**机器人控制节点编写和仿真模拟**

**工作空间创建**

首先在目录下创建工作空间文件树，在命令行中输入catkin\_init\_workspace在新的工作空间内进行初始化操作。



图1 初始化工作空间

接下来使用catkin\_create\_pkg命令创建一个ROS包，将其命名为wanderbot，后面输入的参数为其依赖rospy、geometry\_msgs和sensor\_msgs。



图2 创建程序包

之后我们按照终端的提示在创建的ROS程序包内修改其配置文件package.xml，从图3中我们可以看出之前设置的编译和运行的依赖已经被包含在包中了。



图3 package.xml依赖项

其中我们使用到的geometry\_msgs中包含了控制机器人移动的消息类型Twist，sensor\_msgs中包含了机器人传感器中的红外激光测距的消息类型LaserScan，这个我们在之后启动节点时可以进行查看。

**仿真平台安装和使用**

接下来我们通过命令sudo apt-get install ros-indigo-turtlebot-gazebo安装gazebo仿真平台，然后使用命令

export TURTLEBOT\_GAZEBO\_WORLD\_FILE = "/opt/ros/indigo/share、turtlebot\_gazebo/worlds/playground.world"

设置gazebo平台仿真时使用的地图，然后通过roslaunch turtlebot\_gazebo turtlebot\_world.launch启动该ROS包中的节点，可以看到如下的图形界面。



图4 gazebo仿真平台

可以先使用ROS中提供的turtlebot\_teleop程序包用键盘对控制，命令为roslaunch turtlebot\_teleop keyboard\_teleop.launch。



图5 键盘控制说明

**传感器信息获取**

接下来我们编写一个节点获取传感器的信息，该脚本的内容为获取发布在/scan话题上LaserScan类型的消息（本质上是一个一维数组，表示每个角度上探测到的距物体的距离），取探测到其正前方的物体距离并输出。



图6 获取传感器信息脚本

效果如图所示。



图7 仿真平台效果

相应的传感器数据输出为：



图8 传感器输出

**消息分析**

接下来我们对gazebo仿真出的ROS节点上的话题进行分析，使用rostopic info命令分析和查看话题的消息类型、发布者和订阅者。



图9 话题信息查看

其中控制仿真平台中机器人的速度输入的是/cmd\_vel\_mux/input/teleop话题，消息类型是Twist，该类型包含如下6个参数，分别是：



图10 消息信息查看

分别用于控制turtlebot机器人底座的速度和角速度，此处使用了ROS对执行子系统（turtlebot机器人底座）的抽象和封装，便于程序员对其进行控制。

**控制程序编写**

直到了这些话题的相关参数，我们就可以编写程序，使机器人进行简单的的自主移动和避障，该脚本通过启动节点和gazebo平台的ROS节点利用消息进行交互，实现传感器发现障碍物后调整角度进行前进。





图 11 控制程序代码

效果如下：





图12 仿真避障效果