Design and Production of Autonomous Obstacle Avoidance Robot Based on ROS

Jialin Wang [1, a], Le Zhang [2, b], Kang Liu[3, c] and Chao Yang[4, d]

[1] Southwest Petroleum University, Chengdu 610500, China;

[2] Southwest Petroleum University, Chengdu 610500, China;

[3] Southwest Petroleum University, Chengdu 610500, China.

[4] Southwest Petroleum University, Chengdu 610500, China.

Abstract

With the rapid development of science and technology, people's demand for service-oriented robots is becoming stronger, which makes people pay more attention to the research of intelligent mobile robots[1]. The development of robotics is influenced by many factors. Among them, the establishment of environmental map, path planning and automatic obstacle avoidance are very important research contents. ROS can build a robot system to achieve various functions, effectively realize the hardware management of robots and improve the control efficiency. This design is based on ROS system, through cooperating with the omnidirectional motion robot chassis, to verify it in practice, so that it can complete the functions of map building, autonomous navigation, real-time obstacle avoidance and so on. The chassis of the robot uses the mainstream STM32 series microcontroller as the main control unit. The microcontroller realizes the speed PID control of the motor. It receives and parses the control instructions of the ROS system timely and effectively through the serial port, and converts the received data into the parameters of the robot through the algorithm analysis.

Keywords

ROS, omnidirectional mobile robot, autonomous obstacle avoidance, path planning, SLAM.

1. Introduction

Nowadays, it's a developing rapidly generation for robots, but there are also many uncertain conditions in the working environment, so it's an inevitable trend of developing for robots to monitor around in time and posit itself.

In recent years, many robot software system appears, especially ROS(Robot Operating System) which can bridge between software and hardware, own great compatibility with programming, perfect debug interface and simulation environment, and can search easily opening source for code. Compared with other software system, it can put different node into different computers, which can make every part run independently, so that reduce the running burden of the computer assembled by the robot and increase the stability of the system.

The technology of autonomous barrier-free robot is comprehensive and applicated broadly, which develops well in family, entertainment venues, security sites, disaster relief, military environment and so on, so many institutions at home and aboard pay much attention to it. The autonomous barrier-free robot which has great abilities to detecting environment and measuring based on ROS is increasing rapidly and developing diversely.

2. System design

The system block diagram is shown in Figure 1.1. The system mainly is contained by controllers which includes mainly PC with ROS and microcontroller, actuators including mainly DC gearbox motor and sensors including encoders for calculating the speed of every wheel, the photoelectric switch for measuring obstacles and the lidar.

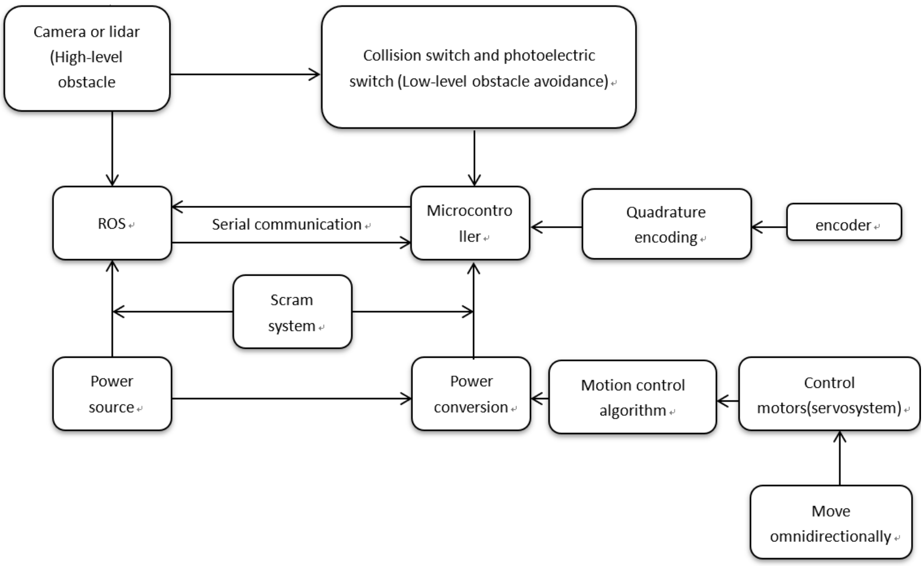

Figure 2.1 The system block diagram

The core of the system is ROS system and STM32 microcontroller. The system acquires the state of the robot by IMU, determines the boundaries of the environment or obstacles by the lidar, and then plan the way rationally. After receiving the information of controlling movement from the system, the microcontroller starts to control the robot moving. If meeting obstacle when moving, ROS starts to plan the new way for avoiding it.

3. The omnidirectional chassis design

3.1 Hardware composition and principle of omnidirectional chassis

The omnidirectional chassis schematic is shown as Figure 3.1. The chassis can finish the complex movement, which makes the robot avoid obstacles more smoothly and efficiently.

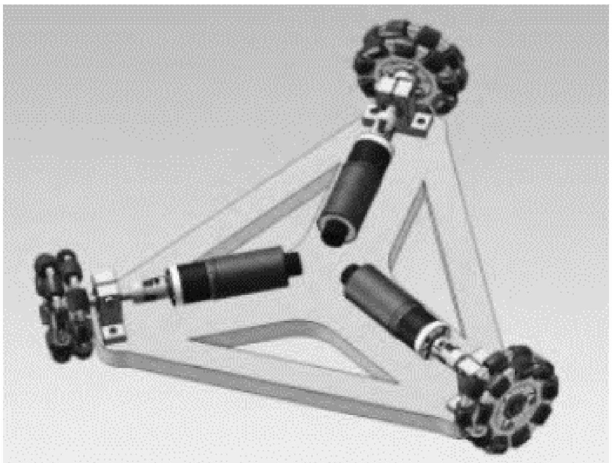

Figure 3.1 The omnidirectional chassis schematic

The angle between the wheels of three-wheel omnidirectional structure is 120 degrees. When the motors rotate, they provide the torque of tangential direction. Under the influence of three force, the robot can move omnidirectionally. The principle analysis for movement of three-wheel omnidirectional chassis is as follows:

Ideally, define the coordinate system of the environment as XOY, the coordinate system of the robot as $X'O'Y'$, the offset angle of the coordinate system of the robot from the environment as θ, the horizontal distance from the wheel to the center of the body as L, the rotation angular velocity of the robot as ω[2]. Under the coordinate system XOY of the environment, define the velocity component as Vx and V_y. Under the coordinate system of the robot as $X'O'Y'$, define the velocity component as Vx' and Vy', the rotation velocity of every wheel as V_A, V_B, V_C. The actual meaning of each quantity is shown in Figure 3.2.

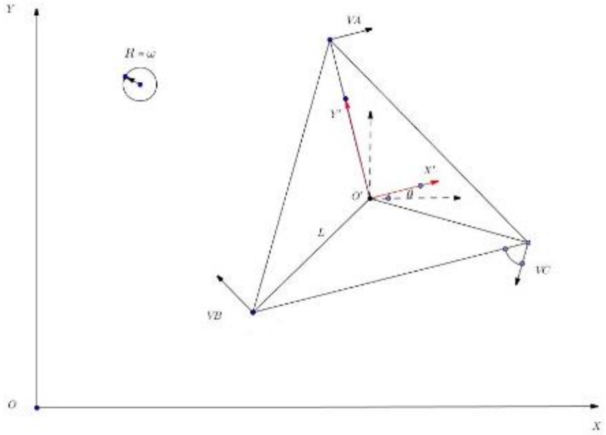

Figure 3.2 The actual meaning of each quantity

The conversion relationship between the coordinate system of the environment and the robot is as follows:

$$\begin{pmatrix} Vx \\ Vy \\ \omega \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} Vx' \\ Vy' \\ \omega \end{pmatrix} \tag{3-1}$$

The relationship between VA, VB and VC, and between Vx', Vy' and ω in $X'O'Y'$ is as follows:

$$\begin{pmatrix} VA \\ VB \\ VC \end{pmatrix} = \begin{pmatrix} 1 & 0 & L \\ -\cos 60° & \sin 60° & L \\ -\cos 60° & -\sin 60° & L \end{pmatrix} \begin{pmatrix} Vx' \\ Vy' \\ \omega \end{pmatrix} \tag{3-2}$$

By the conversion formula, the velocity of the robot can relate to the velocity of single wheel. Match the speed closed loop of the motors, the robot can finish move omnidirectionally.

3.2 Chassis control program design

The chassis control system schematic of the robot is in Figure 3.3. Ubuntu based on ROS sends data by USB-to-TTL interface. The microcontroller transforms the whole data receiving to control signal of the motor by the parsing function. The speed feedback of the motor is through the encoder on the motor, which in turn completes the speed closed loop.

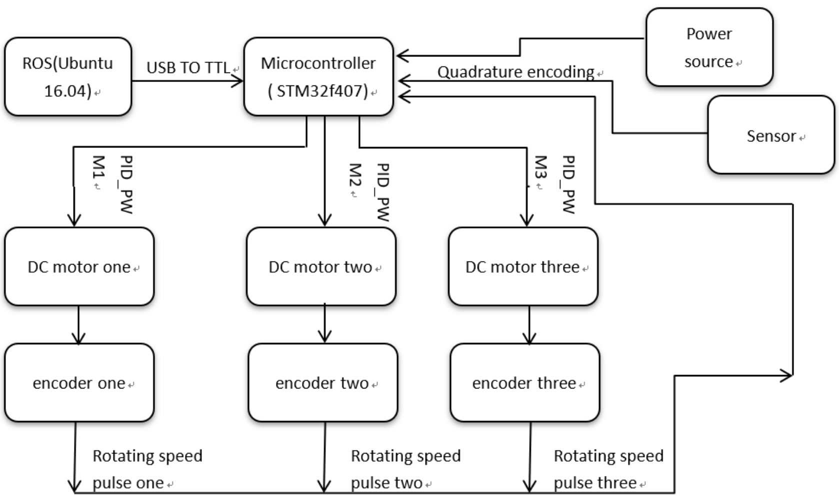

Figure 3.3 The chassis control system schematic of the robot

Microcontroller interact data with sensors by serial ports and IO ports. In the design, there're several duties for the microcontroller: control of the motor, omnidirectional motion algorithm, photoelectric detection of low-level barrier avoidance, serial communication, Dbus communication. The function for every duty is as follows:

Control of the motor: execute the motor's PID algorithm in a 20ms interruption, and monitor in time the speed parameters V_a,V_b,V_c.

Omnidirectional motion algorithm: monitor V_x,V_y, and convert them to the speed V_a,V_b,V_c of every wheel by motion algorithm.

Photoelectric detection of low-level barrier avoidance: refresh the detection environment of the sensor in time by photoelectric sensors, and after filtering finish simple barrier-avoidance function.

Serial communication: as the bridge between microcontroller and ROS host, single-chip port need to convert serial data receiving into motion parameters. ROS receives data from the sensor by USB-to-TTL interface, and runs algorithm.

Dbus communication: receive remote sensing signals, parse motion data, control chassis' movement, and prevent losing control while debugging ROS.

4. Navigation framework

4.1 Navigation prerequisites by ROS

ROS offers many functional packages, and there're open sources for each feature in wiki of ROS. Gmapping is used to posit and map, and Move_base is used to navigate and avoid barrier. After compiling, the program is encapsulated in ROS as a runnable node, which makes debugging and development easier.

The navigation package can be regarded as the algorithms for various sensor information. The robot needs to meet the following conditions for realizing navigating[3]:

The structure of the robot is two wheels with differential steering(non-complete system) or omnidirectional wheels (full system) and regular. Meanwhile, the robot can use Twist to control.

Post information for all joints and sensors.

Detect and post information for accelerated, linear and angular velocity, etc.

Acquire information about the environment to build the map and locate.

4.2 Mapping by ROS(SLAM)

Gmaping which is based on RBPF and builds the map and locate based on the lidar and IMU is used in the design. After realizing the function of every node, package configuration files and the function files as a launch file. When running, the robot can generate 2D maps for path planning and positioning.

4.3 Navigation and barrier avoidance by ROS

The navigation framework based on move_base is shown in Figure 4.1 including the global planner, local planner aimed at environmental mutations, and nodes aimed at helping autonomous navigation. Amcl and map_server are optional parts in the Figure. Others in the Figure are not in ROS, and can be realized by external conditions.

Path planner aims at selecting the shortest path from the start to the end. When the program is running, get the current and target position of the robot, move_base will work in costmap of ROS. When meeting the obstacles, move_base will put it into the local planner and replan the path. The outputs of move_base are the speed and the direction of the robot, and then send the speed to the chassis for driving the robot.

In the design, use above navigation framework to finish barrier-avoidance task.

Run a navigation barrier-avoidance demo in the rviz, when the robot works, it can deal with the obstacles as the expansion. After input the target position and angle, the robot can pass the obstacles without collision. When the obstacle appears suddenly, the robot avoid the obstacle as the refreshed way under the control of the local planner. The robot finishes path navigation and barrier avoidance which is shown in Figure 4.2.

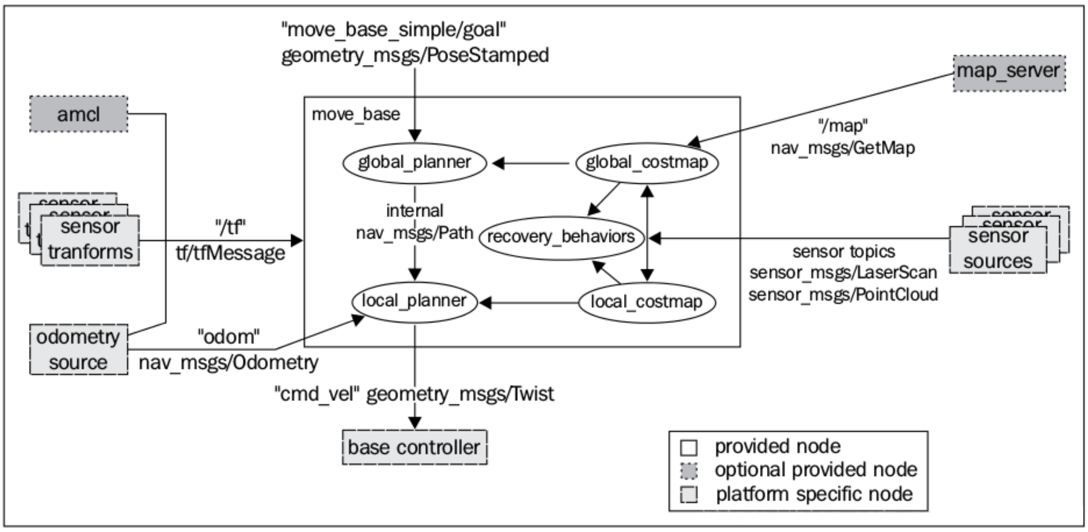

Figure 4.1 The navigation framework based on move_base

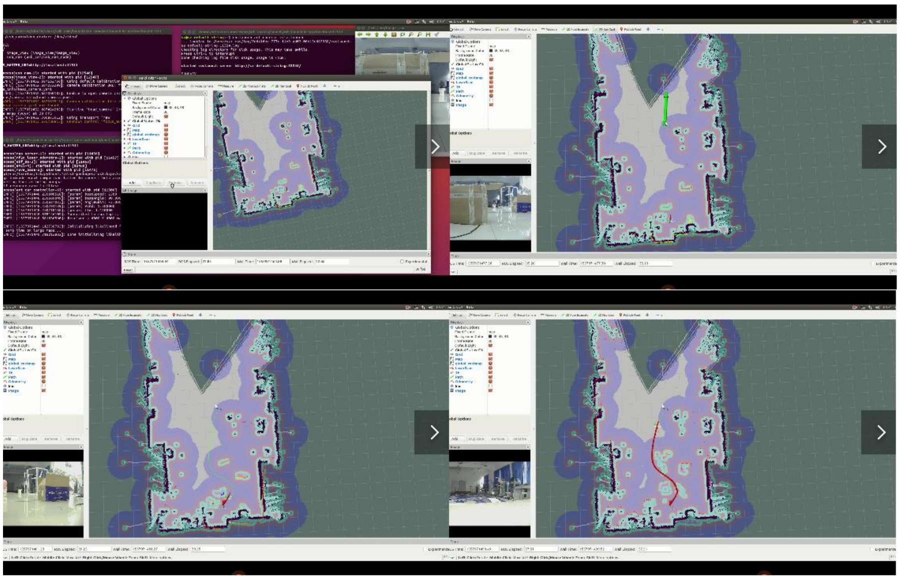

Figure 4.2 The effect of practical navigation and obstacle avoidance

5. Summary

The design based on three-wheel omnidirectional chassis of the omnidirectional wheel can monitor the control signal which can lead the robot moves omnidirectionally in the plane. Write the control program of the chassis and communication program serving ROS. For realizing navigation and avoiding obstacles, search the global path planning based on A-algorithm and the barrier-avoidance technology based on DWA local path planning.

Establish a platform for the mobile robot based on ROS, whose hardware including Ubuntu host, the lidar, IMU, the power supply, etc. In Ubuntu, design the framework of the software system. Use gmapping algorithm to build map, and write C++ code of the navigation system and assemble function pack, in order to have the functions of navigation and obstacle avoidance.

From the results of the experiment, the robot in the design can run smoothly and avoid unknown obstacles safely and quickly.

References

[1] Ye He. Research on the SLAM technology of mobile robot based on ROS[D]. Jiangxi: Nanchang University, 2018.
[2] Dan Hai. Design and Control of Omnidirectional Mobile Platforms[D]. University of National Defense Science and Technology, 2005.
[3] Zhenzhong Yu, Weicou Zheng, Xin Liu, etc. Local Path Planning for Mobile Robot in Time Based on Kinect[J]. Computer engineering, 2013, 39(4): 243-247.