Journal of Applied Science and Engineering, Vol. 21, No. 3, pp. 351–360 (2018) DOI: 10.6180/jase.201809_21(3).0006

Auto-maps-generation through Self-path-generation in ROS-based Robot Navigation

Shih-An Li[1], Hsuan-Ming Feng[2]*, Kung-Han Chen[1], Jian-Ming Lin[1] and Li-Hsiang Chou[1]

[1]*Department of Electrical Engineering, Tamkang University,*
Tamsui, Taiwan 251, R.O.C.
[2]*Department of Computer Science and Information Engineering, National Quemoy University,*
Kinmen, Taiwan 892, R.O.C.

Abstract

This paper applies a virtual robot operating system (ROS) platform to concurrently perform the automatic map generation and appropriate path planning for robot navigation applications. The powerful ROS works as a self-constructed robotic facility to perfectly achieve the maps generation and robot localization functions. LiDAR dynamically scanned the required information from the outsides environment and matched the visual maps for reaching the best path coverage and predicting the accurate robot location. ROS-based GAZEBO plant with the flexible and friendly interface is taken to simultaneously imitate the robot environment. In the illustrated experiments, an efficient A* algorithm is approved to build the near optimal routing path within one second executing time. Hector SLAM technology is employed to automatically generate the robot maps for completing nonlinear and complexed navigation applications.

Key Words: Robot Operating System, GAZEBO Simulator, A* Algorithm, LiDAR

1. Introduction

Due to its available modular architecture, robot operating system (ROS) is flexible for designing the specific software in the robot-based platform. The ROS essentially satisfies the perfect and simple communication framework to concurrently manage all processes. ROS is a favorable tool to support an efficient interface and functions module to handle all active jobs in different processes [1]. Based on the contributions of many ROS searchers, there are more software feasibilities in the robotic module, which have been appeared and successfully implemented in the past decade. Thus, their reorganized functions and reused abilities are highly improved in more robotic systems [2,3]. Recently, more and more practical functional modules are available open sources in all over the world. These goods proved that the developed resource will be validly accumulated to achieve the systematic software for people's requirement and objective. ROS supplies people the unanimous and available communication unit to simplify the handling process for huge amount of data. ROS offers the efficacious design on the computational integrations of hardware/software applications. ROS brought the cross-cutting capability for designers to easily implement their thinking in various platforms.

There are various complex and nonlinear tasks appearing to be solved in our daily life. Several autonomous technologies have been taken in the past fewer years to achieve the self-location robot applications. In robot navigation, it is imminently catch the required space information, such as distance and orientation via a series of sensors in the case study. These technologies also records speed and angle of robot to estimate their poses and positions in the working space. In robot navigation application, it repeated these steps to reconstruct the environ-

---

*Corresponding author. E-mail: hmfeng@nqu.edu.tw

ment maps. An intelligent mobile robot performs the great sensory perception in order to realize the autonomous information from the environment of applications. These objects' data is continuously detected and tracked from different space sensors to actually decide robot's next actions [4]. Simultaneous localization and mapping (SLAM) is first introduced by Smith and Cheeseman in [5]. The SLAM is combined both IMU (inertial measurement units) and odometry information to improve the predictive accuracy of robot localization [6]. Robot development software for automatic application generally contains three important technologies (i.e. self-location, path plan and map deployment) to present spatial reality in search space. The visible ROS-based platform is proposed to pay the close attention to conveniently deploy the robot's accurate position maps and make the more improvement on the best selection of path planning and accuracy localization. SLAM contained the great ability for mobile robot to simultaneously update the location and refine the path planning. The actual robot position and precise maps generations are considered to primarily approach into the goal. Therefore, the SLAM-based navigation robot system with the great localized abilities is an important technology to perfectlly handle huge amount of space data [7,8].

Laser range finders are the most popular sensor and it is proposed in this study for achieving the high accuracy and fast reaction. The novel Hector SLAM is known the great matching technology with the high specifications rangefinder to get the consecutive scanning laser data. The required information approaches multi-resolution accuracy grid maps to predict the real robot position [9,10]. Due to the obvious advantage of laser sensory equipment, the SLAM is widely developed in real-life applications. The hardware-based particle filter through the low cost laser scanner is applied together to successfully implement the SLAM-based navigation applications in an uncertain environment [11].

LiDAR collected the spin-image data to create the 3D model with respective to the cooperation of multiple robots. A new three-dimensional feature is auto-generated with the reflection intensity of laser sensor to match the primary curve in the clouding point [12]. LiDAR keeps large scanning range to produce the available visual maps for achieving the full self-location of robot in an unknown environment. SLAM is favorable to complete the real-time robot navigation applications in the indoor scene even if it lost or unknown the necessary routing information in advance [13,14]. The method recovers the weakness of service robot to offer more great jobs in homestay. SLAM actually extended the scanning maps around the robot machine. Thus, the fully routing maps are increased through the maximal discovering area of mobile robot at the same time. Light detection and ranging sensors are known the most popular 3D scanned and a reconstructed machine to predict the global robot position in the ground environment [15].

Due to avoid obstacles and reduce execution time, the mobile robot determined the minimal routing path to lead it into the desired target based on the heuristic objective functions [16]. Path planning is known the most important topic in the application of mobile robot to optimize the shortest distance between the initial point and the last destination. The A* algorithm is applied to develop the adapt path planning in the direct of user-defined cost functions for approximating the desired navigation purpose [17]. A* algorithm is a family of graphic search algorithm, which suitably regulates heuristic cost functions to take the advantages of greedy search algorithm and uniform-cost search (Dijkstra algorithm) to minimize the total cost.

In this study, a brief presentation of the ROS platform and the embed gazebo simulator is shown in section 2. A designed architecture and the A* algorithm presents the path planning in section 3. In section 4, a simulation of the GAZEBO maps generations is shown at ROS platform. The conclusion and future research is illustrated in the final section.

2. Gazebo Simulator Design

Software simulator called Gazebo is regarded as a 3D platform. Robotics Research Lab of University at Southern California has firstly employed and conducted the plant to execute the approximated model [18]. Gazebo player works as the TCP type server through the network architecture. Gazebo player received sensor information from the mobile robot and sent the related control signal to guide the robot action. Gazebo simulates an environment plant to fit the boundary of object in the matrix format of 2D dotted data. Gazebo creates its related outside environment to efficiently approach the target of Multi-Agent or complicated system design. Details are discussed in the follows.

2.1 Gazebo Architecture

In the design of Gazebo architecture, we flexibly create mass, speed, friction and all other physical properties of objects to match the physic conditions from the real world. For example, it contains different actions of push, pull, bump and grab, to simulate an accurate motion. Gazebo platform offered the powerful simulated capacity to guide the related robot motion for fitting various circumstances conditions.

Gazebo is created the virtual world in this simulation by rendering capabilities and dynamic analysis. That is, a real-time behavior is composed of the cascaded actions by forcing to the related joint parts of robot. Gazebo simulates the suitable force or torque in this moment to move objects or articular of robot for interactively acting with each other in the circumstances. The properties of robot models and control strategies must be properly set in advance to drive the virtual motion as the expected response in the real world. Gazebo plant perfectly simulates the dynamic feature from the produced robot models. This paper is focused on generating the training scene with the collected information of indoor space to enhance the reliability of robot motion.

Gazebo is considered as a software simulator which gives a favorable operation interface between the client and server nodes. Designer produces the modularity of actuator, sensors and other required self-made elements through Gazebo's plant to configure the real robot module in the virtual plant. Figure 1 illustrates the detail Gazebo's software module in the blocked diagram. This is an applicable communicated way through the graphic interface for user to conveniently manage a control signal and its response. The perfect module design way is suit to implement the real world functions in ROS simulator.

Gazebo contains the flexible technique to simulate the physical states of various robotic functions in the real-world, such as gravity, light source, etc. the Open Dynamics Engine is determined to detect the suitable outside conditions. For example, movement for robot's joint, collision detection for object's flying, rotating rate for object's movement with respect to its different mass. OpenGL is working as a program library to support the conversion from 2D to 3D patterns. This study choices an OpenGL and its utility Toolkit GLUT to setup as the operator's graphic interface. The OpenGL acts as the great property to plug-in an external module. Its own the higher stability to completely support the authentic display and realize its character. Even if the open dynamics engine needs the large computational resource, the model simulation is perfectly approached into the higher visibility with the slight graphic load.

2.2 Robot Operating System

Robot operating system (ROS) contains the flexible software platform to describe the complex robot feature and to control the robot's behavior through the distributed packages in the computer-based architecture. ROS satisfies the essential resource integration ability which contains tools, libraries and the other embedded software module to approach the desired purpose. ROS works with the common developed tool to design the unify interface which perfectly manipulate all processes in all different jobs. ROS is a distributed computation machine to expend robotics function and conveniently composes all

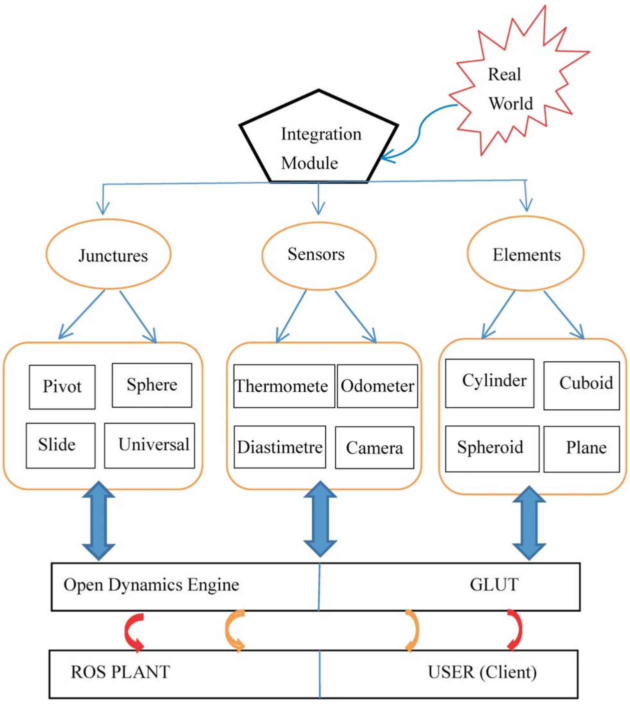

Figure 1. Gazebo platform in the ROS architecture.

the required sub-software within the easy communication interface. Based on user interface design, the more feasibility is appeared to manipulate software module. In addition, the reformed functions and repeated usability of ROS are highly recognized. Therefore, ROS is a powerful tool that it can regulate all computational resource to validly be accumulated for both human's requirements and desired objectives of systematic software.

Due to the great support of ROS, it is comfortable in powering the wish of world's robot. The unanimous and available communication module in ROS is available to simplify the processing of the complicated huge data. ROS efficiently offers resources integration on both hardware and software modules to achieve various applications. ROS owns the cross-cutting ability to easily construct the system architecture for fast implementing the required functions.

3. Maps Generation and Location Systems

3.1 System Architecture

The system architecture in three layers for the proposed stratagems of self-locations purposes are described by the block diagram of Figure 2. In this graphic, the mobile robot keeps running and continuously getting the outsides information from the environment sensors (i.e. odometer or other outside detected elements) for building the environment maps. This SLAM technologies embed in ROS is convenient to approach the self-localization purpose.

From the starting environments of sensors parts, it owns robot hardware parts (i.e. motor drive and sensors). This paper uses LIDAR and odometer as perceptive elements to detect the correct robot location. In strategy layer, the Hector SLAM is proposed to on-line generate the maps by extracting actually the trajectory feature of mobile robot. This way is directly control the robot plant into the desired location by utilizing the valid path planning and navigation maps. In communication Interface design, a web-based technology is fully developed a cross-platform for human to conveniently work in Windows, Linux or IOS platform. In this article, RViz is the powerful user-friendly interface with the graphic environment to handle the specific applications. RViz package is the great support for user's data monitor to plot the related information response in ROS.

Hector SLAM system architecture is firstly developed by the Team Hector Darmstadt of Technische Universität Darmstadt. This system is based on the LIDAR to apply 2D SLAM algorithm and its core concept is used the Inertia Measurement Unit (IMU) to estimate pose of robot in search space. The Hector SLAM with Occupancy Grid Map method is proposed to transform the robot position into a real environment. LIDAR-based mobile robot system always used the great Grid Map implementation to solve the navigation problem. Hector SLAM gathered only the scanning match of LIDAR position sensor to complete the 2D-based Grid Maps in this study.

3.2 Maps Generations Procedure

In procedure of maps generation, robot position must be converted into a numerical value to obtain the precise navigations. In order to approach this goal, the calculated location is directly inferred by the differential or interpolated way to completely determine the real position value. An image convergent-divergent display technology with the Bilinear Filtering method is applied to solve this problem in this paper. The real position value is fitting the distributed samples by the bilinear filtering method. It uses the nearest four positions of the estimated object and go on the following calculation to approach. Figure 3 shows an example of bilinear filtering method,

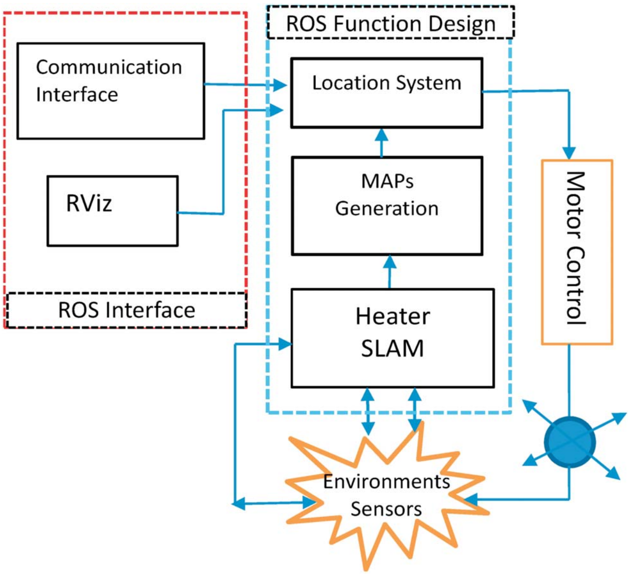

Figure 2. Self-localizations and maps generations in the ROS architecture.

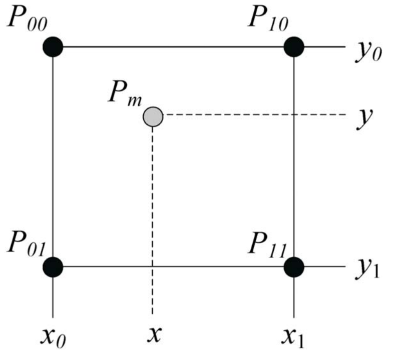

Figure 3. Bilinear filtering examples.

where P_m is the approximated value to be converted as the continuous curve from any occupancy grid maps position [19].

The symbol of mapping position is assigned as P_m. The related position from the occupancy grid map is regarded as $M(P_m)$ and its gradient formula form being described by the following form:

$$M(P_m) \approx \frac{y - y_0}{y_1 - y_0}\left(\frac{x - x_0}{x_1 - x_0} M(P_{11}) + \frac{x_1 - x}{x_1 - x_0} M(P_{01})\right) \tag{1}$$

where the bilinear filtering method choices these closest points P_{00}, P_{01}, P_{10} and P_{11} to approximate the $M(P_m)$ in this case. Derivatives for the specific coordinate x and y are separately obtained by the following two formulas:

$$\begin{aligned}\frac{\partial M}{\partial x} &\approx \frac{y - y_0}{y_1 - y_0}(M(P_{11}) - M(P_{01})) \\ &+ \frac{y_1 - y}{y_1 - y_0}(M(P_{10}) - M(P_{00}))\end{aligned} \tag{2}$$

$$\begin{aligned}\frac{\partial M}{\partial y} &\approx \frac{x - x_0}{x_1 - x_0}(M(P_{11}) - M(P_{01})) \\ &+ \frac{x_1 - x}{x_1 - x_0}(M(P_{10}) - M(P_{00}))\end{aligned} \tag{3}$$

Due to the recursively computation in the previous mathematic functions, all gridded position will be gradually converted into the appropriated value. Laser scanner utilizes the transformed maps with the relation $\xi = (p_x, p_y, \psi)^T$ to extract the minimal error value ξ^* into the estimative end-points in any m-th data set by Eq. (4):

$$\xi^* = \arg\min_{\xi}\left\{\sum_{i=1}^{n}[1 - M(S_i(\xi))], \xi = 1 \sim m\right\} \tag{4}$$

where ξ^* is the minimal value through the operation of arg min. The $S_i(\xi)$ is settled as the approximated function of ξ to reach the best position at the ray end of $S_i(S_{i,x}, S_{i,y})^T$ in the world coordinates spaces.

Therefore, robot position is described by formulas (5) of the world coordinates spaces.

$$S_i(\xi) = \begin{pmatrix}\cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi)\end{pmatrix}\begin{pmatrix}S_{i,x} \\ S_{i,y}\end{pmatrix} + \begin{pmatrix}p_x \\ p_y\end{pmatrix} \tag{5}$$

After assigning $Si(\xi)$ into function $M(Si(\xi))$, its related occupancy grid map is determined to obtain the best evaluated error $\Delta\xi$ by the following formulas. This procedure reaches the minimal error value by satisfying the conditions of (6).

$$\sum_{i=1}^{n}[1 - M(S_i(\xi + \Delta\xi))]^2 \to 0 \tag{6}$$

Several steps are repeated to approach the desired targets by Gauss-network concept for seeking the minimal error value of Eq. (9). The Gauss-network approximation is applied to solve the minimal cost function $\Delta\xi$ problem. Formula is defined by Eq. (7),

$$\Delta\xi = H^{-1}\sum_{i=1}^{n}\left[\nabla M(S_i(\xi))\frac{\partial S_i(\xi)}{\partial \xi}\right]^T [1 - M(S_i(\xi))] \tag{7}$$

In this study, the first order of Taylor series is selected and expanded its value at nodes $M(Si(\xi + \Delta\xi))$ by the following equation:

$$\sum_{i=1}^{n}\left[1 - M(S_i(\xi)) - \nabla M\left((S_i(\xi))\frac{\partial S_i(\xi)}{\partial \xi}\Delta\xi\right)\right]^2 \to 0 \tag{8}$$

The partial differentiation is directly taken into the previous formulas to approximate the desired zero purpose.

$$\begin{aligned}&2\sum_{i=1}^{n}\left[\nabla M(S_i(\xi))\frac{\partial S_i(\xi)}{\partial \xi}\right]^T \\ &\left[1 - M(S_i(\xi)) - \nabla M(S_i(\xi))\frac{\partial S_i(\xi)}{\partial \xi}\Delta\xi\right] = 0\end{aligned} \tag{9}$$

The Gauss-network approximates the minimal cost function of $\Delta\xi$ by the formula 10:

$$\Delta\xi = H^{-1}\sum_{i=1}^{n}\left[\nabla M(S_i(\xi))\frac{\partial S_i(\xi)}{\partial\xi}\right]^T [1 - M(S_i(\xi))] \quad (10)$$

In this paper, Hessian matrix H is defined by Eq. (11) and $\nabla M(Si(\xi))$ is deployed in here to obtain the derivative of $Si(\xi)$ in the matrix form of formula (12)

$$H = \left[\nabla M(S_i(\xi))\frac{\partial S_i(\xi)}{\partial\xi}\right]^T \left[\nabla M(S_i(\xi))\frac{\partial S_i(\xi)}{\partial\xi}\right] \quad (11)$$

$$\frac{\partial S_i(\xi)}{\partial\xi} = \begin{pmatrix} 1 & 0 & -\sin(\varphi)S_{i,x} - \cos(\varphi)S_{i,y} \\ 0 & 1 & \cos(\varphi)S_{i,x} - \sin(\varphi)S_{i,y} \end{pmatrix} \quad (12)$$

We utilize $\nabla M(Si(\xi))$ and $\frac{\partial S_i(\xi)}{\partial\xi}$ to get the minimal value of $\Delta\xi$, which approaches toward the best matching position between the Laser scanning points and a real weights maps.

3.3 The A*-based Location Approximations System

In order to direct the mobile robot into the desired goal, the A*-based learning algorithm is the great ideal to resolve the crucial location approximation problems.

A* algorithm likes a graph search in navigation application. A* algorithm usually deployed an exhaustive cost function to find the least-cost of summation length from the start node to the goal [20]. This A* algorithm is widely used to extract the minimal distance of path plaining. The SLAN is a space navigation technology for mobile robot location approximations. The performance evaluated function is derived by the following heuristic cost Equation (13):

$$F(N) = G(N) + H(N) \quad (13)$$

$F(N)$ is the distance summation of path-cost function $G(N)$ and heuristic function $H(N)$. $G(N)$ gets the total length from the start node to the current position. $H(N)$ is called the momently estimated length cost from the current node to the target position. The optimal path selection cumulates the total length for each visiting node in a traveling cycle. The navigation procedure estimates the smallest traveling path in the step by step way. Pseudo code of A* algorithm with the evaluated function [21,22] is illustrated as follows:

```
Input ($N_{init}$, $N_{goal}$, G, H, Dist)
Create Break pool; Create Connect pool; Bring $N_{init}$ into Break pool;
Check Break pool is not empty;
 While the current node not goes into Ngoal
   Go to Neighbor search cycle
     Do neighbor search cycle
       Check every node N″ in neighbor of current best N* node and not in
       the connect pool.
       G(H″)=G(N*)+Dist(N*, H″)
       H(N″)= Dist(N*, H″)
       F(N″)= G (N″)+H(N″)
       If F(No″)< F(Nn″) not change N* node and no refresh path
       Else
           insert N″ node into the connect pool and the current best N*
           node list is changed to N″ node.
     End neighbor search cycle
 End A* algorithm
 Get the final best node N* list in the Connect pool
```

Figure 4. Pseudo code of A* algorithm.

N_{init} and N_{goal} are navigation initial position and final target, respectively. Dist is a measured function to denote the length cost between two nodes. N″ is a every possible travel node. No″ and Nn″ are denoted as the old and new visiting nodes, respectively. N* is considered as the best node N* list.

Based on previous explanation of A* algorithm. The flow diagram of A*-based location approximations system design is described as follows:

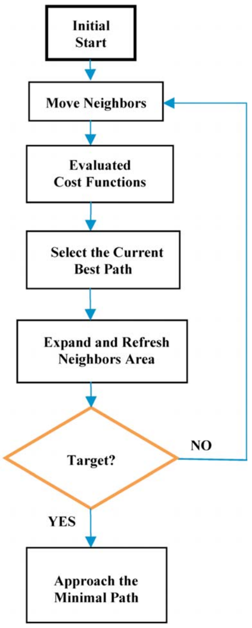

Figure 5. A*-based location approximations flowchart.

Figure 6 and Figure 7 show exploration of path planning cases by the A* algorithm. The blue, red and black circle positions are individually assigned as start, path and goal, where white area is free area and black rectangle is wall. In illustrated simulations, the near optimal total length for Figure 6 case is 192.873. The other summation length for the best exploration of A* algorithm in Figure 7 is 492.979. These two simulations present that A* algorithm can efficiently find the near optimal path within 30 seconds.

4. ROS-based Gazebo Simulations

This study completes several simulations through the Gazebo platform. We utilizes Husky a200 robot as an experimental hardware to implement its work [23]. Husky a200 is differential motion type robot. Laser rangefinder is mounted on the above robot plant. Figure 8 shows the indoor environment simulation by Gazebo software.

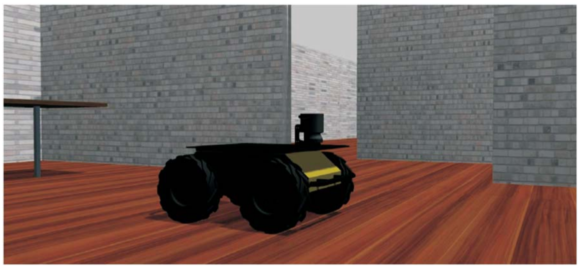

Figure 8. Husky a200 simulations in indoor environment by Gazebo software.

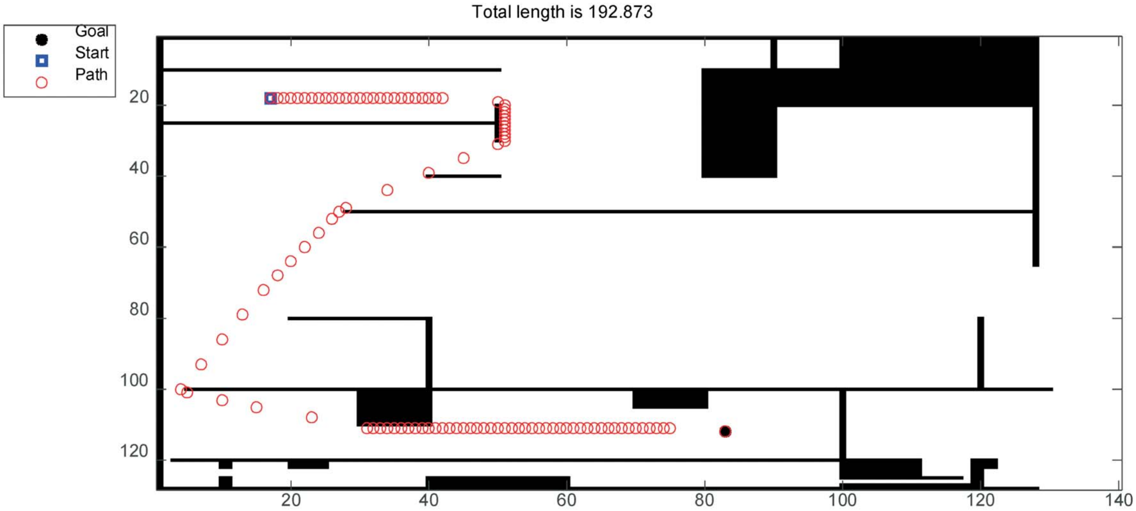

Figure 6. A*-base path planning for case1 simulation.

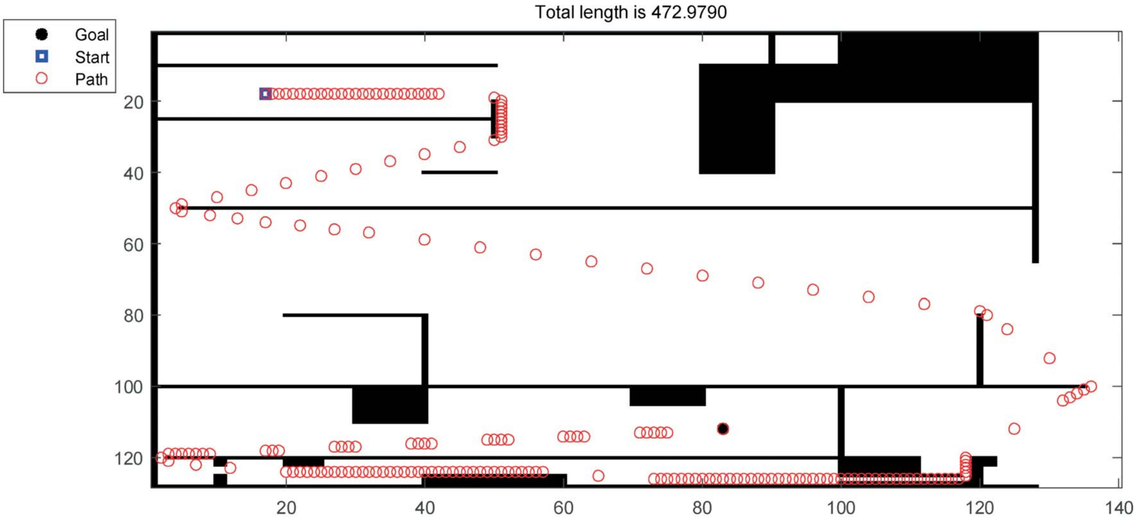

Figure 7. A*-base path planning for case 2 simulation.

These physics parameters for Husky a200 in Gazebo platform almost approximate the real world conditions. There are summarized in the list of Table 1.

RViz graphical interface is the friendly tool in ROS to transform the topic's data into the related graph in Figure 9. This way supports user to conveniently view the result by the powerful stratagem of human thinking. All recorders for Robot, laser scanning nodes, roads and trajectory states are successfully accessed from the Topic. Therefore, such favorable information is drawn in this simulation plant. The simulated system keeps an on-line data updated ability to support the big amount of assistance in the developed software plant.

The maps generations of navigation environments are firstly proposed to determine the path planning in mobile robot experiments. The maps are gradually created to direct that the mobile robot not only smoothly moves but also appropriately explores every visited space. The novel technology of Hector SLAM extracts the known environment information to reach the desired goal. Much more information is collected from the outsides to achieve the best accurate navigation applications. Figure 10 presents Hector SLAM generation maps with the resolution 5 cm/cell. This simulation shows that the ROS-based simulator offers the great support for designer to develop various robotic software services. This virtualization simulator contains the higher reusable abilities to excellently approach the desired goal by the primary regulation in various parameters. Figure 11 shows an experimental simulation case for robot navigations. This experiment completes the whole navigation scenes, which include coordinates maps and laser scanning area within 5 minutes.

Table 1. Husky a200 physics condition list

Size	987*570*247 mm
Weight	50 kg
The most high speed	1.0 m/s

Figure 9. Example of Rviz interface in ROS plant.

The mobile robot uses Hector SLAM technology to concurrently build maps from initial (0, 0) to seven middle positions. Figure 12 shows a navigation experiment from the whole environment. Simulation results include coordinates, maps and laser scanning area in the % unit. Figure 12(a) presents initial position (0, 0) and its scanning area about 30%. Figure 12(b) presents that robot now moves into position (1.0, -0.60) and its scanning area is about 60.95%. Figure 12(c) shows robot's position at (1.5, 2.0) and its scanning area reaches about 71%. Figure 12(d) shows that the mobile robot is going into the

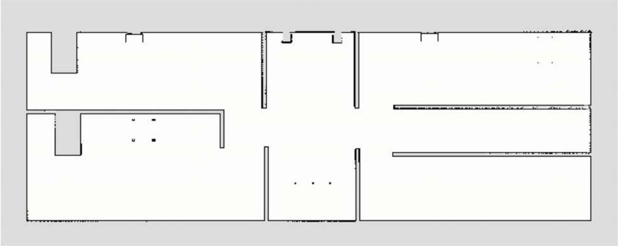

Figure 10. An generated maps with Hector SLAM.

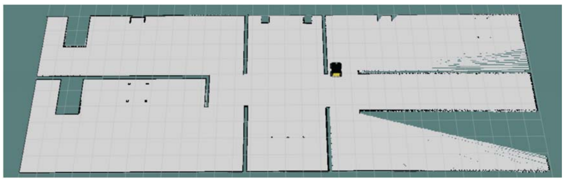

Figure 11. An 2D maps generation in the navigation experiment.

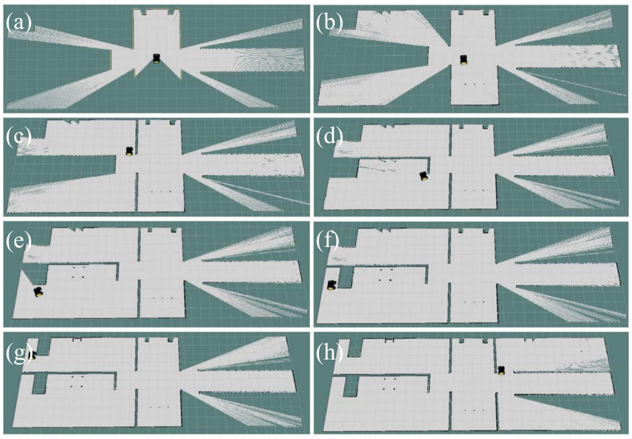

Figure 12. (a) (0, 0) about 30%. (b) (1.0, -0.60) about 60.95%. (c) (1.5, 2.0) about 71%. (d) (-1.5, 5.0) about 71.92%. (e) (-2.0, 13) about 72.74%. (f) (-1.5,14) about 75.28%. (g) (3.0, 14) about 78.87%. (h) (1.2, -3.2) about 90.36%.

position (-1.5, 5.0) and its scanning area about 71.92%. Figure 12(e) and Figure 12(f) are individually shown that robot positions are reaching at (-2.0, 13) and (-1.5, 14). Their scanning areas will increase to 72.74% and 75.28%, respectively. In Figure 10(g), the robot stands at (3.0, 14) location and its scanning area is about 78.87%. Finally, Figure 12(h) shows that the mobile robot goes into (1.2, -3.2) position and it can get the great 90.36% scanning area. Total time cost is not over 5 minutes for this simulation after reaching the 90% scanning area.

The laser rangefinder is mounted on the front of robot head, its scanning range is between -135° and 135°. The mobile robot turns direction into the other rear or far way, and travels its plant into different area for exploring the overall space maps. This study follows an anticlockwise way to rotate robot direction. Left half side part with the higher priority is firstly explored, and then robot goes through the right side to discovery the other space in an unexpected environment. Figure 13 shows coverage and accuracy navigations of two dimensional maps generation and robot's trajectory after completing the space exploration.

5. Concluding and Future Works

This paper develops a self-path-planning by A* algorithm. Robot concurrently generates maps and its location with the SLAM technology. This study is performed to apply in mobile robot navigation applications even if in an unknown environment. This strategy combines the grid occupancy type maps generation and A* algorithm with heuristic function approach to complete applications of the self-path planning, map generation and robot navigation through the ROS-based GAZEBO simulator. In our implementations, the Hector SLAM system works with only one laser rangefinder to automatically generate maps and determine its localization in an unknown environment. Our future works are that the ROS-based simulator owns the higher reusability and support the efficient navigation technique to implement the actual applications of smart home robot.

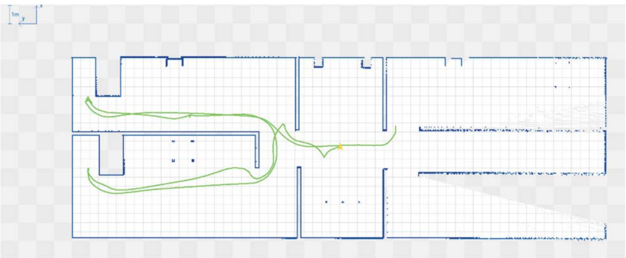

Figure 13. Robot navigation results of ROS plant in an unexpected environment.

Acknowledgements

This work was supported by the Ministry of Science & Technology of Taiwan, ROC, under grant MOST 105-2221-E-032-023, 104-2221-E-507-004 and 105-2221-E-507-002.

References

[1] Kumar, N., Vámossy, Z. and Szabó-Resch, Z. M., "Robot Obstacle Avoidance Using Bumper Event," Proc. of the 11th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI 2016), May 12–14, Timioara, Romania, pp. 485–489 (2016). doi: 10.1109/SACI.2016.7507426

[2] Choi, H., Kim, E. and Yang, G.-W., "Scan Likelihood Evaluation in Fast SLAM Using Binary Bayes Filter," Proc. of 2013 IEEE 11th IVMSP Workshop (2013). doi: 10.1109/IVMSPW.2013.6611891

[3] Jasna, S. B., Supriya, P. and Nambiar, T. N. P, "Remodeled A* Algorithm for Mobile Robot Agents with Obstacle Positioning," IEEE International Conference on Computational Intelligence and Computing Research (ICCIC) (2016). doi: 10.1109/ICCIC.2016.7919535

[4] Wu, X., Abrahantes, M. and Edgington, M., "MUSSE: a Designed Multi-ultrasonic-sensor System for Echolocation on Multiple Robots," Proc. of 2016 Asia-Pacific Conference on Intelligent Robot Systems, 20–22 July (2016). doi: 10.1109/ACIRS.2016.7556192

[5] Smith, R. C. and Cheeseman, P., "On the Representation and Estimation of Spatial Uncertainty," *International Journal of Robotics Research* (1986). doi: 10.1177/027836498600500404

[6] Montemerlo, M., Thrun, S., Koller, D. and Wegbreit, B., "Fastslam: a Factored Solution to the Simultaneous Localization and Mapping Problem," Proc. of 2002 Eighteenth National Conference on Artificial Intelligence, pp. 593–598 (2002).

[7] Dissanayake, M. W. M. G., Newman, P., Durrant-Whyte, H., Clark, S. and Csorba, M., "An Experimental and

Theoretical Investigation into Simultaneous Localization and Map Building," *Experimental Robotics VI*, Vol. 250, Springer London, pp. 265–274 (2000). doi: 10.1007/BFb0119405

[8] Medina, S., Lancheros, P., Sanabria, L., Velasco, N. and Solaque, L., "Localization and Mapping Approximation for Autonomous Ground Platforms, Implementing SLAM Algorithms," 2014 III International Congress of Engineering Mechatronics and Automation (CIIMA), Oct., pp. 22–24 (2014). doi: 10.1109/CIIMA.2014.6983431

[9] Balasuriya, B. L. E. A., Chathuranga, B. A. H., Jayasundara, B. H. M. D., Napagoda, N. R. A. C., Kumarawadu, S. P., Chandima, D. P. and Jayasekara, A. G. B. P., "Outdoor Robot Navigation Using Gmapping Based SLAM Algorithm," Moratuwa Engineering Research Conference (MERCon), Moratuwa, Sri Lanka, 5–6 April (2016). doi: 10.1109/MERCon.2016.7480175

[10] Lepej, P. and Rakun, J., "Simultaneous Localisation and Mapping in a Complex Field Environment," *Biosystems Engineering*, Vol. 150, pp. 160–169 (2016). doi: 10.1016/j.biosystemseng.2016.08.004

[11] Sileshi, B. G., Oliver, J., Toledo, R., Gonalves, J. and Costa, P., "On the Behaviour of Low Cost Laser Scanners in HW/SW Particle Filter SLAM Applications," *Robotics and Autonomous Systems*, Vol. 80, pp. 11–23 (2016). doi: 10.1016/j.robot.2016.03.002

[12] He, Y. and Mei, Y., "An Efficient Registration Algorithm Based on Spin Image for LiDAR 3D Point Cloud Models," *Neurocomputing*, Vol. 151, Part 1, pp. 354–363 (2015). doi: 10.1016/j.neucom.2014.09.029

[13] Kohlbrecher, S., Meyer, J., von Stryk, O. and Klingauf, U., "A Flexible and Scalable SLAM System with Full 3D Motion Estimation," 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), pp. 155–160 (2011). doi: 10.1109/SSRR.2011.6106777

[14] Kohlbrecher, S., Meyer, J., Graber, T., Petersen, K., Klingauf, U. and von Stryk, O., "Hector Open Source Modules for Autonomous Mapping and Navigation with Rescue Robots," RoboCup 2013: Robot World Cup XVII, pp. 624–631 (2014). doi: 10.1007/978-3-662-44468-9_58

[15] Hussein, M., Iagnemma, K. and Renner, M., "Global Localization of Autonomous Robots in Forest Environments," *Photogrammetric Engineering & Remote Sensing*, Vol. 81, No. 11, pp. 839–846 (2015). doi: 10.14358/PERS.81.11.839

[16] Guruji, A. K., Agarwal, H. and Parsediy, D. K., "Time-efficient A* Algorithm for Robot Path Planning," *Procedia Technology*, Vol. 23, pp. 144–149 (2016). doi: 10.1016/j.protcy.2016.03.010

[17] Goto, T., Kosaka, T. and Noborio, H., "On the Heuristics of A* or a Algorithm in ITS and Robot Path-planning," Proc. of 2003 IEEWRSJ International Conference on Intelligent Robots and Systems, Las Vegas, Nevada, pp. 1159–1166 (2003). doi: 10.1109/IROS.2003.1248802

[18] Koenig, N. and, Howard, A., "Design and Use Paradigms for Gazebo, an Open-source Multi-robot Simulator," Proc. of 2004 IEEE International Conference on Robotics and Automation (ICRA), pp. 2149–2154 (2004). doi: 10.1109/ROBOT.2004.1302377

[19] Trivun, D., Šalaka, E. and Osmanković, D., "Active SLAM-based Algorithm for Autonomous Exploration with Mobile Robot," Proc. of 2015 IEEE International Conference on Industrial Technology (ICIT) (2015). doi: 10.1109/ICIT.2015.7125079

[20] Nguyen, H. K. and Wongsaisuwan, M., "A Study on Unscented SLAM with Path Planning Algorithm Integration," Proc. of 2014 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 14–17 May (2014). doi: 10.1109/ECTICon.2014.6839824

[21] Leddisaravi, K., Alitappeh, R. 1. and Guimari'ies, F. G., "Multi-objective Mobile Robot Path Planning Based on A * Search," Proc. of 6th International Conference on Computer and Knowledge Engineering (ICCKE 2016), October 20–21, 2016, Ferdowsi University of Mashhad. pp. 7–12 (2016). doi: 10.1109/ICCKE.2016.7802107

[22] Ueland, E. S., Skjetne, R. and Andreas, R. D., "Marine Autonomous Exploration Using a Lidar and SLAM," Proc. of 36th International Conference on Ocean, Offshore and Arctic Engineering (ASME 2017), Vol. 6 (2017).

[23] Huang, W. H., *Design of Mapping and Exploration System with ROS*, M.S. Desertation, Tamkang University, Taiwan (2016).

Manuscript Received: Oct. 25, 2017
Accepted: Feb. 27, 2018