



EXPLORATION OF UNKNOWN ENVIRONMENT WITH ACKERMAN MOBILE ROBOT USING ROBOT OPERATING SYSTEM (ROS)

M.S. Hendriyawan Achmad, Nur Afzan Murtdza, Nor Anis Aneza Lokman, Mohd Razali Daud, Saifudin Razali and Dwi Pebrianti

Robotics and Unmanned Research Group, Instrument & Control Engineering Cluster, Faculty of Electrical and Electronics Engineering, Universiti Malaysia Pahang, Pekan, Pahang, Malaysia

E-Mail: hendriyawanachmad@gmail.com

ABSTRACT

In this paper, authors present a series of work in order to explore unknown environment consists of path and obstacles with the Ackerman model of wheeled mobile robot (car-like). Robot operating system (ROS) is used as a basic operation platform to handle the entire of operation, such as sensor interfacing, 2D/3D mapping, and path planning. ROS is an open source framework and huge constructions consist of methods. The Ackerman mobile robot is a car-like robot as commonly sees, and techniques that have been done in this experiment can be applied to the commercial vehicles as a part of autonomous navigation system which is emerge as big issue nowadays. In this work, we had composed robust existing methods to solve the mapping problem with Ackerman mobile robot. It was concluded that the performance of the proposed work is robust for large mapping within unknown construction building.

Keywords: mapping, ackerman mobile robot, robot operating system.

INTRODUCTION

Mapping is an important step when a mobile robot wants to explore the unknown environment [1], and robot should be able to move and find the pathways to avoid obstacles and achieve the goal [2]. Mobile robot navigation technology is now increasingly sophisticated, with technology of 2D/3D mapping [3]. Many studies have been done on the implementation of 2D/3D SLAM (simultaneous localization and mapping) to determine the goal position respect to the current position of the robot in an area that has not been recognized previously [4].

There are many studies discussing about environment mapping by using a robot with non-holonomic differential model, but it's need a major modifications when adapt to the commercially of the shelf (COTS) vehicles environment due to complexity of kinematics design. Car-like robot which is use Ackerman steering become a solution model to synchronize between research and real implementation in term of SLAM base autonomous navigation.

ROS is an open source framework which has capability to handle all software layers from low-level up to high-level layers. Because of that, everyone all over the world can build and share ROS stack and package for certain purposes. High demand on ROS stack from researchers was increasing exponentially the number of ROS stack production year to year. Therefore, ROS provide good methods which were can be chosen base on our own desires and purposes. It is also the product of trade offs and prioritizations made during its design cycle [5].

The contribution of this work is a SLAM solution that is able to describe an unknown environment as a map and possible to reduce the risk on rescue missions for instance. We were able to simultaneously localize and build real time map more than 2 km² of our robotic laboratory area less than one hour. The purpose of this

experiment was not merely to demonstrate implementation of ROS in car-like robot, but to produce a 2D and 3D map to a particular environment for use by other type of robots for path planning in future operations. Our results satisfy the qualification compare with ground truth.

This paper is organized as follows. Section 2 describe related researches which have been done to strengthen our work, then Section 3 describe the Ackermann steering, Ackermann pose estimation, structure of ROS, and followed by the steps required to generate a map using hector SLAM in Section 4. At each stage, come up with solution issue is illustrated and analyzed in section 5. Finally, Conclusions are described in the last Section. Figure-5 shows the test bed use in this experiment.



Figure-1. Ackerman mobile robot (with Kinect and PTU-Hokuyo LRF).

RELATED WORK

S. Gariddo *et al.* have presented an efficient mapping method on unknown environments with mobile



robot. The key is solving the simultaneous localization and mapping problem with good exploration strategies. This paper presents the solutions to answer the problems of exploration and build map with a single robot. The exploration algorithm is proposed based on the Voronoi fast marching (VFM) method to determine a motion plan toward the most unexplored and free zones of the environment [6]. S. Chen *et al.* have presented a probabilistic fuzzy system which is designed to reduce stochastic and stochastic uncertainties for more precise localization and map building. The experimental results authenticate the proposed method has success and reliable for mobile-robot localization and map building with various unexpected disturbances [7]. S. Kohlbrecher *et al.* have presented a method such as open source software modules for the development of complex abilities which include Hector-SLAM in deteriorate urban environment. The results have been successfully applied and tested originally in the annual RoboCup Rescue competition [8]. A gentle discussion of the Robot Operating System (ROS) has been presented by Quigley *et al.* [5] and Li *et al.* [9] presented map building using sonar sensors and Dezert-Smarandache Theory (DSmT). In this work the authors used computational techniques, such as probability, fuzzy sets, neutro-sophic, and neural networks theory for map building. They used a Pioneer robot for experiments and built a 3D grid map of the environment.

SYSTEM DESCRIPTION

Ackermann steering

Ackerman steering (another name is kingpin steering) gave a solution about the difference of angles between steering wheels during vehicle turning. This difference is caused by the fact that each wheel should turn in a different radius to avoid slip, so the inside wheel has to be tilted with more angle than the outside wheel. This principle precisely prevents tire slippage (especially at higher speeds). This design is realized by double pivoting system, where the pivots are working with range of angle. The angle has to guaranty that the Ackermann axis, end of the pivot, and center of the rear axle are in line. So that robot will achieve better driving behavior [10].

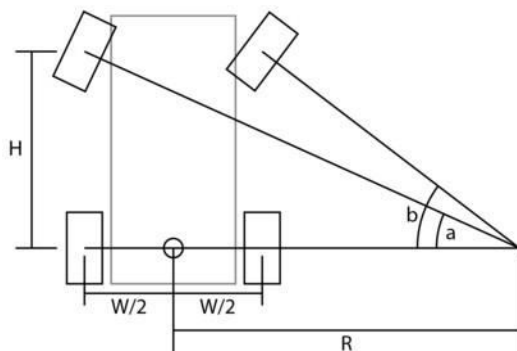


Figure-2. Ackermann steering model.

Figure-2 shows the concept of Ackermann steering. During make a turning, the angle of front wheels are different in order to make same pointing (cross section) among right tire axle, left tire axle, and rear tire axle. The intention of Ackerman geometry is to achieve perfect trajectory when following the path around a curve without slip. The angles b and a can be calculated by:

$$a = \arctan\left(\frac{H}{R + W/2}\right) \quad (1)$$

$$b = \arctan\left(\frac{H}{R - W/2}\right) \quad (2)$$

Ackermann pose estimation

The pose is estimated by its position (x, y) and the bearing (θ) . The actuations are the robot speed (v) and the steering angle (δ) . The only parameter is the wheelbase L .

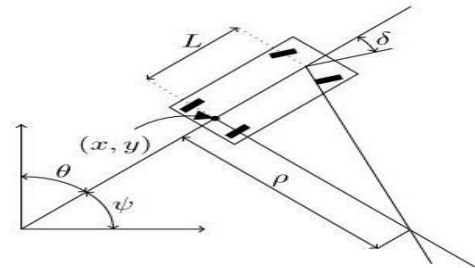


Figure 3. Ackermann pose estimation

From Figure-3 the model is:

$$\dot{x} = v \times \cos(\psi) = v \times \cos\left(\frac{\pi}{2} - \theta\right) \quad (3)$$

$$\dot{y} = v \times \sin(\psi) = v \times \sin\left(\frac{\pi}{2} - \theta\right) \quad (4)$$

$$\theta = \frac{v}{L} \times \tan(\delta) \quad (5)$$

Bearing (θ) is critical value to determine latest position in localization [11]. Error on this part will cause drifting on position measurement. We can measure the bearing by use steering angle (δ) and robot speed (v) as Equation (5), IMU, visual odometer, or laser odometer. Fusing among them will suppress drift error and get the best estimation of robot pose respectively.

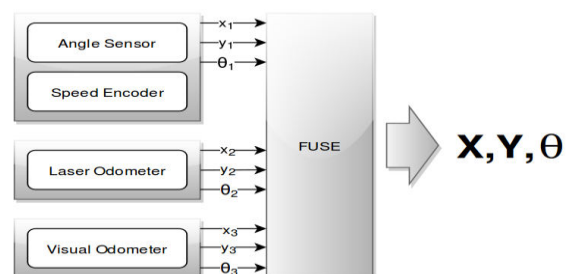


Figure-4. Sensors fusing for 2D pose estimation.



In general, pose estimation problem as estimating the robot state, known as $P_t = [X \ Y \ \theta]^T$, using wheel encoders with steering angle sensor, laser odometer, and visual odometer. The estimate position of X and Y are the Universal Transverse Mercator (UTM) coordinates [12] of the robot. As an estimator, extended Kalman filter (EKF) [13] is a good option to do so, using robot model as shown by Equation (3-5) with another position sensors output as shown by Figure-4.

Robot operating system (ROS)

Building software for robots is quiet tough. Moreover, the technology and scope of robotics continues to grow. Different models of robots can have significant varying hardware and software. Researchers work on their own way with lack of open community supports, then so many case of recurrent of an invention as we all know the term of reinventing the wheel. Since hard to find equal capabilities and experience to work with robot framework of any single researcher, robotics software architectures must be developed for large-scale software integration purposes. To meet these requirements, many robotics researchers, have previously designed a big variety of frameworks to handle complexity and accommodate rapid prototyping of software for experiments. Currently, many robotic software frameworks are used in academia and industry [14]. Each of these frameworks was designed for a particular purpose for sure, as a response for thread-off from other available frameworks, or to improve the existing ones as a competitive effort in industrial technology.

ROS was created to fulfill a specific set of requirements when developing large-scale service robots as a part of the STAIR project by Stanford University and the Personal Robots Program by Willow Garage, but unintentionally the resulting architecture is much more common and easily accepted than the service-robot and mobile-manipulation domain [5].

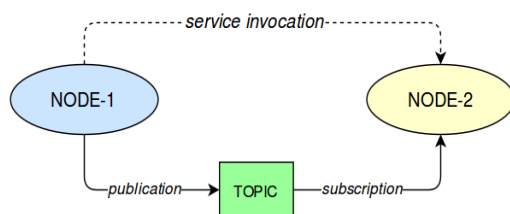


Figure-5. ROS basic concepts.

Every single node run on ROS has particular function, being publisher or subscriber for other nodes. Topic is a name that is used to identify the content of the message. Messages are transmitted via a transport layer with publish/subscribe semantics. Figure-5 shows a node transmitted a message (publication) to a given topic, while another node captured a message (subscription) from a given topic. Publisher and subscriber architecture accommodates communication among separate software nodes, which could be co-located on the same processor or

on different processors. The main purpose of this concept is use to allow individual researchers to be able to keep their own programming style, while still maintain the integrity of the entire system [15].

Figure-6 describes ROS communication procedure use in information exchange among nodes. Publication of messages from single node to multi-node with same topic at same time is possible, preceded by service request signal from node before broadcast the message between publisher and subscriber as shown in Figure-5.

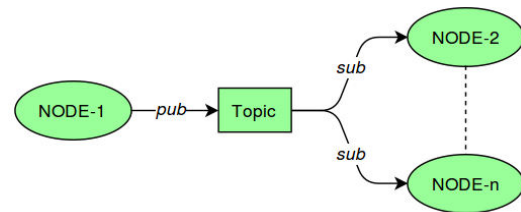


Figure-6. ROS node communication.

Hector SLAM

Hector SLAM integrates a 2D SLAM system based on laser scan matching and 3D navigation technique involving an inertial measurement unit (IMU) [16]. Hector SLAM does not use the odometer information from rotary encoder and steering angle measurement, which gives the widely access to implement this approach in aerial and ground robots operating in uneven terrains. However, it needs high accuracy of position measurement by using IMU or LIDAR odometer in order to ensure the performance of localization and mapping.

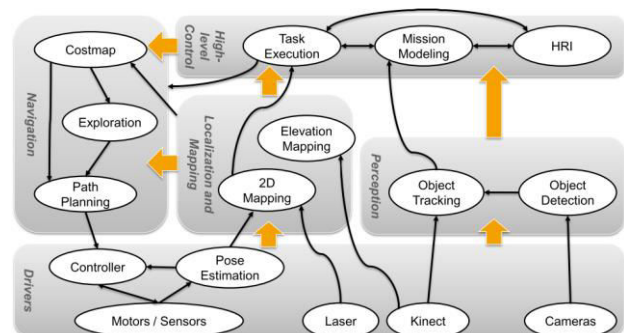


Figure-7. Hector SLAM concepts for mobile robot.

Figure-7 describes the 2D pose estimation is obtained subsequent to optimization of laser beam alignment with the map given so far. Once the endpoints have projected and mask with the actual map then the occupancy grid map are estimated. Scan matching problem is answered by utilizing a Gaussian-Newton equation, which provides method to finds the rigid transformation that fits the map with the laser endpoints. Finally, the 3D state estimation for the navigation filter is relying on EKF. However, an Inertial Measurement Unit (IMU) is an optional choice, such as in the case of aerial



robots with lack of laser or visual features. Otherwise, it will not necessary to be used regarding features abundant [17].

RESULT AND DISCUSSION

Odometer performance

Table-1. Bearing error measurement.

	Bearing Error (rad)	
	Visual odometer	Laser odometer
Mean	0.00076	0.00106
Std. Dev.	0.00477	0.00355

Odometer is key-factor which determines the performance of localization and mapping. Invalid measurement data from this step will mess final result in term of SLAM. This experiments use Laser odometer [8] and Visual odometer [18] to measure the rotation (bearing) and translation of rigid body of the robot, then compare with ground-truth. Table-1 shows bearing error measurement.

The numbers of samples in bearing error observation are taken from 17 difference angles from -40° to $+40^\circ$ with 10° resolution, and compare with ground-truth. Figure-8 shows the performance of bearing measurement by visual odometer and laser odometer.

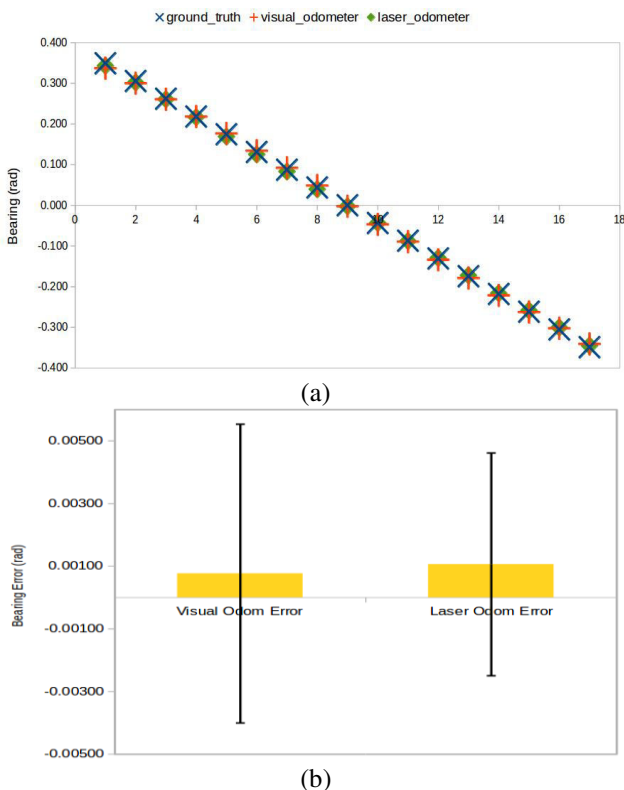


Figure-8. Bearing measurement. (a) Visual & laser odometer performance (b) Statistical error.

Figure-8(b) shows mean of bearing error measurement from both of visual and laser odometer respect to the ground-truth. The mean error from visual odometer is less than laser odometer, but the standard deviation error tends to be wider. Nevertheless, the performance of both sensors is relatively precise in terms of small amount of errors, as shown in Figure-8(a).

Meanwhile, to demonstrate the performance of translation measurement by the odometer, Table-2 shows translation error measurement.

Table-2. Translation error measurement.

	Translation Error (m)			
	Laser odometer		Visual odometer	
	X	Y	X	Y
Mean	0.02106	-0.01394	0.01938	-0.01331
Std. Dev.	0.00739	0.00917	0.01690	0.01211

The numbers of samples in translation error observation are taken from 16 position points in square, and compare with ground-truth. Figure-9 shows the performance of translation measurement by visual odometer and laser odometer.

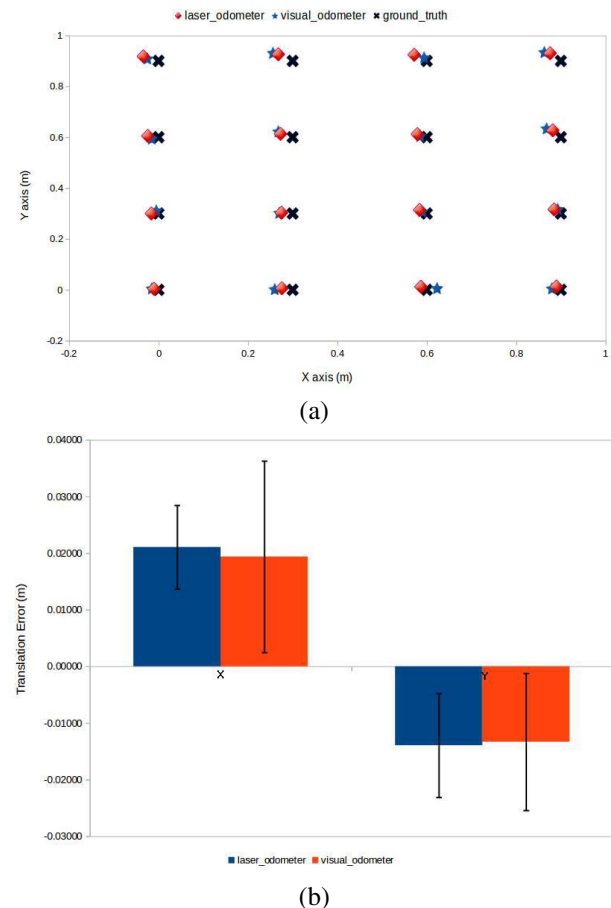


Figure-9. Translation measurement. (a) Visual & laser odometer performance (b) Statistical error.



Figure--9(b) shows mean of translation error measurement from both of visual and laser odometer respect to the ground-truth. The mean error from visual odometer is less than laser odometer, but the standard deviation error tend to wider. Nevertheless, the performance of both sensors is relatively precise in term of small amount of errors, as shown in Figure-9(a).

ROS configuration

Robot operating system in these experiments is taking an important role as a tool with regard to the success of the mission. Figure-10 describe ROS configuration for mobile robot exploration.

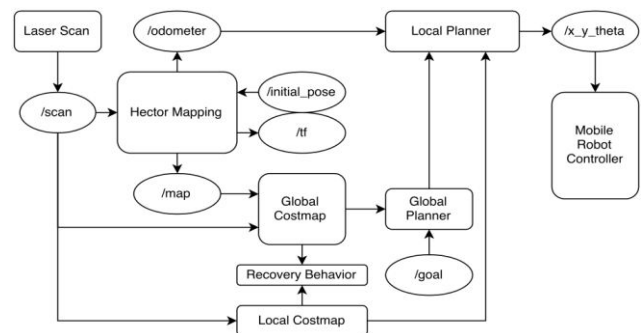


Figure-10. ROS configuration for exploration mission.

During exploration mission by a mobile robot, currently position (x,y,θ) is always updated respect to the target position (goal). Local planner will play this role.

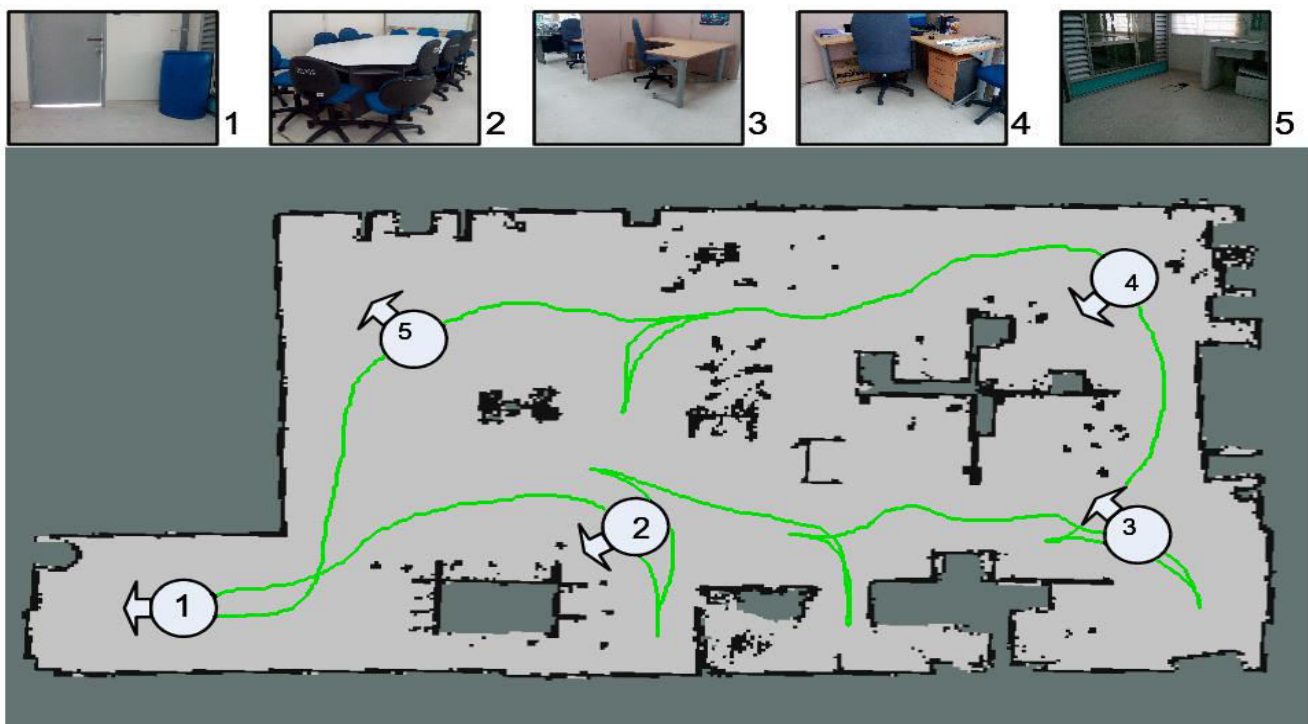


Figure-11. 2D mapping result.

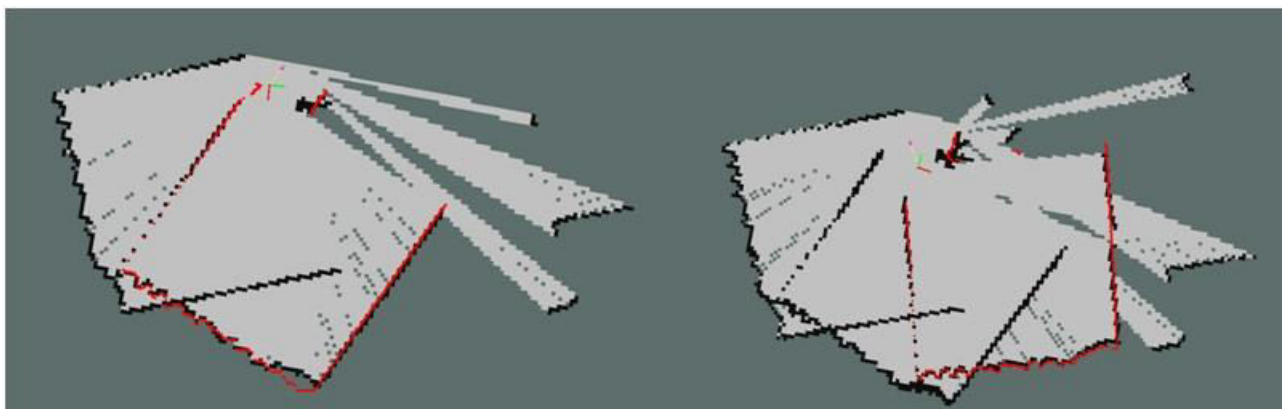


Figure-12. Mapping misalignment.



Figure-10 shows close loop system where global map is reconstructed from sparse maps issued by laser scan within Hector mapping, which is updated simultaneously with rigid body (base) movement of mobile robot to achieve the goal position. Beside produce a map, Hector mapping provide update currently position as odometer. Local planner use odometer information to calculate how far currently robot position with the target position base on map which has been made, and find the path by use A* algorithm. Once path created, robot will follow every single coordinate point along the path as temporary target until arrive at the final destination (goal). In this experiment, goal position is set up manually. But, it is possible to set up goal automatically by use probability function or artificial intelligent (AI).

Environment mapping

Exploration of unknown environment by mobile robot is expected to provide an obvious perception about the structures look like, can be walls, obstacles, or anything which prevent robot to pass it through.

Figure-11 shows the result during environment mapping in Robotics Lab. FKKE-UMP. Dark grays show unexplored area, and cannot use as path even though it is a path in fact. Black dots show obstacles. And the rest are free area where free from obstacles and can be use to create a path. The green line shows robot movement trajectory during exploration. The details of exploration depend on how deep robot has going through. Shape and dimension of robot body become important information to design maneuver algorithm through the obstacles. A path which has been created is could be different with robot movement trajectory for Ackermann mobile robot. In term of robot motion planning, a path is treated as a target and actual trajectory as the output. Poor alignment causes serious problem for achieving the goal, for example in door way. Thus, robust controlling method is needed in this matter, PID for instance.

Instead of use rotary wheel encoder and steering angle sensor, laser odometer has a weakness. Figure-12 shows misalignment during mapping without wheel odometer. This error is caused by miscalculated of the odometer given by Hector mapping. The odometer value is obtained by laser feature extraction then compared geometrically with extracted frame before, known as scan match. The accuracy is depending on how many same features have been collected from 2 frames. After scan match, translation and rotation value are obtained. Predicted position and orientation value is relative to it is initial pose. Thus, changing position and orientation of the robot is detected when there is different geometry between 2 frames with similar features in scan match, otherwise no changing at all. The odometer error occurs in this experiments when the laser did not find objects at it is longest distance in certain ranges of beam angle, yet another angles obtain a constant measurement, for example in the long hall.

Laser odometer misalignment may appear in visual odometer as well, if the environment surrounds lack

of features or none textured. Sensors fusion come up with the solution for this problem as shown in Figure. 4, but yet to be conducted as it is a future works.

CONCLUSIONS

Unknown environment exploration using Ackermann mobile robot tend to be more difficult than differential mobile robot which is commonly used by researcher because of Ackermann mobile robot has bigger turning radius. Hence, the design of motion planning has faced more complex challenges, particularly in narrow place with many obstacles. ROS aims to facilitate face these challenges.

In the other hand, the results of this work will much contribute to the navigation system in some areas in term of autonomous navigation, a public transport vehicles to prevent accident, an industrial vehicles to support the supply chain management systems, and vehicles for rescue purposes in the extensive damage area to accommodate the evacuation team based on exploration map which has been constructed.

Since the current result was obtained a 2D exploration map, in the future work involves a method to construct 3D exploration map using 3D depth measurement devices, tilting laser, RGB-D camera, or stereo camera for instance.

ACKNOWLEDGEMENTS

This project is supported by Malaysian Ministry of Higher Education under Exploration Research Grant Scheme, ERGS, RDU130604.

REFERENCES

- [1] D. F. Wolf, and G. S. Sukhatme. 2008. Semantic Mapping Using Mobile Robots. *IEEE Trans. Robot.* 24:245–258.
- [2] C. Chen and Y. Cheng. 2008. Research on Map Building by Mobile Robots. *IEEE - Second Int. Symp. Intell. Inf. Technol. Appl.* 2:673–677.
- [3] R. Zlot and M. Bosse. 2014. Efficient large-scale 3D mobile mapping and surface reconstruction of an underground mine. *Springer Tracts Adv. Robot.* 92:479–494.
- [4] B. B. Cortes, U. Valle, B. Piv, and X. C. Solé. 2011. Indoor SLAM using a Range-Augmented Omni directional Vision. *IEEE International Conference on Computer Vision Workshops.* 280–287.
- [5] M. Quigley, K. Conley, B. Gerkey, J. FAust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng. 2009. ROS: an open-source Robot Operating System. *International Conference on Robotics and Automation.* 3:5.



- [6] S. Garrido, L. Moreno, and D. Blanco. 2009. Exploration and mapping using the VFM motion planner. *IEEE Trans. Instrum. Meas.* 58:2880–2892.
- [7] S. Chen and C. Chen. 2013. Probabilistic fuzzy system for uncertain localization and map building of mobile robots. *IEEE Trans. Instrum. Meas.* 61:1546–1560.
- [8] S. Kohlbrecher, J. Meyer, T. Graber, K. Petersen, U. Klingauf, and O. Von Stryk. 2014. 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)*. 8371:624–631.
- [9] X. Li, X. Huang, and M. Wang. 2006. Robot Map Building From Sonar Sensors and DSMT. *An Int. J. Inf. Security*. 20:104–121.
- [10] J. Hrbáček, T. Ripel, and J. Krejsa. 2010. Ackermann mobile robot chassis with independent rear wheel drives. *Proceeding of 14th Int. Power Electronics and Motion Control Conference (EPE-PEMC)*. 14:46–51.
- [11] S. Thrun. 2005. Probabilistic robotics. Cambridge-MIT Press.
- [12] GEOINT Sciences. 2007. The Universal Grid System. *Coordinate Systems Analysis (CSAT)*. 1–8.
- [13] G. Welch and G. Bishop. 2006. An Introduction to the Kalman Filter. *UNC-Chapel Hill*. 7:1–16.
- [14] J. Kramer and M. Scheutz. 2007. Development environments for autonomous mobile robots: A survey. *Autonomous Robots*. 22:101–132.
- [15] K. DeMarco, M. West, and T. Collins. 2011. An implementation of ROS on the Yellowfin autonomous underwater vehicle (AUV). *OCEANS 2011*. 1–7.
- [16] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf. 2011. A flexible and scalable SLAM system with full 3D motion estimation. *9th IEEE International Symposium Safety, Security, Rescue Robotics (SSRR)*. 155–160.
- [17] J. M. Santos, D. Portugal, and R. P. Rocha. 2013. An evaluation of 2D SLAM techniques available in Robot Operating System. *IEEE International Symposium Safety, Security, Rescue Robotics (SSRR)*. 1–6.
- [18] M. Labb and F. Michaud. 2014. Online Global Loop Closure Detection for Large-Scale Multi-Session Graph-Based SLAM. *IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS)*. 2661–2666.