SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348 – 8549 /doi:10.14445/23488549/IJECE-V7I11P102

Volume 7 Issue 11, 7-12, November 2020
© 2020 Seventh Sense Research Group®

Reseach and Designing a Positioning System, Timeline Chemical Maping for Multil-Direction Mobile Robot

Pham Ngoc Sam[#1], Tran Duc Chuyen[#1]

[#]*Faculty of Electrical Engineering, University of Economics - Technology for Industries, Vietnam*

Received Date: 17 October 2020
Revised Date: 28 November 2020
Accepted Date: 30 November 2020

Abstract

In this paper, the authors presents the method of designing the simultaneous positioning and mapping system (SLAM) about way for mobile robots. Based on the navigation system control, the map simultaneously performs navigation for the movement of the mobile robot when there is an obstacle. Mobile robots are required to both achieve local obstacle avoidance and follow a global path in the process of moving on a virtual environment consisting of known stationary obstacles and dynamic obstacles. All tasks are performed on a omnidirectional robot with high-performance processor for central processing tasks, depth cameras and sensors. The results show the effectiveness, the research direction of using the Robot operating system to control and monitor autonomous robots, self-driving cars as well as developing intelligent robot systems, industrial and civil applications.

Keywords: *Robot Operating System, GAZEBO, Simultaneous Localization and Mapping (SLAM), omnidirectional Robot, Navigation.*

I. INTRODUCTION

The currently, robots have the power to spread throughout the modern industry. Anything that requires repetitive and precise work in an environment that can be carefully controlled and monitored makes Robots a good candidate to replace humans. Current advanced research is rapidly approaching the ability to automate very difficult tasks, although we can consider them tedious (such as in traffic, in the military, driving, in industry, etc.). Thanks to that advantage, autonomous robots are used to replace humans in factories, where robots perform lifting, transporting goods and installing components on production lines, [1, 2].

Therefore, Robots are made to go and do what humans either cannot, or do not want to do. Model SLAM describes the process of building a map of an unknown environment and computing at the same time the robot position with the constructed map. Both steps depend on each other. A good map is necessary to compute the robot position and on the other hand just an accurate position estimate yields to a correct map [9, 11].

In recent years, because navigation, localization, path planning and mapping are an essential step in a mobile robot, many of the approaches used is simple heuristics successively searching for face connected cells or configuration with or without theminimization of a criterion. Therefore, it is widely used in the design of control systems for various types of objects such as self - driving cars and autonomous robots as shown in [2, 9, 10, 13, 16].

The Simultaneous mapping and mapping algorithm (SLAM) could play an indispensable role in robot navigation and become an increasingly hot topic in robot applications. This is a method that allows the creation of incremental maps and recognition of surrounding obstacles in the unknown environments in [3]. The robot's position can be successfully estimated without using expensive sensors and the mobile robot creates an incremental map about environment.

Authors also use essential tools to execute the navigation plan on autonomous robots in many different working environments, to create diversification for the working process of Robot. The quest for mobile robot path planning has received considerable attention from many researchers. Several studies in this area have focused on avoiding moving obstacles in complex situations, including the avoidance of static and dynamic obstacles, [5, 12, 15].

The inspired by the aforementioned studies, the aim of this article focuses on implementing SLAM to recognize robot locations and environmental conditions in unknown areas as well as perform autonomous navigation with the mobile robot on the Jetson TX1 processor.

II. THE MULTANEOUS MAPPING AND MAPPING OF FOR OMNIDIRECTIONAL ROBOT

In the design of motion control systems for mobile robots, there are more modern ways to perform motions effectively that require optimization in the robot's trajectory in order to achieve the goal, avoiding obstacles and smooth operation. In this case, performing concurrent mapping and mapping is an outstanding way to overcome the problem that is the lack of information about the operational environment. The SLAM-based algorithms used for the robot when it performs its first movements in an unknown environment are extracted based on ambient data generated from a number of sensors and cameras. To this end, many algorithms have been developed around the world for the purpose of extracting maps that facilitate precise movements of robots. In fact, the method is widely used to perform the SLAM mission in an indoor or outdoor environment when the robot is in motion. This technique involves constructing a road map of different environments and using this map to infer the location of the robot's path. Once the collection and processing of information from the environment through sensors and map creation is completed, the generated map can be saved and used in the path navigation for the robot, [8, 9, 10, 14, 16].

To describe the SLAM mission is used for omnidirectional robots. In general, the process of creating this task is organized by the main components: robot component, control component and SLAM component. The first part covers the transition between the robot link and the most important thing is the relationship between the robot base coordinates and the position of the sensor, the cameras will collect environmental data. The control part is designed to control the robot's movement through keyboards or automatic control circuits. The last part that plays a major role in performing the SLAM task with Gmapping is the central microprocessor element. Additionally, the map_sever tool can be used to save the generated map in both an image file and a data file that aids the robot's travel and navigation process, [9, 11, 13, 16].

III. THE MOTION SYSTEM DESIGN FOR OMNIDIRECTIONAL ROBOTS

A. The controller design for robot

Through the study, the following results were obtained: The omnidirectional omnidirectional robot 90 degrees apart. Oxy gen represents the global coordinate axis, the distance between the wheels and the robot center is determined. The robot's motion will be determined for the navigation stack on the way, or when encountering robot obstacles. As the global coordinate chosen in Figure 1, it is clear that the robot's velocity contains three components: the linear velocity along the Ox axis and the Oy axis an angular speed.

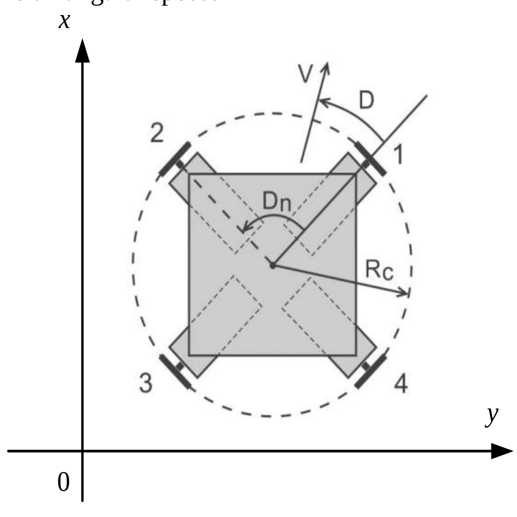

Fig 1: The coordinate axis system omnidirectional robot

On figure 1 R_c robot radius, angle between motor 1 and motor n spindles D_n is equal to 90^0 (n-1). Initial motors mount design assumed fixed mounting all four motors and wheels to the box bottom plate. However, such design does not ensure all wheels touch floor constantly and first motion experiments demonstrated path instability. Then authors added simple suspension to two adjacent wheels. It improved path stability dramatically even without any springs and absorbers. Other two wheels are kept fixed mounting.

If we give qual to zero then move is straight-line motion and corresponded motor number n power Pnl can be defined by formula (1) in accordance with [11].

$$P_{nl} = K.K_l.V.\sin(D + D_n) \qquad (1)$$

Here K is power per cm/s factor defined by motor characteristics K_l, (at this formula), K_t, and K_r (at next formulas) are calibration factors connected to wheels friction and motors resistance and depend on velocity, course and curve values. These factors discussed below.

To implement motion at curve different from zero we have turning the robot for full circle relative to the floor as soon as the robot traveled 2/C path length. This is equivalent to adding the robot rotation around its geometrical center at circular frequency ω = V.C, with ω is speed. Robot geometry difine $\omega.R_c$ = w.r equity for the rotation, where r is wheel radius and w is the wheel rotation circular frequency. Required rotation means adding supplement velocity, V_t = w.r to each motor perpendiculary to the motor axis. Using V_t in formula (1) and considering D + D_n =90^0 we can calculate additional motors power P_{nt} required for turning.

$$P_{nt} = K.K_t.V.(R_c.C) \qquad (2)$$

Total motor power for move motion is Pn = Pnl + Pnt. Time to apply calculated motor power is T = L/V.

The move motion type cannot ensure rotation around the robot geometrical center. Rotate motion used instead and specified by: Angle A to turn, Frequency F of rotations in angle units per second. For this case motors power P_{nr} and time T can be calculated as:

$$P_{nr} = K.K_r.2\pi.F \qquad (3)$$

$$T = A / F \qquad (4)$$

The benefit of these two simple motion types is motor power permanence in time for each individual motion. The motors power is a constant of time at equations (1) to (3). This reduces motor controller algorithms complexity. However, such useful motions as, for example, linear travel and simultaneous robot rotation cannot be executed by applying static motor powers. The such motion implementation requires dividing travel path to short sections (about 2 cm in the robot path length) and issuing separate move or Rotate command to motor controller per each section. At maximum robot speed 100cm /s, it requires sending up to 50 commands per second from onboard computer to motor controller that may overload communication channel and the controller command queue.

B. The navigate for robot

The map creation, mapping and route planning are important navigation tasks for autonomous robots. In a ROS environment, navigational data values provide appropriate buttons and themes that reliably and accurately move the mobile robot from one location to another. The navigation stack creates a secure path for the robot to work on, by processing data from the image metrics, sensors and environmental maps. For navigating the mobile robot, the stack is a mapping system for building the map system, while the memory contains the local planner and the global planner is responsible for linking them to achieve get navigated targets and solve robot path planning problems. The geometry process creates the thread and takes velocity data from the thread to ensure the robot reproduces the rotational speeds for the robot. The navigation stack creates a secure path for the robot to make its best and safest path, by processing data from the image measurement, sensor stitching, and environmental mapping stages. When problem navigate the omnidirectional robot, the setup of the stack is a mapping system for building the map, which contains the local planner and the global planner responsible for linking them to achieve Navigate goals and solve complex scheduling problems. The geometry that publishes the topics will get the velocity data from the newer thread and make sure the robot reproduces these path speeds when it encounters the robot's direction problem wall following theory for mobile robot is a technique makes mobile the robot navigate beside the wall without shocking it and also avoid hitting any obstacles during navigation near the wall wall following is useful for SLAM projects of indoor environment because mobile robot can move in most area of the plan without supervisor by using wall following technique.

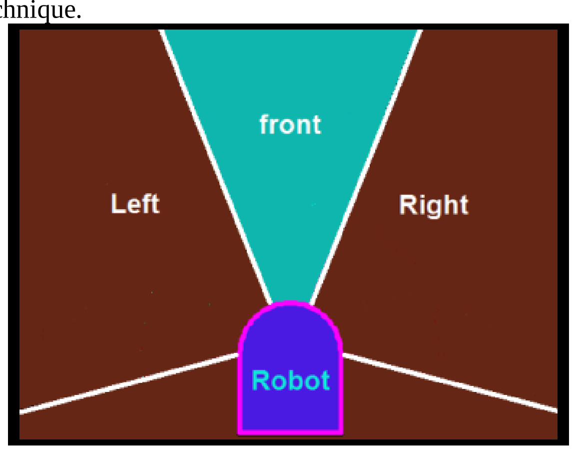

Fig 2: The robot's visibility range and action when making navigation

The global planning tool and local planning tool: The global planning tool takes over the omnidirectional mobile robot's current position and desired target to create the shortest path to navigate the robot considering the obstacles from the static map. However, the actual path where the omnidirectional mobile robot is that of the local planner, it can be considered as a controller that performs more tasks. The local scheduling tool combines current sensor readings to create avoidance strategies for animal obstacles and generate trajectories to follow the global path to navigate the omnidirectional mobile robot .

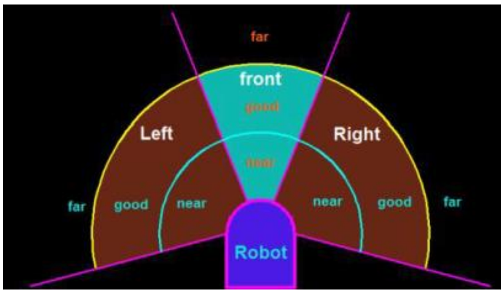

Fig 3: The vision range of the robot performing the navigation process

In each situations, the mobile robot react, a suitable movement for following the right wall by making the right side of mobile robot to be at good range to follow the right wall and not allowing all sides of robot be at near range to avoid crashing any walls and obstacles at all time of navigation. Change the reaction of any one of these situations will affect the behaviors of work, like not following the wall or crashing it.

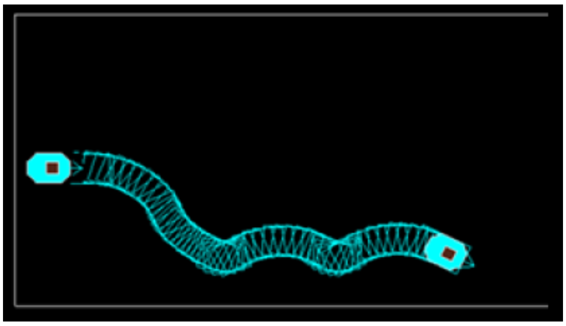

Fig 3: The navigating in open space until Robot finds a wall

The Plan tool to position according to the map structure when this authors using SLAM, the robot's current position and desired target let the robot out the shortest path to arrive at the navigation decision for the robot to consider the obstacles objects from the static map. However, the actual path the robot will choose to take is that of the mapping planning program, which can be thought of as a controller that performs many better tasks. With this local map planner, it's always been to incorporate existing sensors to create strategies to avoid objects on the way for animal obstacles and create trajectories to give the robot omnidirectional follow the outlined global path.

IV. THE SIMULATION RESULTS AND EXPERIMENTATION

When reseach evaluate the capabilities of mobile omnidirectiona robots, some tasks are performed in simulations by designing a robot simulator as well as an actual robot platform.

A. The model of omnidirectional robot

The hardware architecture of the omnidirectional robot is shown in Figure 4, where each module performs several tasks in the robot's operation:

Fig 4: Components of the Omni robot

In the experiments, the authors actually perform some tasks in the operation of omnidirectional robots on smart machines. The computer is embedded with the microchips linked to the robot's control system to directly process information from a series of sensors, then transmit commands to the application microcontroller. Then, if taking pictures from the environment facing omnidirectional robots, as well as measuring the distance between the omnidirectional robot and when encountering an unknown obstacle, the omnidirectional robot is equipped with cameras to recognize and send data. controller data to process information. In which, the host camera on the robot's head is the robot's eye including the smart camera and the identification camera while it can perform 360 degree rotation and the laser scan range within 13.5m creates map data. used for the mapping process. The control circuit uses new onchips such as STM32FXXX and TMS32F411XX which will recognize the control signal from the computer and then direct the signal to the MOSFET bridge circuit to operate the motors to control the robot's speed.

B. The simulation results

To evaluate the capabilities of mobile omnidirectional robots, some specific tasks are performed in simulations by designing an omnidirectional robot simulation program that is similar to an actual robot platform walking in the environment. Reality: There are also many different directions like people, who are going to encounter obstacles and need to handle obstacle avoidance situations. When the omnidirectional robot performs the navigation process automatically identifies through smart sensors, the smart cameras will avoid these obstacles safely. Here we perform simulations on gazebo software to simulate and create maps. This is a software with a powerful support tool to visually simulate SLAM in the working environment of omnidirectional robots.

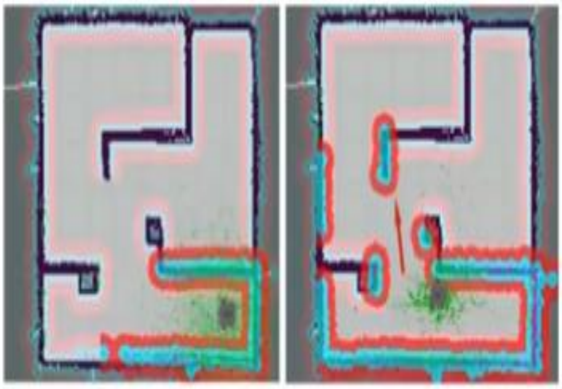

Fig 5: Visual map builder in the GAZEBO environment

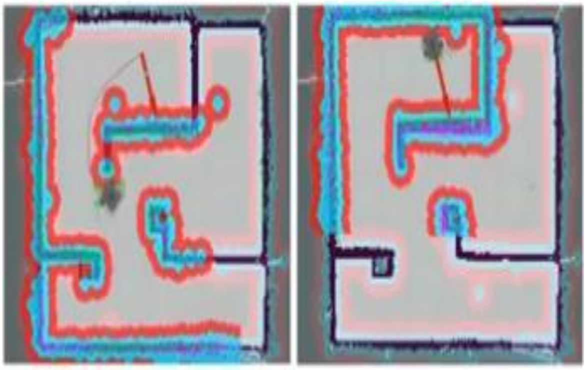

Fig 6: The model of an omnidirectional mobile robot depicted in GAZEBO

Figures 5 and 6 show the map built on gazebo, this map creates strict walls of the omnidirectional robot's path, during the movement of the omnidirectional robot it always identifies the lips. field around the robot to get the essential data that will be used to build and create a road map for the robot.

Fig 7: The obstacle avoidance simulation

The figure 7 shows the environment with the detection of obstacles that are walls in GAZEBO. When motion begins, sensors update local grid maps with new obstacles in the field of view, the cost map uses laser sensor information to create a local cost map before the omnidirectional robot. The local planner publishes the local plan with configurable view distance and local path modifications based on this distance, the chosen view distance is 0.7m. The local path created by the local planner is defined as the green line while the red line shows the global path of omnidirectional robot activity. As can be seen in Figure 7, after avoiding the obstacle, the omnidirectional robot can follow the desired path for a short time and omnidirectional robot move to the target.

C. The experimental results

The design of the simultaneous mapping positioning system for the omnidirectional mobile omnidirectional robot has achieved the following results:

In figure 8 illustrates the results of the navigation when combining 2 RPLidar sensors and Astra camera on a real robot platform in a realistic map environment.

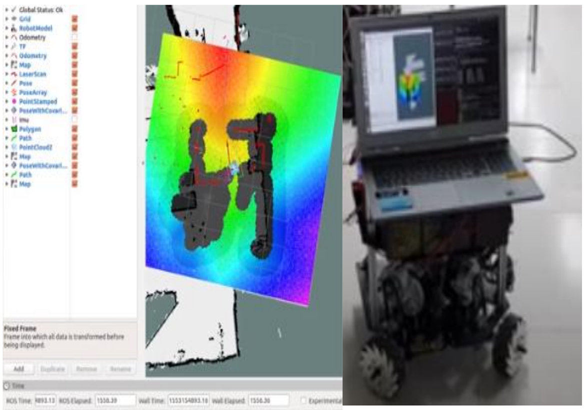

Fig 8. The navigation for omnidirectional robot in actual map

The results of this study show that the working efficiency of the robot is really good when it comes to tracking the path created by the global planner and avoiding obstacles that are animal or stationary obstacles not on the plate. The global map and omnidirectional robot are still aware. The outer edges of the walls and humans are also successfully recognized by the laser scanner and marked with black dots and lines shown in Figure 8. Furthermore navigation problem for omnidirectional robot in real map with dynamic obstacle has also been solved, this issue also indicates that reseach and designing a positioning system, timeline chemical map for multil-direction mobile omnidirectional robot, results are higher than the studies in [3, 9], this shows that this study is completely applicable in practice, in industry, military and defense and security.

V. CONCLUSIONS

In this article the authors presented a new problem on the work of building omnidirectional mobile robots. The research process shows good efficiency of Slam model when using data stacks to construct maps in 2D environment. The results of this article also successfully carry out route map planning for omnidirectional robots when traveling in normal mode and where there are both static and unidentified obstacles due to counter The local path is constantly updated in the simulation environment and the actual images are always accessible. These tasks were based on data generated from smart sensors, smart cameras. The omnidirectional Robot Hardware has also been tailored to facilitate the integration of peripherals based on Robot Operating System. Furthermore, robot activity can be monitored through an intuitive tool. Therefore, these studies can completely apply in industry and civil when robot problems are developing.

ACKNOWLEDGMENT

This study was supported by Department of Electrical Industry, Faculty of Electrical Engineering University of Economics - Technology for Industries, No. 456, Minh Khai Road, Hai Ba Trung, Ha Noi of Viet Nam; http://www.uneti.edu.vn.

REFERENCES

[1] M.Galli, R. Barber, S. Garrido, and L. Moreno, "*Path planning using Matlab-ROS integration applied to mobile robots*" in 2017 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), 2017, pp. 98-103: IEEE. "doi: 10.1109 / ICARSC. 2017. 7964059".

[2] I. Zamora, N. G. Lopez, V. M. Vilches, and A. H. Cordero, "*Extending the openai gym for robotics: a toolkit for reinforcement learning using ros and gazebo*" arXiv preprint arXiv:1608.05742, 2016.

[3] B. M. da Silva, R. S. Xavier, and L. M. Gonçalves, "*Mapping and Navigation for Indoor Robots under ROS: An Experimental Analysis*" Creative Commons CC BY license, 2019. "doi: 10.20944/preprints 201907.0035.v1.

[4] L. Zhi and M. Xuesong, "*Navigation and Control System of Mobile Robot Based on ROS*" in 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), pp. 368-372: IEEE, 2018.

[5] P. Marin-Plaza, A. Hussein, D. Martin, and A. d. l. Escalera, "*Global and local path planning study in a ros-based research platform for autonomous vehicles*" Journal of Advanced Transportation, p. 115 -123, vol 8. 2018.

[6] K. Zheng, "*ROS Navigation Tuning Guide*," arXiv preprint arXiv:1706.09068, 2017.

[7] M. Keller, F. Hoffmann, C. Hass, T. Bertram, and A. Seewald, "*Planning of optimal collision avoidance trajectories with timed elastic bands*" IFAC Proceedings Volumes vol. 47, no. 3, pp. 9822-9827, 2014. "doi: 10.3182/20140824-6-ZA-1003.01143".

[8] Do Quang Hiep, Ngo Manh Tien, Nguyen Manh Cuong, Pham Tien Dung, Tran Van Manh, Nguyen Tien Kiem, Nguyen Duc Duy, "*An Approach to Design Navigation System for Omnidirectional Mobile Robot Based on ROS*", (IJMERR); pp: 1502-1508, Volume 11; Issue 9; 2020.

[9] Roan Van Hoa, L. K. Lai, Le Thi Hoan, "Mobile Robot Navigation Using Deep Reinforcement Learning in Unknown Environments", SSRG International Journal of Electrical and Electronics Engineering (SSRG-IJEEE), pp. 15-20, Volume 7 Issue 8 , Aug 2020.

[10] V. Matt and N. Aran, "*Deep reinforcement learning approach to autonomous driving*," ed: arXiv, 2017.

[11] Nguyen Thanh Tuan, "*Base Deep learning*", The Legrand Orange Book. Version 2, last update, August 2020.

[12] Charu C. Aggarwal, "*Neural Networks and Deep Learning*", Springer International Publishing AG, part of Springer Nature, 2018.

[13] X. Ruan, D. Ren, X. Zhu and J. Huang, "*Mobile Robot Navigation based on Deep Reinforcement Learning*", Chinese Control And Decision Conference (CCDC), 2019.

[14] Andrea Bacciotti**,** *Stability and Control of Linear Systems***,** Publishing Ltd; Springer Nature Switzerland AG, 2019.

[15] R. Weideman, *Robot Navigation in cluttered environments with deep reinforcement learning*, Thesis, Faculty of California Polytechnic State University, June 2019.

[16] Francis X. Govers, "*Artificial Intelligence for Robotics"*, Published by Packt Publishing Ltd. Livery Place. 35 Livery Street, Birmingham, B3 2PB, UK, 2018.