ros在实验室可以发挥作用的地方：

数据记录且可视化（rviz），可以尝试做主控（ros的通信机制），物理仿真（gazebo），激光雷达slam（cartographer）。

先大概说说整个ros上层框架吧。

如果把一个ros的工作空间workspace看作一个国家的话。当今世界都是互联互通的，没有哪个国家可以不依赖其他国家独立存在。同样的在我们自己的ros工作空间下，也需要依赖其他的依赖包才能正常工作比如基本的c++库Python库。

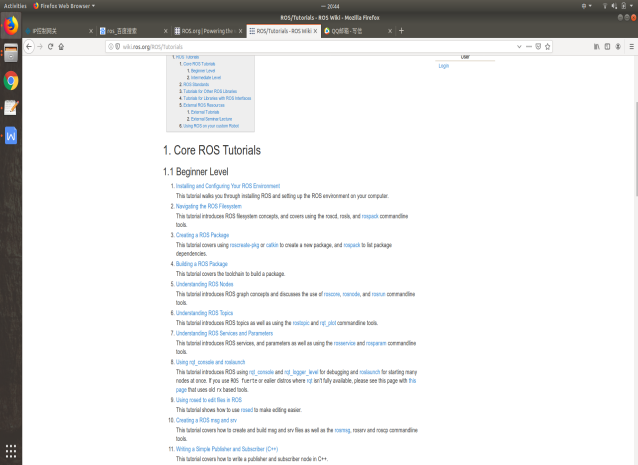
当我们要在ros上完成一项工程的话，就相当于当一个国家要实现一个5年计划。这一个5年计划可能包含很多不同方面的工程，比如造原子弹，载人航天，改革开放等等。而且这些工作在进行时，我们可能也会引进那些已经成熟的相关技术成果，通过消化吸收后再应用到我们自己的具体项目上。同样的，我们的工作完成后，也可以将我们成熟的技术成果再提供给别国使用，互利共享。共同推进这个技术的发展...在ros里面这就相当与是功能包的概念，这种共享的思想也是整个维持ros生态的关键和优势。

而当我们在做其中一个具体的工程时，比如载人航天工程，也就是编写ros工作空间下一个具体的功能包时。在做工程时，我们首先要分工，将一个大工程分成一个个小工程，分派给一个个专门的部门去做，这里的各个部门就是ros里面node节点的概念。

每个部门被分开来做专门的事情。每个部门在内部功能上是独立，同时每个部门之间也是需要合作的。那么这么多部门要有条不紊的工作，肯定需要有个专门的管理中心负责协调和信息的分发，以及一些相关规定。他们是这样工作的：每个部门把自己处理完的且别的部门需要的数据后，按需分类打包成一个个个文件，并且命名好送给管理者。这样别的部门就可以只对管理者说自己要什么名字的文件就能轻松的拿到想要的数据。这个就是ros里的话题通信机制，刚刚说的文件就是话题，管理者就是rosmaster。如果一个节点需要数据的话就先定义一个订阅者，指出要订阅的话题名字，每当有其他节点在这个话题上发布了消息，那么就会进入相应的处理函数中处理得到的数据。如果一个节点需要对外发送消息，先定义一个发布者，并且指出话题名，然后将处理好的数据打包好发布到rosmaster上。当然ros还有的通信机制还有服务，动作，类似就不一一详述了。

我刚初学的一些建议：

先看完官方文档：<http://wiki.ros.org/ROS/Tutorials>，和我上传的那些视频。初学的话只要看到urdf就够了，后面的其他功能等前面ros基础都理解了之后再看会轻松很对多。



一般刚开始学的话得全部跟着做完，对ros有个大概印象，同时一边学c++。刚开始学的时候最好不要细扣语法，和为什么。一般就是先看别人的注释的代码，和一些功能包的官方例程。因为这个ros编程很多东西都是模板型的，比如怎么和发布话题，刚开始学的时候就跟着写就行了，然后在模板下实现自己想要的功能。用着用着才能加深理解，否则光看语法，光看为什么，看完还是不会写。

刚开始用ros的时候，很不习惯：“怎么debug阿”程序跑的不对都不知道从哪找问题。我总结的几点查问题的方式：

1在程序里发标志位，通过ROS\_INFO打印到屏幕上，特别是程序编译不通过时或者怀疑有哪部分功能缺失时

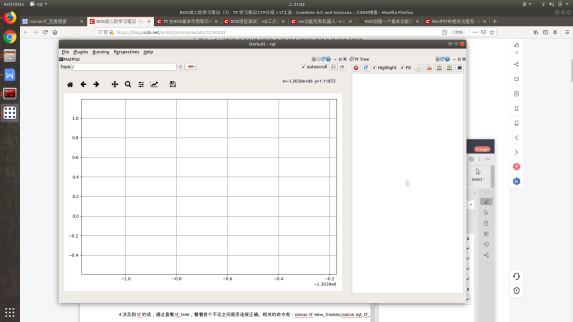
2通过rostopic list ，rostopic echo 话题名 ,查找当前发布在rosmaster的话内容是不是有问题

3通过rosrun rqt\_graph rqt\_graph查看当前各个节点之间通信关系是否正确，谁发布了什么话题，谁订阅了什么话题。谁和谁是否建立了联系，谁是否被启动了

4涉及到tf的话，通过查看tf\_tree，看看各个不见之间是否连接正确。相关的命令有：rosrun tf view\_frames ; rosrun rqt\_tf\_tree rosrun rqt\_tf\_tree

5 ros有一个专用的调试工具rqt ，命令行就是rqt。里面有很多插件plugins可以用来可视化数据。

6 rosbag可以用来记录运行数据，方便回放当时运行情况



对于ros功能包的使用我想特别说一下，我第一次接触ros就是用一个和四足机器人有关的库。一上来就看源码，linux刚接触，ros的机制也刚学，c++那么高深，四足的相关理论又不是很懂。最终也没能看下去，ros学习也搁浅了。所以使用功能包时，我的建议是只看它对外的接口就行。一般我们用别人写好的包大多数源代码是不用我们理解的和修改的（除了一些针对不同对象的配置文件，而且也只需要修改一些封装好的参数即可），除非有特殊需求。因为作为一个成熟的功能包，大多数情况下里面就是个黑盒子，对于使用者就只要提供这个功能包想要的输入就行了。在使用的过程中，使用我上述的那些调试工具再慢慢理解这个功能包是怎么运行的，然后再深入看源码。比如tf功能包，我们只要把两个link之间的相对关系发布给tf就行了，他就能帮我们输出想要一些的相对坐标和姿态，初学的时候我们也没必要看tf是怎么实现的。再比如cartographer这个slam库，看官方文档里说需要odom，imu，scan的数据，我们按照他要的格式提供给他就行了，然后根据实际效果修改一些具体参数，之后再去深入研究内部。