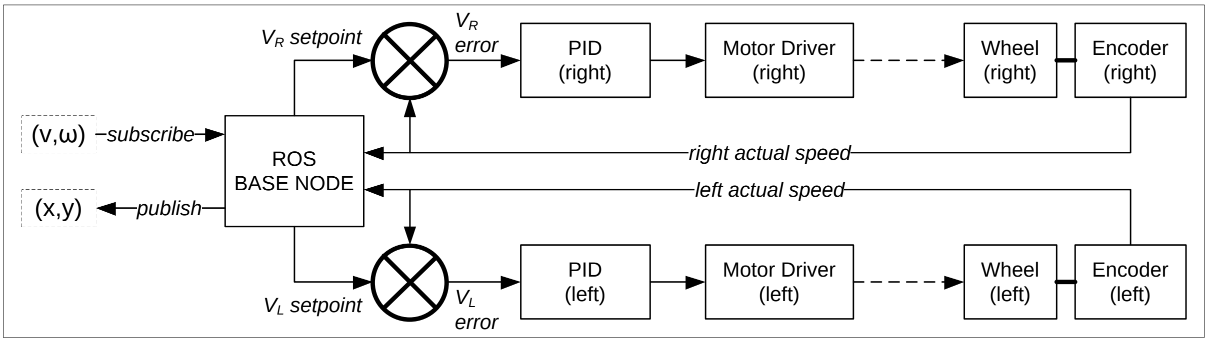

Fig.8. PID-based Motion Controller System for Differential Type of Mobile Robot.

B. Mechanicals-Electronics Structure of TurtleBot

The TurtleBot robot model used in this study is version 3 which has more complete specification than previous versions, including the use of 2-D LIDAR sensor which replaces the RGB-D sensor in the previous version. Table 1 describes the property of electronics and mechanicals system of TurtleBot V3.

Table 1. The Electro-Mechanical Property of Turtlebot V3.

Property	Parameter
Max. Linear Velocity	0.22 m/s
Max. Angular Velocity	2.84 rad/s (162.72 deg/s)
Maximum Payload	15kg
Size (LxWxH)	138mm x 178mm x 192mm
Weight	1kg
MCU	32-bit ARM Cortex®-M7 with FPU (216 MHz, 462 DMIPS)
IMU	Gyroscope 3 Axis Accelerometer 3 Axis Magnetometer 3 Axis
Peripheral	UART x3, CAN x1, SPI x1, I2C x1, ADC x5, 5pin OLLO x4
Battery	Lithium polymer 11.1V 1800mAh / 19.98Wh 5C

The robot kinematic model shown in Fig.7 and expressed by (8)-(10) refers to the design of the robot structure as shown in Fig.8. The kinematic is related to the parameters of the wheel radius (r) and the distance between the wheels (L) on the kinematic equations of differential type mobile robot.

C. ROS Framework Implementation

The distributed operating system for the robot is intended to regulate the work of every part of the robot, both sensing operations, and motion control operations. This research uses ROS (Robot Operating System) as a distributed operating system to facilitate the process of developing robot software. The concept of ROS as shown in Fig.5 allows the process of developing a robot system partially but still integrated into the ecosystem. ROS allows the development process to be done in groups and can prevent the occurrence of re-inventing the wheel which will take longer development time. Figure 9 shows the ROS implementation diagram on the remote-control system of the mobile robot.

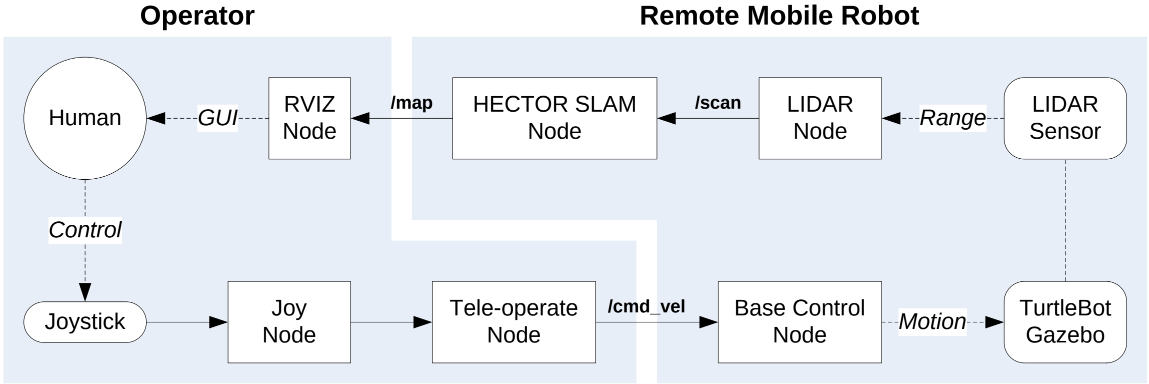

Fig.9. The Configuration of ROS Ecosystem on The Robot Software System to Perform The 2-D Mapping Function.

Figure 9 provides an overview of the concept of interaction between human operators and mobile robots through visual feedback information consisting of maps and locations. Messages from the base controller which translated from joystick device are subscribed by the robot in the form of linear velocity (*v*) and angular velocity (*ω*). Each robot sensor releases information known as publish. In the experiment, the simulation was done using TurtleBot 3 robot with 2-D LIDAR sensor to get a 2-D map of surrounding environment. The SLAM (simultaneous localization and mapping) function to get location estimation, as well as 2-D mapping, runs Hector ROS node [18]. Maps will be presented visually using the RVIZ ROS node as visual feedback between a human operator and a mobile robot.

D. *Occupied 2-D Map*

When human control the robot remotely, the robot also runs two functions simultaneously, namely: position estimation and 2-D mapping (SLAM - simultaneous localization and mapping). Ideally, position estimation function is performed based on three sensor inputs, namely: 2-D LIDAR, IMU (inertial measurement unit), and wheel odometry. However, this experiment uses only one type of input which obtain distance measurement data by LIDAR based on HECTOR SLAM method. Figure 10 shows the result of 2-D map which is constructed by a mobile robot with remote manual control. The quality of constructed map is indicated with black dots which measured by the 2-D LIDAR distance sensor. The actual map shown with yellow dots is used as ground truth. The map constructed by mobile robot is almost aligned with the reference map (ground truth), indicating the error value of the position estimation function is relatively low.

Fig.10. 2-D Mapping Results by The Mobile Robot Using 2-D LIDAR Sensor Based on HECTOR SLAM Method on The ROS Platform.

The occurrence of position estimation errors results in map misalignment, as seen in Fig.11. The occurrence of map misalignment is generated by HECTOR SLAM due to position estimation error. Generally, the position estimation error is caused by two things. First, 2-D LIDAR scans have few features (e.g., in less textured areas). Second, the high speed of robot motion is resulting in two LIDAR scanning frames which have few corresponding points so that difficult to estimate both the translational and rotational motion of the robot body.

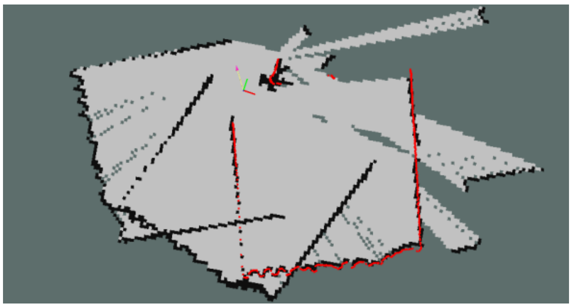

Fig.11. An Example of A Misalignment Error Caused by An Inaccurate Position Estimation.

V. CONCLUSION

The ROS robot software platform provides the ease and timeliness of complex robotic system development by the research team, as well as prevents re-inventing the wheel by other researchers on the same field of work. The benefit of distributed operating systems in the robotic development is to facilitate the interaction between humans and robots in complex environments, both regarding numbers and tasks. The robotic 2-D mapping mission relies on the accuracy of the position estimation known as localization, which is based on the presence of sufficient landmarks around the robot, the more, the better. Each sensor has an uncertainty value at each measurement that will affect the accuracy level of robot localization. In the future, the accuracy of position estimation can be improved by using the concept of multi-sensor fusion, e.g., using Extended Kalman Filter (EKF). ROS ecosystems that run complex operations on distributed communications networks will require high service speed, so the factor of data transmission speeds on the network is a top priority. 2-D mapping operations by the mobile robot should use a combination of multi-sensors to get a low position error. The uncertainty effects on the sensor that could potentially decrease the accuracy of position estimation can be reduced by re-calibrating the sensor for several times iterations. The loop closure technique should be run to correct the mapping results when the robot returns to a previously recognizable position.

ACKNOWLEDGMENT

The research was funded entirely by the Research Grant of DIPA KOPERTIS V Yogyakarta 2017, Grant Number: 244/BPDIPA/VI/2017.

REFERENCES

[1] B. Siciliano, O. Khatib, and F. Groen, Mobile Robots in Rough Terrain, vol. 12. 2004.

[2] H. P. Moravec and A. Elfes, “High Resolution Maps from Wide Angle Sonar,” in IEEE International Conference on Robotics and Automation, 1985, vol. 2, pp. 116–121.

[3] J. Borenstein, H. R. Everett, and L. Feng, Where am I? Sensors and Methods for Mobile Robot Positioning, vol. 119. The University of Michigan, 1996.

[4] S. Parsons, “Probabilistic Robotics by Sebastian Thrun, Wolfram Burgard and Dieter Fox, MIT Press, 647 pp., $55.00, ISBN 0-262-20162-3,” Knowl. Eng. Rev., vol. 21, no. 3, p. 287, Sep. 2006.

[5] S. B. Thrun, “Exploration and Model Building in Mobile Robot Domains,” in Proceedings of the IEEE InternationalConference on NeuralNetworks, 1993, pp. 1–12.

[6] S. B. Thrun, “Learning Occupancy Grids with Forward Models,” in Proceedings - IEEE/RSJ International Conference on Intelligent Robots and Systems, 2001., 2001, vol. 3, pp. 1676–1681.

[7] M. Cummins and P. Newman, “FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance,” in The International Journal of Robotics Research, 2008, vol. 27, pp. 647–665.

[8] L. Jaulin, “Range-only SLAM with occupancy maps: A set-membership approach,” IEEE Trans. Robot., vol. 27, no. 5, pp. 1004–1010, 2011.

[9] S. Riisgaard and M. R. Blas, SLAM for Dummies, vol. 22, no. June. 2004.

[10] G. Grisetti, C. Stachniss, and W. Burgard, “Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling,” in Proceedings - IEEE International Conference on Robotics and Automation, 2005, pp. 2432–2437.

[11] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf, “A flexible and scalable SLAM system with full 3D motion estimation,” in 9th IEEE International Symposium on Safety, Security, and Rescue Robotics, SSRR 2011, 2011, pp. 155–160.

[12] H. Achmad and M. R. Daud, “3D Image Construction Using Single LRF Hokuyo URG-04LX,” in Colloquium On Robotics, Unmanned Systems And Cybernetics 2014 (CRUSC 2014), 2014, pp. 76–79.

[13] S. Marden and J. Guivant, “Improving the Performance of ICP for Real-Time Applications using an Approximate Nearest Neighbour Search,” in Proceedings of Australasian Conference on Robotics and Automation, 2012, pp. 1–6.

[14] P. Besl and N. McKay, “A Method for Registration of 3-D Shapes,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 14, no. 2, pp. 239–256, 1992.

[15] G. Welch and G. Bishop, “An Introduction to the Kalman Filter,” SIGGRAPH 2001, vol. 7, no. 1, pp. 1–16, 2001.

[16] A. Milstein, “Occupancy Grid Maps for Localization and Mapping,” in Motion Planning, 2008, pp. 381–408.

[17] 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.

[18] S. Kohlbrecher, J. Meyer, T. Graber, K. Petersen, U. Klingauf, and O. Von Stryk, “Hector open source modules for autonomous mapping and navigation with rescue robots,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 8371 LNAI, pp. 624–631, 2014.