

轮式机器人1-4周实验报告

3200102888 米博宇

第一周

实验内容

使用socket包实现程序间通信。分别编写sender和receiver程序，实现在sender端发送数据，在receiver端输出的效果。

```
1  #receiver
2  _ENDPOINT = ("127.0.0.1",10000) #端口
3  import socket
4  sock = socket.socket(socket.AF_INET,socket.SOCK_DGRAM) #创建socket，服务
   器间网络通信，数据报型
5  sock.setsockopt(socket.SOL_SOCKET,socket.SO_REUSEADDR,1) #设置超时时间1ms
6  sock.bind(_ENDPOINT) #关联端口号
7  data,addr = sock.recvfrom(65535) #最大接受65535字节
8  print(data.decode('utf-8'),addr) #解码data
9
10 #sender
11 sock.sendto(line.encode('utf8'),_ENDPOINT) #发送编码后的数据
```

实验效果：在两个终端中分别运行两程序，在sender端通过键盘/文件输入字符，在receiver端显示。

第二周

实验内容

- 使用pynput监听键盘输入
- 使用matplotlib绘图
- 使用numpy进行矩阵运算

使用pynput.keyboard监听键盘输入

```

1 from pynput.keyboard import Key, Listener
2 def on_press(key) #按下按键时触发
3 hasattr (key, "char") #普通按键
4 # 创建监听
5 with Listener(on_press=on_press) as listener:
6     listener.join()

```

使用matplotlib.pyplot绘图

```

1 import matplotlib.pyplot as plt
2 plt.plot([p3[0], p1[0]], [p3[1], p1[1]], 'k-') #画线
3 plt.plot(x_trac, y_trac, 'b--') #画两个列表对应的折线
4 plt.xlim(-10+x, 10+x) #画图范围

```

使用numpy进行矩阵运算

```

1 import numpy as np
2 T = np.array([math.cos(theta), -math.sin(theta), x],
               [math.sin(theta), math.cos(theta), y], [0, 0, 1])) #定义矩阵
3 p1 = np.matmul(T, p1_i) #矩阵乘法
4 ra = np.linalg.inv(a) #求逆矩阵
5 vr = np.reshape(vr, (3, 1)) #改变规模

```

线程控制

```

1 import threading
2 t = threading.Thread(target=receive)
3 t.start()

```

新建线程运行receive函数，同时运行绘图函数。

第三、四周

ros的使用

```

1 roscd #进入文件夹
2 roslaunch #根据launch文件启动节点
3 rosrn #运行程序
4 catkin_make #依次调用cmake和make

```

编写发布者和订阅者

固定的坐标变换 (world to map)

在launch文件中使用一行代码表示

```
<node pkg="tf" type="static_transform_publisher"
name="course_map_tf" args="0 0 0 0 0 world_base map 100" />
```

创建一个固定坐标变换程序

```
1 rospy.init_node('fixed_tf_broadcaster')
2 br = tf.TransformBroadcaster()
3 while not rospy.is_shutdown():
4     br.sendTransform((0.0, 0.0, 0.0),
5                     (0.0, 0.0, 0.0, 1.0)
6                     rospy.Time.now(),
7                     "world_base",
8                     "map")
9     rate.sleep()
```

并向launch文件添加

```
<node pkg="tf" type="fixed_tf_broadcaster.py"
name="broadcaster_fixed"/>
```

实验心得与建议

通过前四周实验，我对ros的工作原理、使用方式和机器人运动有了初步了解，学习了numpy, matplotlib等常用库的使用，对python的面向对象编程部分有了更多的了解。在实验内容中，我印象最深刻的是ros的软件包构建和管理机制。只需要编写一个launch文件即可避免繁琐的运行各个节点程序的过程。

我认为实验的难点在于ros的使用：ros文件结构较为复杂，命令较多。在前四周实验中，与message、topic相关的操作较少，因此我对此部分的掌握较为薄弱，仅仅停留在教程层面上。而教程所提供的指导往往难以较为全面，因此我应该多参考其他资料，掌握一定的调试程序能力。

此外，建议在每周的实验文档中添加一些必要函数的参数说明，我在实验3: 编写消息发布者和订阅者实验中，理解rospy.Publisher, rospy.Subscriber两个函数花费了大量时间，一些参数也只是与教程相同，对这些参数的选取原则知之甚少。如果在实验文档中有相关指导，我认为实验效率会得到较大提升。

最后，感谢助教gg的辛苦付出 😊