

# ROS控制器仿真

## 编译和启动

解压后进入工作空间

```
cd controller_sim
```

编译

```
catkin_make
```

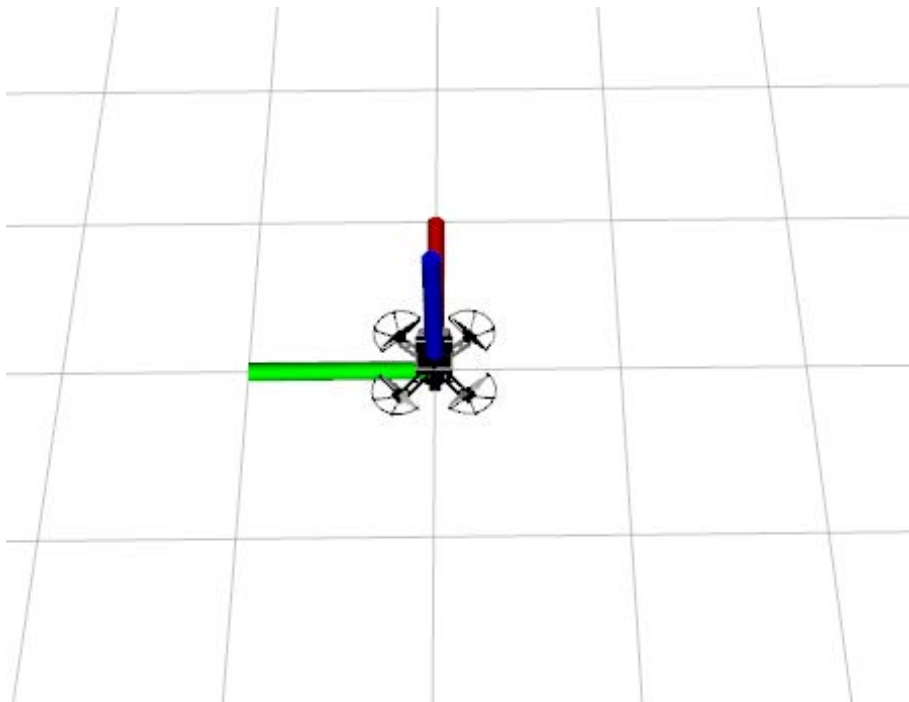
配置工作空间

```
source devel/setup.bash
```

运行程序

```
roslaunch so3_quadrotor_simulator simulator_example.launch
```

可见到一个飞机模型和坐标系。



## 编写控制器程序

需要编写的程序位于linear\_control.cpp的calculateControl函数，其中给定期望的状态des，当前无人机状态odom，惯性数据imu，以及在增益系数gain，需要计算控制指令u

```
void calculateControl(const Desired_State_t &des,
                    const Odom_Data_t &odom,
                    const Imu_Data_t &imu,
                    Controller_Output_t &u,
                    Gain gain);
```

此函数涉及的各种数据类型，可参见controller\_utils.h，如下图对Desired\_State\_t和Controller\_Output\_t两种结构进行了定义。Desired\_State\_t包含了位置、速度、加速度和加加速度，以及姿态的四元数。Controller\_Output\_t则包含姿态和推力。

```
struct Desired_State_t
{
    Eigen::Vector3d p;
    Eigen::Vector3d v;
    Eigen::Vector3d a;
    Eigen::Vector3d j;
    Eigen::Quaterniond q;

    Desired_State_t(){};
};

struct Controller_Output_t
{
    // Orientation of the body frame with respect to the world frame
    Eigen::Quaterniond q;

    // Collective mass normalized thrust
    double thrust;
};
```

\*给定的输入未必完全需要，按照自己的算法来实现即可。

## 如何调试

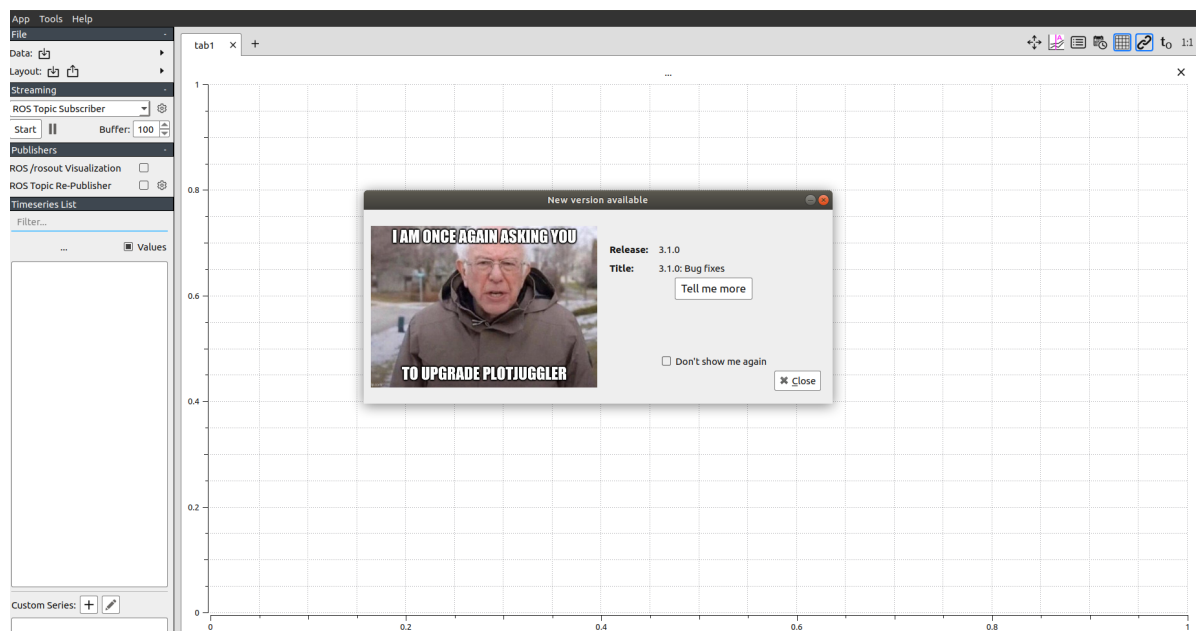
使用rosviz记录下无人机模拟器的里程计信息（推荐在roslaunch启动前先打开另一个终端开始录制）

```
rosviz record /sim/odom
```

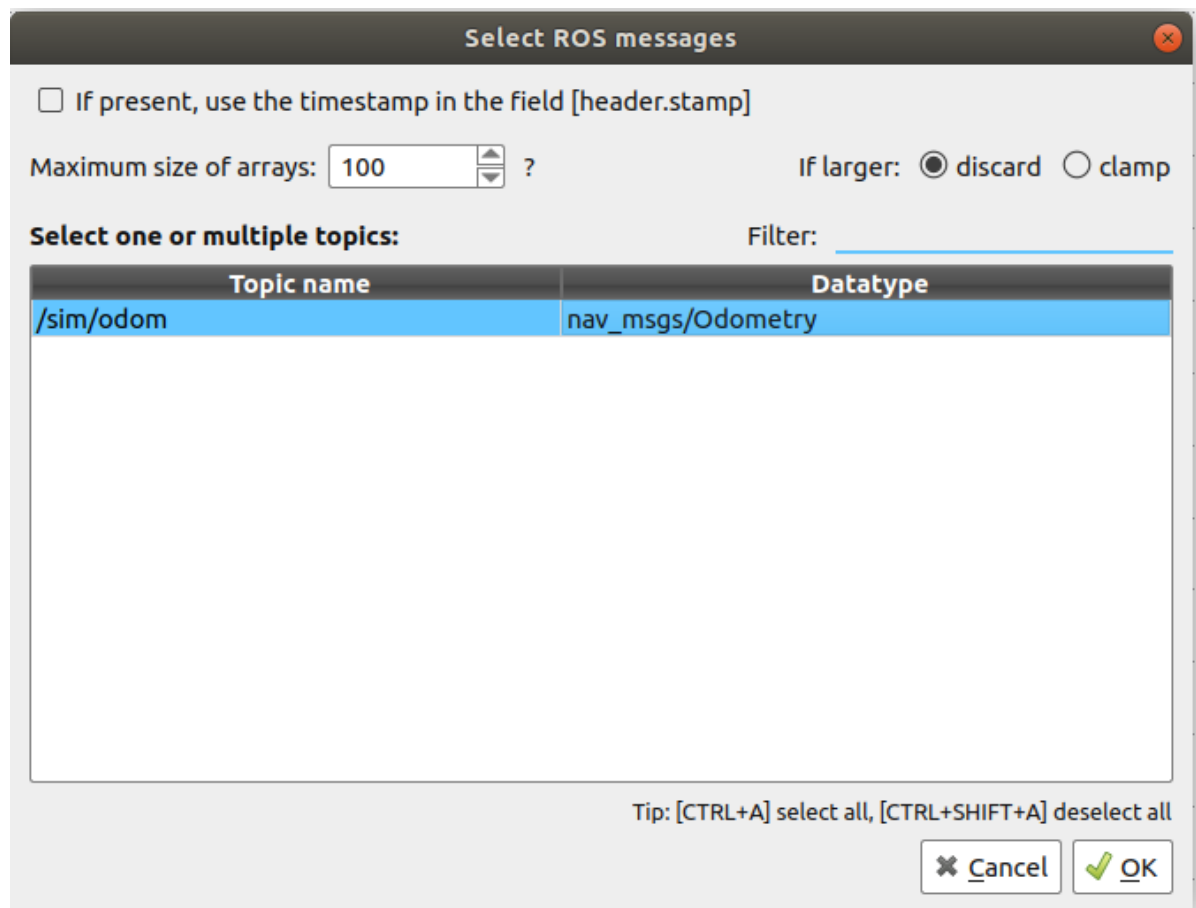
ctrl+c停止录制

打开plotjuggler软件（详见<https://github.com/facontidavide/Plotjuggler>）

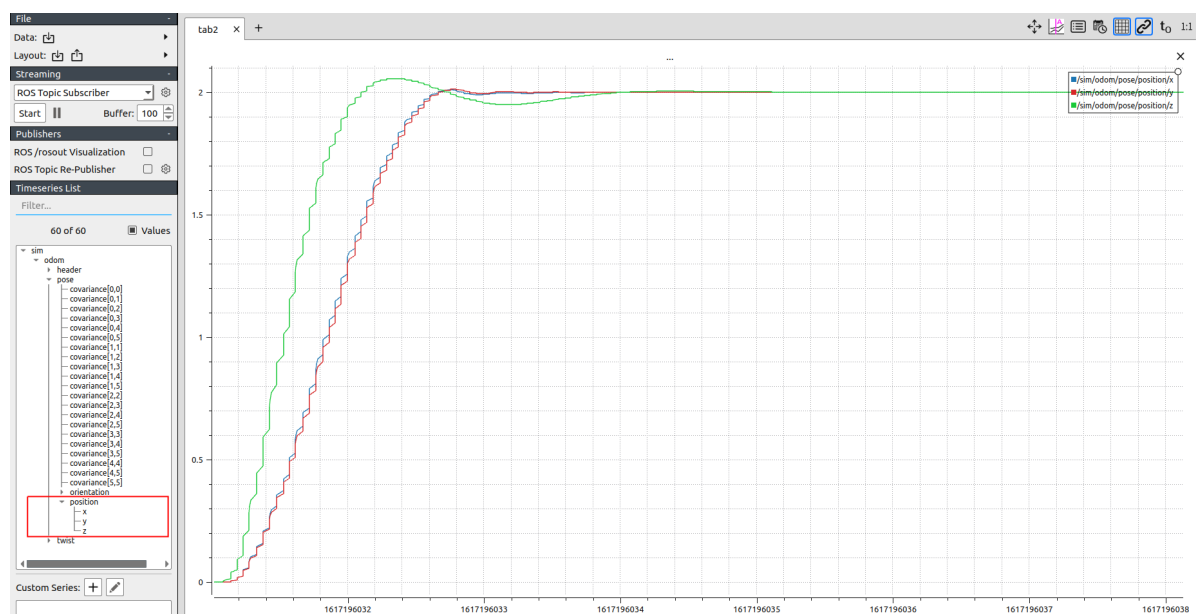
```
rosviz plotjuggler plotjuggler
```



不理睬图中的老头，选择左上角Data旁的标记打开刚才录制的rosviz



然后在上图中选择我们录制的topic，也就是/sim/odom，然后OK打开。



在左边列表中可打开记录下的topic的数值，选中/sim/odom/pose/position/x拖到右边的图上即可显示曲线。类似的，/sim/odom/pose/orientation可查看姿态，/sim/odom/twist/linear可查看线速度。

在仿真中让无人机换一个位置？

```
rostopic pub /position_cmd quadrotor_msgs/PositionCommand "header:
  seq: 0
  stamp: {secs: 0, nsecs: 0}
  frame_id: ''
  position: {x: -1.0, y: -1.0, z: 2.0}
  velocity: {x: 0.0, y: 0.0, z: 0.0}
  acceleration: {x: 0.0, y: 0.0, z: 0.0}
  jerk: {x: 0.0, y: 0.0, z: 0.0}
  yaw: 0.0
  yaw_dot: 0.0
  kx: [0.0, 0.0, 0.0]
  kv: [0.0, 0.0, 0.0]
  trajectory_id: 0
  trajectory_flag: 0"
```

修改上述position中x,y,z的值即可改变不同的位置。

## 如何修改增益数值

\*可以直接在LinearControl::calculateControl函数里给定参数，每次修改参数需要重新编译运行。

如果不编译直接修改数值，可以在roslaunch文件中修改，然后直接重新启动roslaunch即可。

原理是这样的：在controller\_sim/src/so3\_quadrotor\_simulator/launch中

```
<param name="gains/kx/x" value="1.0"/>
<param name="gains/kx/y" value="1.0"/>
<param name="gains/kx/z" value="1.0"/>
<param name="gains/kv/x" value="1.0"/>
<param name="gains/kv/y" value="1.0"/>
<param name="gains/kv/z" value="1.0"/>
```

通过roslaunch的参数传递，赋值给输入给LinearControl::calculateControl函数的gain变量，从而是实现参数传递。

controller\_sim/src/so3\_control/src/so3\_control\_nodelet.cpp Line203 - Line208

```
n.param("gains/kx/x", gain.Kp0, 5.7);  
n.param("gains/kx/y", gain.Kp1, 5.7);  
n.param("gains/kx/z", gain.Kp2, 6.2);  
n.param("gains/kv/x", gain.Kv0, 3.4);  
n.param("gains/kv/y", gain.Kv1, 3.4);  
n.param("gains/kv/z", gain.Kv2, 4.0);
```

上图的param（ros::NodeHandle::param）函数的用法是：根据第一个字符串变量在roslaunch中寻找匹配的变量名，如果找到就赋值给第二个变量，否则就使用第三个数字变量作为默认值。