2017 2nd International Conference on Advanced Robotics and Mechatronics (ICARM)

Indoor Intelligent Patrol Robot Based on ROS Architecture

Zhihua He[1], Lei Cheng[1], Wenlei Zheng[1], Mengxia Sun[1], Qiuyue Yu[1]

1. School of Information Science and Engineering, Wuhan University of Science and Technology, Wuhan, 430081
E-mail: 1017353632@qq.com
E-mail: chenglei@wust.edu.cn
E-mail: 463796572@qq.com
E-mail: 1121119811@qq.com
E-mail: fanny_yqy@163.com

Abstract: In the robot development, different robot developers have to realize their specific software architecture and hardware implementation, which leads to the complexity of the system that different developers design is not the same, different architecture can make a big difference . If developers want has other functions on the basis of the original robot, it is difficult to achieve rapid transfer of software, and in the design of the system, need to do more cumbersome, repetitive work module etc.. This makes the development cycle may be very long. ROS provides a similar combination of Observer mode and Mediator mode communication architecture, which can meet most scene of the robots use, and ROS has a fairly large node code base, the use of ROS can greatly reduce the workload of the robot system design, reduce duplication developers, improve code reuse rate. In this article, the paper introduces the algorithm of particle filter, and then puts forward a design method of indoor intelligent patrol robot based on ROS communication architecture. By using the navigation function package of ROS, the autonomous navigation of indoor intelligent patrol robot can be realized, and the navigation process can be carried out with escaping obstacles in time , intelligence is very obvious in the progress . Experiments show that this method is fast and efficient, and it is a very efficient indoor intelligent patrol robot design method.

Key Words: Communication architecture, particle filter, autonomous navigation, intelligent patrol robot

1 INTRODUCTION

With the development of the economic and social, large-scale hotels, shopping malls, logistics and warehouses and other large-scale flow of people, the number of indoor intelligent patrol robot demand is increasingly urgent [1]. At present, mobile robots have been greatly developed [2-3]. Multi - sensor information fusion technology, such as Bayesian estimation, Kalman filter, particle filter, etc . Positioning and navigation technology, including global and local path planning; intelligent control technology, such as neural network control and so have a great development. Indoor patrol robot is a kind of new service robot which has just begun to show out the head angle in recent years, which can realize the function of environment perception, route planning and dynamic decision. The use of the robot will greatly save manpower and material resources, improve work efficiency, and the study of service-oriented robot is of great significance. At present, Europe and the United States patrol robot research has made great progress, but mainly for the military field; In addition, some developed countries also use patrol robots to carry out security work in the transport hub, storage base and so on. The United States and Japan have the most in-depth study, and MDARS-E robot which comes from the United States can identify obstacles around, when not around the time , It will stop and inform the master . He can be on duty when on the self-monitoring, you can find abnormal circumstances in a timely manner alarm . China is also beginning to conduct research in this area, and also made some progress in the path planning of the mobile robot, but in the area of the patrol robot , achievements are less available. It would be a very difficult thing to design a robot from the bottom to start. The navigation of the mobile robot has been a greater development [4-5]. But autonomous navigation is of the most difficulty and focus of patrolling robot technology. With the help of ROS [6-7], it can not only achieve the purpose of fast construction of robot platform, but also complete the task of autonomous navigation and obstacle avoidance real-time of indoor intelligent patrol robot. The related technology is analyzed in this paper.

This work is supported by the Science and Technology Innovation Fund for College Students of Wuhan University of Science and Technology(15ZZB145), the Zhejiang Open Foundation of the Most Important Subjects, and Hubei Province Science and Technology Support Project under Grant 2015BAA018

2 INTRODUCTION OF ROS

2.1 The origin and development of robot operating system

The maximum allowed number of pages is 6.ROS was originally released by Willow Gardge in 2010, formerly known as Stanford Artificial Intelligence Labs to support Stanford Smart Robot STAIR. ROS is open source, is a kind of post-operating system for robots, or secondary operating system, can provide similar operating system functions. ROS and linux precision related tasks scheduling and CPU resource allocation and other work to rely on linux kernel to achieve. The well-designed nodes in the ROS can be executed separately and loosely coupled at runtime to achieve decoupling between functional modules. In addition, ROS file system of the organization

978-1-5386-3260-4/17/$31.00 ©2017 IEEE

to greatly improve the ROS code reusability, from around the world ROS community is to provide a strong ROS code support, greatly improving the ROS developer development efficiency.

2.2 The main features of ROS

ROS is an open source software framework, itself consists of a large number of node code library, through the ROS communication module to achieve the loosely coupled network connection between the various modules, node-topic communication is the core of its communication, it ensures that the various modules Loose distribution. ROS is now supported in many languages to achieve the interface, the most commonly used is C + + and Python, in addition to Octave and LISP and so on. The code in each module in ROS can be compiled using CMake alone, which makes the software testing process simple. In addition ROS provides a large number of gadgets and users to interact, this function is done by the kernel. ROS draws on a lot of mature open source project code, such as Player project driver, motion control and simulation aspects of the code, OpenCV in the visual aspects of the algorithm. ROS is open source and free of charge, and the source code is publicly available and approved by BSD, allowing a variety of commercial and non-commercial projects to be modified and developed.

2.3 Concepts related to the ROS

ROS system has three levels of concepts: file system level, computer graphics level, community level [8].

File system level analysis of the ROS file structure and the work of the core documents required, and then introduce the basic components and their respective functions. ROS (Package) is the organization of software in ROS, including the executable file, relying on the library configuration file, ROS software is a software package or other part of the package to create a program. The function pack list (Manifest) provides information about the licensing, dependency, compilation flags, etc. of the functional package. A functional package (Stack) is a collection of feature specific functional packages. Function package list provides information on the functional package, license information, dependencies, and other information compiled, is essentially a manifests.xml file, through this document to achieve the management of the functional package. The functional package list (Stack manifest) contains a list of functional packages, license information, dependencies with other feature sets, etc.. The concept of file system level is shown in Figure 1.

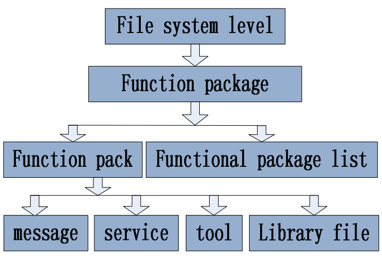

Fig 1. Concept of ROS file system level

A graph is a kind of point to point network of ROS processing data [9]. Under the unified scheduling of node manager ROS Master, the communication among nodes can be realized. Node is an executable program, which can communicate with other nodes through the topic. Message is the data type of the ROS used when publishing or subscribing. Topic can make the corresponding node in the release of this message, you can also subscribe to a topic to receive messages. The node manager (Master) monitors all node subscriptions and releases messages. The most important mechanism for the communication between ROS nodes is message passing. In ROS, news organizations are stored in the topic, when a node wants to share information, it will publish message to one or more of the corresponding topic; when a node wants to receive information, it will subscribe one or more topics it needs. ROS node manager is responsible for ensuring that the release of nodes and subscription nodes to find each other, and the message is released directly from the node transfer to the subscription node, and not through the intermediate node manager to ROS communication between nodes, as shown in figure 2.

ROS community level is a form of expression for the code distribution on the network , each community under the file structure of the file system level file organization, ROS community around the world, by different countries ROS community organizations to develop And maintenance, the ROS community can provide a wide range of open source and free resources for the ROS. At the same time, the new results of the research organization can be uploaded to it and widely available so that users can get the global code.

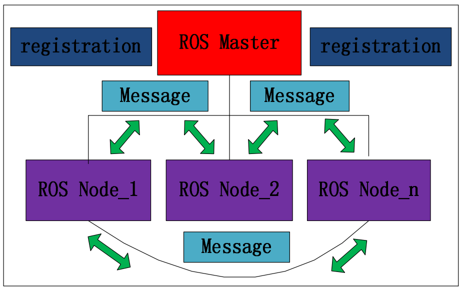

Fig 2. Communication between nodes of ROS

3 PARTICLE FILTER ALGORITHM

3.1 Location algorithm overview

The adaptive Monte Carlo localization in this experiment (AMCL) robot localization algorithm, probability and statistics of mobile robot can be completed under the 2D environment, using ROS platform using particle filter algorithm [10-11] based on the map prior to tracking of robot pose to.

The particle filter is a new filtering algorithm is developed in the last century in the late 90s, the basic idea is to use a random sample to describe the probability distribution, and then on the basis of measurement, the sample size and adjust the position of the particle weights, to approximate the actual probability distribution, and system estimation to sample as the mean value, effectively overcomes the shortcomings of the extended Calman filter. However, there are some weak points in itself, and the computation of particle filter is larger. However, with the increasing of computer processing ability, the limitation of the hardware capability of the early particle filter application is disappearing. The day before, the particle filter algorithm has been widely used in target tracking, navigation and guidance, fault diagnosis, parameter estimation and system identification etc..

The particle filter is by looking for a random sample spread on $p(x)$ in state space to approximate the probability density function of $p(x_k \mid z_k)$, instead of integral to the sample mean to obtain the process state minimum variance estimation, the sample is called "particle". The language of mathematics is described as follows, for a stationary random process, assuming that k-1 moment system posterior probability density for $p(x_{k-1} \mid z_{k-1})$, according to a certain principle for selecting the N random sample, K time measurement information, and time after the state update process, N particles of the posterior probability density is approximately $p(x_k \mid z_k)$. With the increase of the number of particles, the probability density function of the particle is gradually approaching the probability density function of the state.

3.2 Algorithm implementation

The essence of particle filter is that the integral operation is changed into a sum of finite sample points, that is, the state probability density distribution can be expressed as follows:

$$P(x_{0:k} \mid z_{1:k}) = \frac{1}{N}\sum_{i=1}^{N} \delta x_k^i (dx_{0:k})$$

Type $z_{1:k} = \{z_1, z_2, \ldots z_k\}$ is to measure the time k set, z_k is a measure of time that the K value of x_k^i is k, said moment obtained from the probability density function of the I particles, N particle is the amount of $\delta(\cdot)$ is the Dirac function, $P(x_{0:k} \mid z_{1:k})$ represents the probability density of Z observation sequences X. Usually it is difficult to sample directly from $P(x_{0:k} \mid z_{1:k})$. An effective solution is to introduce a well known probability density distribution function $q(x_{0:k} \mid z_{1:k})$, which is a very important problem. $q(x_{0:k} \mid z_{1:k})$ extracted from the N with the weights of the particles, when the new observation data, real-time update the weights of each particle. With the increase of time, the distribution of importance weights becomes more and more inclined. In order to avoid the lack of particles, Gordon and other resampling methods are proposed. The main idea is to remove the particles with small weights, and copy the particles with large weights. Finally, the updated probability density function can be expressed as:

$$P(x_k \mid z_{1:k}) = \sum_{i=1}^{N} \omega_k^i \delta(x_k - x_k^i)$$

Specific steps are as follows:

Step 1: Initialize $k = 0$, sampling $x_0^i \sim P(x_0)$, that is, according to the $P(x_0)$ distribution sampling to get

x_0^i , $i = 1,2,\ldots N$

Step 2: Importance weight calculation. Sampling

$$x_k^i \sim q(x_k \mid x_{0:k-1}^i, z_{0:k}) , \quad i = 1,2,\ldots N ,$$

calculate the importance of weight as follows:

$$\omega_k^i = \omega_{k-1}^i \frac{P(z_k \mid x_k^i) P(x_k^i \mid x_{k-1}^i)}{q(x_k^i \mid x_{0:k-1}^i, z_{0:k})} , \quad i = 1,2,\ldots N$$

Normalized importance weights:

$$\tilde{\omega} = \omega_k^i \Big/ \sum_{i=1}^{N} \omega_k^i$$

Step 3: resampling

According to the $\tilde{\omega}_k^i$ of the $x_k^i, i = 1,2,\ldots N$ set, a new set of N particles is obtained by resampling the importance weights, and the weights of the redistributed particles are as follows:

$$\omega_k^i = \tilde{\omega}_k^i = 1/N$$

Step 4: output

The state estimates are as follows:

$$\tilde{x} = \sum_{i=1}^{N} \omega_k^i \tilde{x}_k^i$$

Variance estimates are as follows:

$$P_k = \sum_{i=1}^{N} \omega_k^i (\tilde{x}_k - \hat{x}_k)(\tilde{x}_k^i - \hat{x}_k)^T$$

4 INDOOR PATROL ROBOT BASED ON ROS ARCHITECTURE

4.1 ros navigation overview

ROS navigation metapackage provides a Gmapping based 2D slam method, figure 3 is its main architecture.

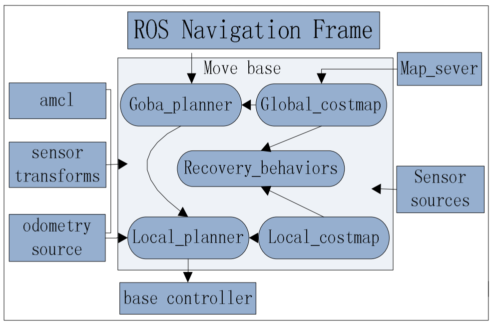

Fig 3. Communication architecture ROS navigation

As shown in Figure 3, we can clearly see the navigation input and output, including input odometer information odometry, sensor_topics or rgbd-camera laser radar, also known a priori map, coordinate transformation information of TF tree, one of the most important is TF tree and sensor data, the output speed is cmd_vel.

Move_base module is responsible for the scheduling of the entire navigation behavior, including the initialization of costmap and planner, monitoring the navigation state, timely replacement of navigation strategies, etc., it is the logical core of navigation. It implements a action interface for the upper node call, given the coordinates of the information and the corresponding coordinates of the location, the implementation of the corresponding security navigation. For the lower level controller, the output is cmd_vel 2D, which specifies the behavior of the navigation. Global Planner is a global planner navigation, accept the costmap generated global costmap and planning from the starting point to the target point of the overall path planning for local_planner to make reference. Local_planner is a local planner navigation, accept the costmap generated local costmap and planning speed. Costmap implements the environment representation of weighted grid map. As an environment representation module in the navigation architecture, it takes on the task of interacting with the sensor data stream and exposing the interface to the planner, which is very important for the accuracy of path planning. Costmap_2d main input from lidar Laserscan or point cloud pointcloud, from TF tree to obtain positioning data, and then in the case of known location data composition.

In ROS navigation, move_base provides a framework in the move_base is specified by the nav_core planner and recovery_behavior base class interface call. Separate from the specific implementation methods. The specific method used by the pluginlib according to different parameters into. This method makes the navigation can greatly increase the customization. Like base_local_planner in the realization of the two local path planning method, global_planner to achieve the A* and Dijkstra two methods, there are more navigation_experimental in this realization. This gives great flexibility to the framework. Navigation can be adapted to many different tasks by different configuration methods.

4.2 Hardware platform

In this experiment, the host computer to run the Ubuntu12.04 operating system PC, the main solution to the sensor data, and the calculation of the speed information released to the lower machine motion control node subscription topic.

Fig 4. URG-04LX laser sensor

The movement controller using STM32 motion control node contains subscription rate information topic, then PC solution after a given speed is sent to the motion controller, controller to achieve a given output, while the encoder data back to the host computer, and realize the closed-loop speed regulation from PID, the host computer and lower computer using RS232 serial communication, communication mode / slave mode, PC for the host, the motion control card from the machine, with a question and answer mode. The moving chassis is made of omnidirectional chassis, which can move more freely by driving omnidirectional wheel. Laser sensor using Hokuyo's URG-04LX, as shown in Figure 4, through the laser sensor to complete the robot composition.

The hardware part of the lower computer system mainly includes power supply system, motor drive circuit and so on. The overall hardware design is shown in figure 5.

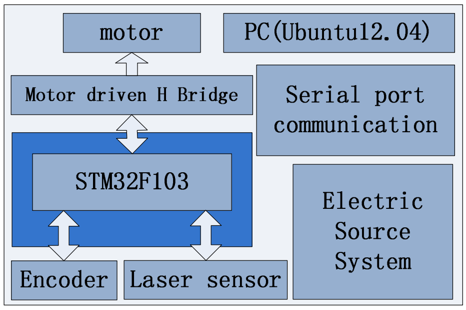

Fig 5. Hardware design of mobile robot

4.3 Software implementation

This project uses ROS as the software architecture of mobile robot, which is supported by ROS (Ubuntu12.04). Indoor robot navigation is mainly used to the Navigation function package, in the ROS Matser node under the unified scheduling to achieve indoor navigation function, in navigation need to use the three packages. Wherein, the move_base packet is programmed according to the reference message from the odometer and the laser, and sent to the bottom layer to make the mobile robot reach the designated position;The gmapping package builds maps based on laser data (or laser data from depth data);AMCL package has been based on some maps to locate, AMCL is a mobile robot in the two-dimensional environment of the probability positioning system. It implements an adaptive Monte Carlo localization method, which includes the use of a particle filter to track the attitude of a robot for an existing map.

The software design is shown in Figure 6, you can see through the navigation solution after the speed information output to the "cmd_vel" topic, news of the geometry_msgs/Twist data type, the data type that is 3D space velocity, the lower machine received from the host machine speed and real-time information with more output.

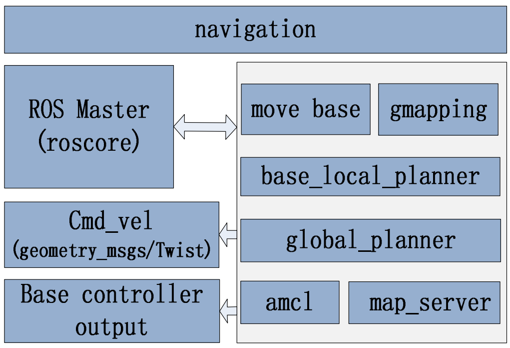

Figure 6. Software design of mobile robot

4.4 Rviz platform simulation

ROS has a powerful visual 3D simulation tool Rviz. Using a priori map and a robot model that comes with Rviz. In the simulation before the need to use the laser radar for composition, and then use the Rviz 2D Nav Goal button to specify any coordinates on the map point, the robot can be accessed in turn. The simulation results are shown in Figure 7.

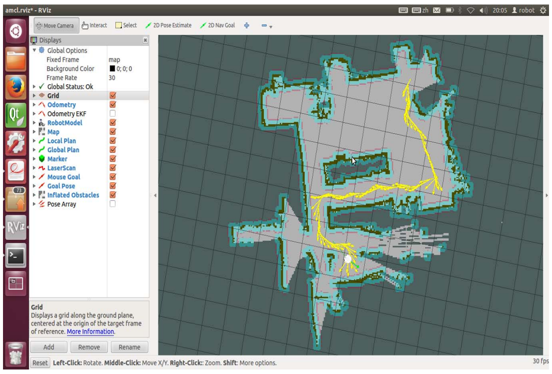

Figure 7. Rviz simulation results screenshot

4.5 Indoor patrol robot experiment

First, map the lab with a lidar. Select four coordinate points and record the coordinates, and then stored to the host computer. Mark the actual position of the four coordinate points just recorded. After the experiment began, traverse the four coordinate points, the experimental process shown in Figure 8.

Figure 8. Robot patrol experiment

The robot traverses four coordinate points in turn. Record the actual coordinates of the robot at these four left points. And then compare with the desired coordinates and make the difference. The results are shown in figure 9.

Target point	1	2	3	4
Accuracy difference	2cm	3cm	2.5cm	2cm

Figure 9. Robot positioning accuracy results

Through the experimental data analysis can know that this method can be completed more accurate robot positioning, navigation. Can meet the indoor patrol robot positioning accuracy requirements. And robots in the navigation process can escape obstacles. Above are the embodiment of robot intelligence.

5 CONCLUSION

In this paper, the development and characteristics of ROS are introduced briefly, secondly introduces the particle filter algorithm based on hardware based running the ROS platform and the software realization, and with the visualization tool Rviz, the simulation experiment of patrol robot indoor set point, the physical experiment in intelligent robot platform. Experiments show that the robot ROS communication architecture can well complete indoor patrol robot task based on the use of ROS is greatly reduced the workload of the robot system design, reduce the repeated work of developers, improve code reuse rate, the robot related project development or verification of the algorithm will have important significance for researchers.

REFERENCES

[1] Chen Huazhi, Xie cunxi, Zeng Dehuai. Research Status and Application Prospect of Patrol Robot. Mechanical and Electrical Engineering Technology. 2003, 32(6): 19-21

[2] Xu Guohua, Tan Ming. Current Situation and Trend of Mobile Robot. Electromechanical Technology & Application. 2001, 3: 7-14

[3] Peng Wengang, Peng Baolin, Liusheng. Research Status and Development Trend of Mobile Robot Navigation System. Electromechanical information. 2009, 36: 69-70

[4] Song Zhangjun. Research Status and Development Trend of Service Robot. Journal of Integrated Technology. 2012, 1(3): 1-9

[5] Wang Tianmiao,Tao Yong, Chen yang. Current Situation and Development Trend of Service Robot Technology. Science China Press. 2012, 42(9): 1049-1064

[6] Shi Xiaolu, zheng Yousheng, Zhang Lei. Intelligent step-by-step embedded motion control system based on ROS. Software engineering. 2016, 19(6): 48-51

[7] Bai Liangliang, Ping Xueliang, Chen Shenglong. Design and Implementation of Distributed Mobile Robot Control System. Machinery Design & Manufacture. 2015, 10: 180-183

[8] Zhang Jianwei. Open source robot operating system ROS [M]. Science Press, 2012

[9] Enrique Fernandez. Learning ROS for Robotics Programming. China Machine Press, 2014 (In Chinese).

[10] Guo xiaosong, Li Yipeng, Guo Junbin.Research on particle filter algorithm and its applications. Computer Engineering and Design. 2009, 30(9): 2264-2269

[11] Zhu Zhiyu. Particle Filtering Algorithm and Its Application. Science Press. 2010(In Chinese).