The robot motion scheme

1. Manual remote control

The robot can be remote control by joystick or the graphic user interface(GUI) on the touch screen. (Figure 1). For the joystick control, instruction data is transmitted to the robot’s onboard chip via 2.4Ghz RF. There are two rockers on the joystick, which correspond to the plane linear velocity of the robot and the angular velocity of the wheel. With the cooperation of the two rockers, the robot can make complex movements in any directions on the plane. This way has good performance in offline environment because it’s simple and stable nature. When the network is allowed, users can control the robot through the GUI on the touch screen, as shown in (Figure 2). There are ten motion modes for users to choose, the button on the bottom is used to adjust the plane linear velocity and rotational angular velocity of the robot. One advantage of using the screen is that the user can observe the current speed feedback of the robot and the steering angle of each wheel in real time. The GUI scheme is suitable for use in situations where simple control is required but feedback is necessary.

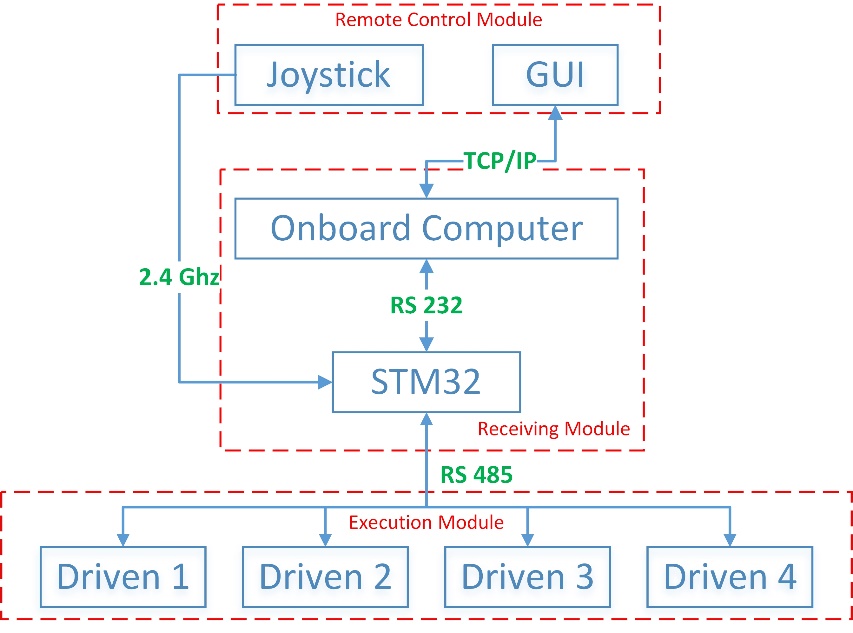


Figure 1 Framework of manual remote-control system

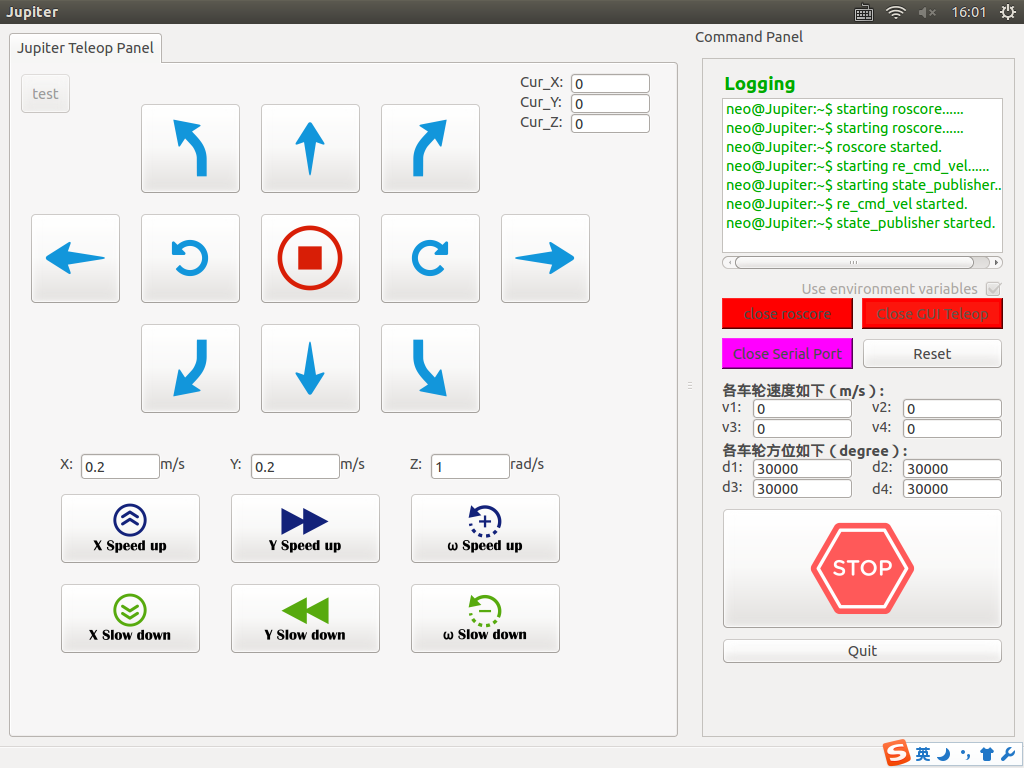


Figure 2 The graphic user interface(GUI) on the touch screen

1. Autonomous navigation

In certain circumstances, robot needs to act autonomously or inspect uninterruptedly, manual remote control is not the best option for these kinds of tasks. So, Jupiter is also designed to has the ability to navigate automatically. The flow of the navigation process is shown in (Figure 3), with the lasers mounted on the opposite corner of the robot, Jupiter can receive environmental information which can be used to locate and map. Considering that the laser used is high in frequency and there are no other sensors such as cameras and odometers installed, the SLAM algorithm used here is Hector SLAM. Laser data is utilized to update the current pose of the robot and the representation of the map, when the start and target points are determined, the Dijkstra algorithm will be used to generate a feasible path connecting two points. In the process of autonomous navigation, the dynamic obstacles will appear on the map in real time in form of occupied grids. Jupiter will use the Dynamic Window Approach to re-plan the feasible path to the target point based on the location and the size of the obstacles. The robot will stand still when it reaches the target point until the next target order is release.

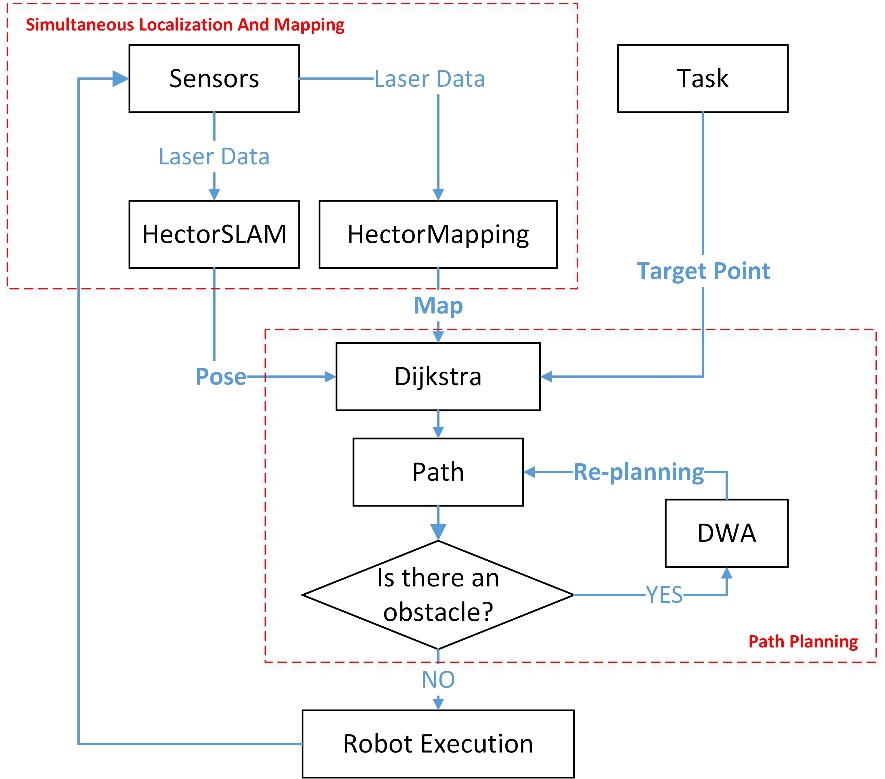


Figure 3 The flow of the navigation process