2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference(IAEAC 2018)

Navigation and Control System of Mobile Robot Based on ROS

Li Zhi[1]
1. Institute of Robotics & Intelligent Systems, Xi'an Jiao Tong University
2. Shaanxi Key Laboratory of Intelligent Robots
Xi'an, China
437158160@qq.com

Mei Xuesong[2]
1. Institute of Robotics & Intelligent Systems, Xi'an Jiao Tong University
2. Shaanxi Key Laboratory of Intelligent Robots
Xi'an, China
xsmei@xjtu.edu.cn

***Abstract*—Control system and navigation are the most concerned topics in autonomous mobile robots. In this essay, control system based on STM32 and ROS (robot operating system) will be introduced, which contains hardware circuit designing, control software and upper computer software. In lower computer, we have realized velocity and current control of DC motors. While some other tasks such as posture calculation, sensor data transfer is also complete in lower computer. The upper computer platform is based on ROS (robot operating system), which is an open system for developers of robot. SLAM (simultaneous localization and mapping) and autonomous navigation have been realized on ROS with the data of laser scan.**

Keywords—autonomous mobile robot; control system; controller; navigation; SLAM; ROS;localization; autonomous movement; laser scan

I. INTRODUCTION

With the development of computer science and robotics today, robots have received extensive attention. As a branch area of robot, mobile robot is widely used in many cases, such as industrial transport, logistics, mobile operation, etc. [1][2] However, there are few open mobile platforms in the market. The precision, stability and ability of second development of them are unsatisfactory, either. While the technology of autonomous movement used most frequently today is line-tracking. Line-tracking is an easy and cheap way to solve this problem. However, it is not intelligent, and not environmental adaptative.

For mobile robot, control system is the most important part, which decide the performance of movement, such as precision and stability. On the other hand, autonomous navigation is one of the indispensable functions of mobile robots. These two aspects are what we want to focus on in this essay.

In this essay, we have designed our own mobile robot as Fig.1. shows. This is a differential drive robot. The front, rear movements and turning movements of the robot are realized by controlling the rotation speed of the left and right wheels.

We have researched and developed an open control system for mobile robots, which contains servo driver and controller. In the article below the architecture of the control system will be introduced.

Meanwhile we have developed upper computer software to realize human-computer interaction, remote control, remote monitoring.

SLAM is a basic part for navigation, which is used to build the map of the environment. In this essay, simultaneous localization and mapping will be finished on the bases of ROS. We used gmapping package to build the map with the data of laser scan, encoder and IMU.

For mobile robot, autonomous movement is the most concerned topic. In this essay, localization, navigation and autonomous move of the robot are all realized with the data of laser scan and the map we've already built. This function is finished by means of navigation stack in ROS.

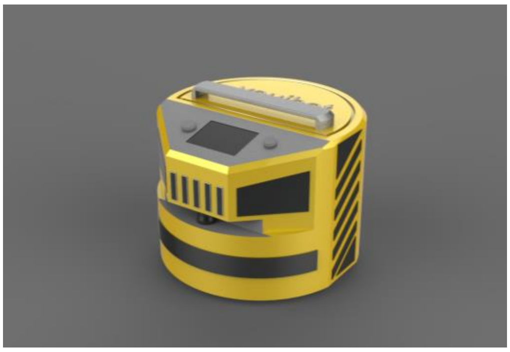

Fig. 1. Mobile robot

II. ARCHITECTURE OF THE CONTROL SYSTEM

A. Hardware Architecture

The robot control system is as the Fig.2. shows. The upper computer based on ROS sends control instructions to the lower computer (main controller) through the USB bus. Main controller controls the DC motor by servo driver when it receives control instructions. DC motor driver is used to complete DC motor servo drive task by sampling the current and encoder signals of the DC motor.

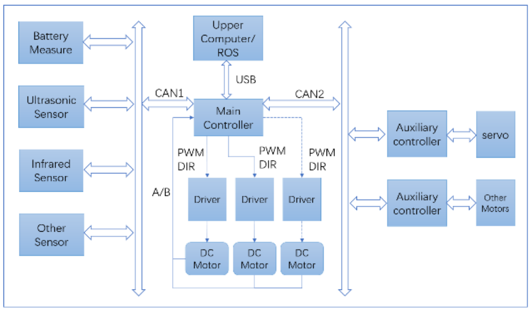

Fig. 2. Hardware architecture of the control system of mobile robot

In addition, main controller is also used to complete the task of calculating the pose of robot using the data of the IMU module on it. Main controller can also communicate with other sensor modules and auxiliary controller through CAN bus, which makes it very easy to extend the control and sensor system. Meanwhile, we can separate a very complex system into many easy modules, which make it easy to maintain or modify the system.

The PCB board of the main controller is as Fig.3. shows. The main control chip of the board is STM32F405.It is a kind of high performance and low power waste chip. We now have integrated power module, IMU module, current sampling module, voltage measuring module on it.

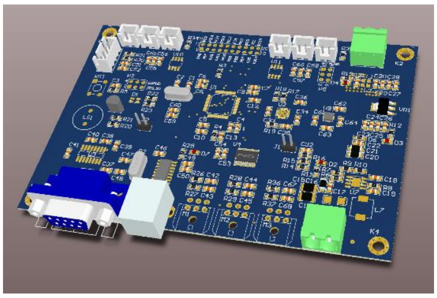

Fig. 3. PCB board of the main contoller

The PCB board of the motor driver is as Fig.4. shows. It can drive both brush and brushless DC motor using power MOSFET. The max current of the drive is about 6A limited by temperature, which can be applied for a wide range of mobile robot. Current loop, velocity loop are all realize in servo driver. It can communicate with main controller through serial bus or signal of PWM and direction.

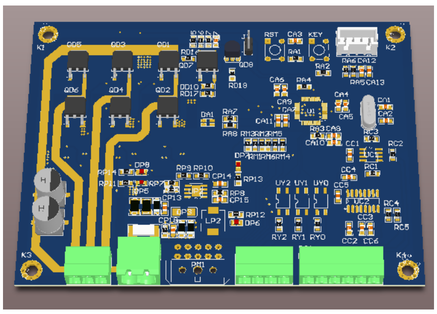

Fig. 4. PCB board of the driver of DC motors

B. *Software Architecture*

Software discussed here is mainly about lower computer, that is, main controller. Lower computer software is based on FreeRTOS. The complex software is divided into several tasks through RTOS scheduling, containing motor control task, IMU calculating task, data transfer task, voltage measuring task, etc.

Some main part of the software block diagram is as Fig.5. shows. At the begin, the program creates some processes. Each process complete one task. Motor control task controls the speed of DC motor at a constant value. The control algorithm of it is PID control. IMU task use the data of MPU6050 to calculate the pose of the robot by means of Kalman filter. Meanwhile data transfer task receives or transmits data from upper computer.

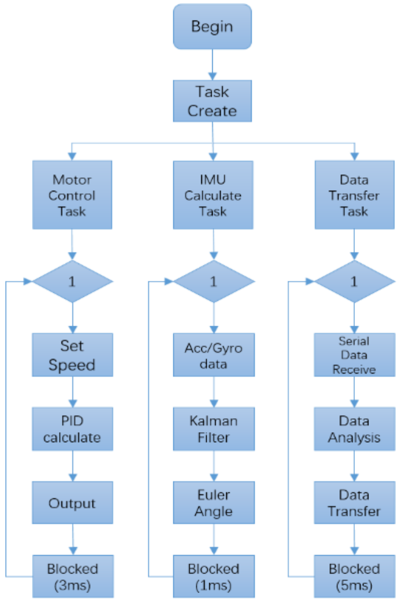

Fig. 5. Block diagram of part of the program

Motor control method of the robot is speed-current loop control, as Fig.6. shows. Control algorithms we used is PID control, which is widely used in most industrial scenes [3][4]. The output of the speed loop is the input of current loop. The feedback control structure is adopted in all two loops. Although its dynamic tracking performance is not very high, it is enough for mobile robots.

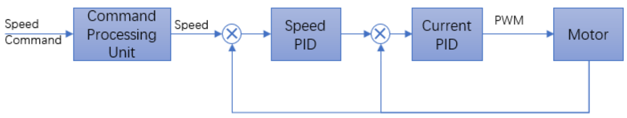

Fig. 6. Block diagram of control algorithm

As a result, our controller has achieved good performance, and each functional module has realized the basic functions. The robot is now moving smoothly.

III. UPPRER COMPUTER SOFTWARE BASED ON ROS

The software of the upper computer is based on ROS (Robot Operating System). It is mainly used for human-machine interaction, remote control and data communication.

ROS now is the leading development environment in robotics [5]. ROS provides libraries and tools to help software developers create robot applications [6]. It provides hardware abstraction, device drivers, libraries, visualizers, message-passing, package management, and more. ROS is a distributed process (that is, the "node") framework, which is encapsulated in functional packages that are easy to be shared and released. ROS provides us with a complete set of communication mechanisms for robots to liberate us from tedious programming.

As Fig.7. shows is a typical graph to describe ROS software. There are some nodes and topics in the graph. Each node, such as /turtlesim node will finish some tasks of robot, while it can communicate with other nodes by means of topic, such as topic /turtle1/cmd_vel. A node can publish or subscribe messages on topics. So many nodes can get together to finish the whole function of the robot. Thus we call ROS distributed robot system. As a result, we use ROS to integrate our host computer code.

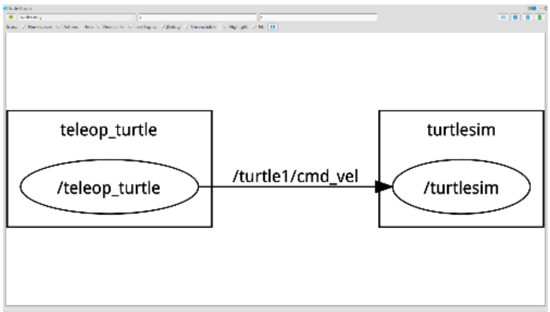

Fig. 7. Description of ROS program

A part of the UI of the robot is as Fig.8. shows, which is built on PyQt and ROS. It can show the speed, position or other status of the robot. While we can get the image data real time with the camera of the robot. And we can push the buttons on it to do some tasks such as control or mapping.

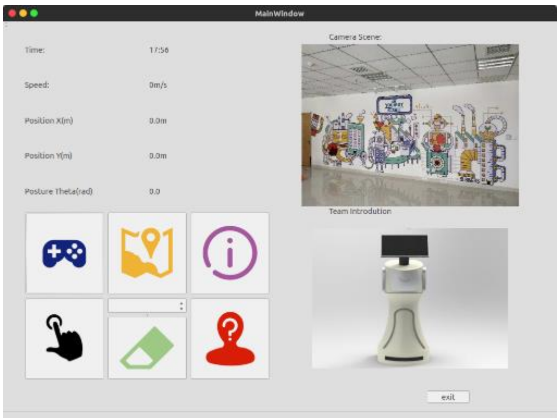

Fig. 8. UI of the Robot

IV. LASER RADAR SLAM BASED ON ROS

Autonomous mobile technology can make mobile robots more intelligent. It has important significance for mobile robots. More and more developers begin to integrate this kind of technology in to mobile robot. SLAM (simultaneous localization and mapping) is a basic problem of it, which can be described as follows. In the unknown environment, the robot begins to move from an unknown position. In the process of moving, the robot is positioned according to the position estimation and map. At the same time, an incremental map is built on the bases of its own positioning, which can be used to realize the autonomous localization and navigation of the robot.

Commonly, sensors like laser-scanners, sonar-sensors and cameras are used most for building the map of the environment [7]. In our robot, we use the laser scan to get the message of the environment. As Fig.9. shows is the laser scan we used. Laser scan can detect the location of target, environment or obstacles by transmitting laser beams.

Fig. 9. Laser scan

We have realized SLAM using the data of the scan laser. The math description of SLAM problems can be regarded as the posterior probability estimation of map and robot pose with laser measurement data and robot pose control data which is already known. To solve this problem, researchers have already done many works. Methods such as EKF, PF are already used to localize the position of the robot and achieved good results. In our robot, we use the method of PF to solve this problem, which is already integrated in the package in Gmapping of ROS.

To use the package of Gmapping, we need provide point cloud data from the laser scan, odometry data calculated from encoder of DC motors of the base and tf transform from laser scan to the base of robot.

Tf transform data shows the position and posture between two objects, which can be described as position vector and quaternion. ROS has already integrated this kind format of data.

Odometry shows the position and posture of the robot from where it begins. As Fig.10. shows, odometry can be calculated by the formulas below:

$$X = \int dX = \int (v_x \cos\theta - v_y \sin\theta) dt \tag{1}$$

$$Y = \int dY = \int (v_x \sin\theta + v_y \cos\theta) dt \tag{2}$$

$$\theta = \int d\theta = \int \omega dt \tag{3}$$

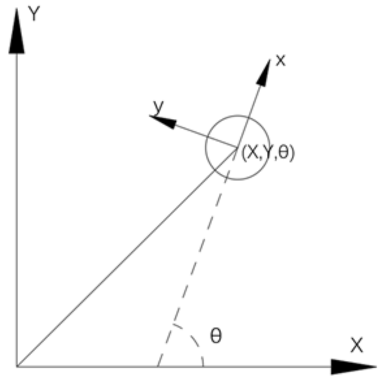

Fig. 10. Odometry calculate figure

When we get the odometry data of the robot, we get a rough estimate of the position and posture. This estimate data is not so accurate cause the wheel may skid with the ground. In the procedure of SLAM, we will get a more accurate position estimate when combining odometry data with laser scan data.

As a result, we get incremental map data of our office as Fig.11. shows. There are many other algorithms for 2D-SLAM, such as Hector Slam, CRSM SLAM, etc. [8] Each method has its own characteristics, but this is not the focus of our discussion, so we do not elaborate too much.

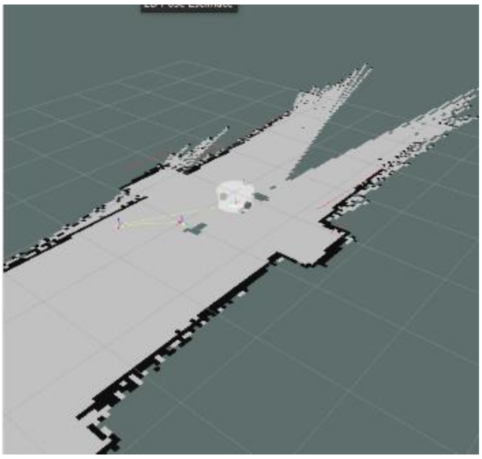

Fig. 11. Incremenrtal map

V. NAVIGATION BASED ON ROS

Navigation pack is one of the most powerful features of ROS system. Autonomous navigation and motion can be realized by it. And this pack can only be used on differential drive robot or omnidirectional movement robot.

The framework of navigation stack is as Fig.12. shows. In navigation stack, amcl (adaptive Monte Carlo localization) is a localization method based on Monte Carlo. It is realized by means of particle filter. The idea of particle filtering is to scatter particles randomly in the map space. When the robot moves, particles will also move with robot. In each calculation period, the higher the matching degree of the scan laser data of a position to the environmental map, the particles in the position will gain a higher weight. The particles in the map will be resampling according to the weight, and the particles in the map will gradually converge to a point, then this is the maximum likelihood estimation of the robot's position. We need provide odometry data and laser scan data when we call this pack.

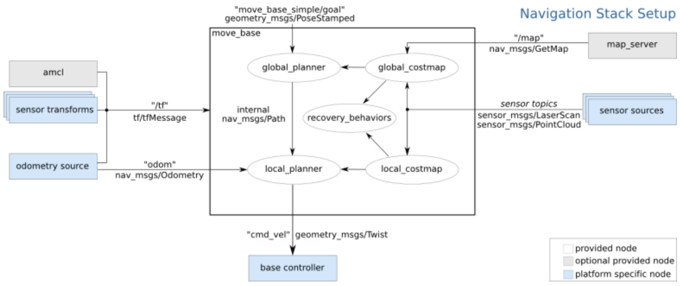

Fig. 12. Navigation stack

As Fig.13. shows, we can give a rough estimate of the position. Particles at first is Dispersed. After a certain period of movement, the particles gradually converge to the vicinity of the robot as Fig.14. shows.

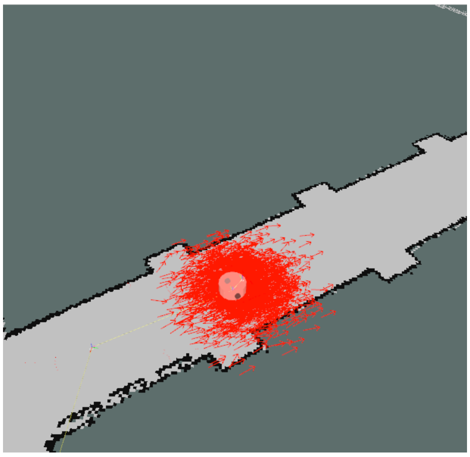

Fig. 13. Particle localization

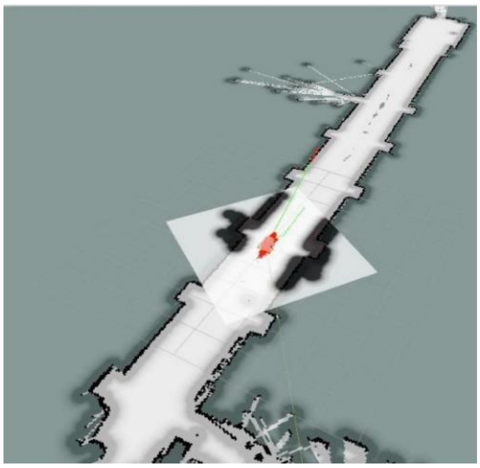

Fig. 14. Particles localization and global path planning

Path planning refers to the process of finding the shortest path based on environment and obstacles between the starting point and the terminal point. After getting the position of the robot in map and the position which robot want to arrive, move_base pack will do global path planning in global costmap. The path planning algorithm has been well developed so far. Algorithm like A*, RRT are well tested to solve path planning problem [9]. Some optimal algorithms, such as GA, AI, PSO, are also used for path planning [10].

When the robot meets obstacles on the way from starting point to the ending point, move_base pack will put obstacles into local costmap and replan the path in local costmap. The output of the move_base pack is the speed of the robot. Then we send speed commands to the base of robot to drive the robot.

At last, we realized navigation just as Fig.15. shows.

Fig. 15. Localization and navigation experiment

VI. CONCLUSIONS

In this paper, we developed and researched on auto mobile robot. The whole control system is developed and has been applied to the control of mobile robots and has achieved good control performance. Main controller, DC motor driver have been detailed designed.

Laser scan SLAM is realized in this paper on the robot we designed. We use the data of IMU, encoder of the motors to calculate the odometry value. By means of gmapping we finally get the incremental map of the environment.

At last we have finished localization and navigation of the robot in the map we built. We use the method of particles filter to get the estimate of localization. And we use move_base in ROS to plan the path. Finally, the robot arrived at the goal point.

REFERENCES

[1] M. Köseoğlu, O. M. Çelik and Ö. Pektaş, "Design of an autonomous mobile robot based on ROS," 2017 International Artificial Intelligence and Data Processing Symposium (IDAP), Malatya, 2017, pp. 1-5.

[2] R. C. Arkin and R. R. Murphy, "Autonomous navigation in a manufacturing environment," in IEEE Transactions on Robotics and Automation, vol. 6, no. 4, pp. 445-454, Aug 1990.

[3] J. Pongfai and W. Assawinchaichote, "Self-tuning PID parameters using NN-GA for brush DC motor control system," 2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Phuket, 2017, pp. 111-114.

[4] M. Xu, H. Zhang and H. Tang, "Design of motion control system for robot car based on DSP," 2017 29th Chinese Control And Decision Conference (CCDC), Chongqing, 2017, pp. 7494-7497.

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

[6] K. Takaya, T. Asai, V. Kroumov and F. Smarandache, "Simulation environment for mobile robots testing using ROS and Gazebo," 2016 20th International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, 2016, pp. 96-101.

[7] P. Fritsche and B. Wagner, "Comparison of two radar-based scanning-techniques for the use in robotic mapping," 2015 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO), Colmar, Alsace, France, 2015, pp. 365-372.

[8] M. Rojas-Fernández, D. Mújica-Vargas, M. Matuz-Cruz and D. López-Borreguero, "Performance comparison of 2D SLAM techniques available in ROS using a differential drive robot," 2018 International Conference on Electronics, Communications and Computers (CONIELECOMP), Cholula, 2018, pp. 50-58.

[9] Y. Chen, X. Wang, S. Hong, X. Zhong and C. Zou, "Motion planning implemented in ROS for mobile robot," 2017 29th Chinese Control And Decision Conference (CCDC), Chongqing, 2017, pp. 7149-7154.

[10] Qian Zhang, Ming Li and Xuesong Wang, "Global path planning method of mobile robot in uncertain environment," 2010 Chinese Control and Decision Conference, Xuzhou, 2010, pp. 4320-4324.