

# 使用Allan方差法进行IMU标定

在IMU采集数据时，会产生两种误差：确定性误差和随机性误差，为获得精确的数据，需要对上述两种误差进行标定。

1、确定性误差

确定性误差主要包括bias(偏置)、scale(尺度)、misalignment(坐标轴互相不垂直)等多种。常使用六面静置法标定**加速度计**和**陀螺仪**的确定性误差。

2、随机误差

随机误差主要包括：高斯白噪声、bias随机游走。**加速度计**和**陀螺仪**随机误差的标定通常使用Allan方差法，Allan方差法是20世纪60年代由美国国家标准局的David Allan提出的基于时域的分析方法。

3、Allan方差图读取误差系数

Allan方差法可用于5种随机误差的标定：

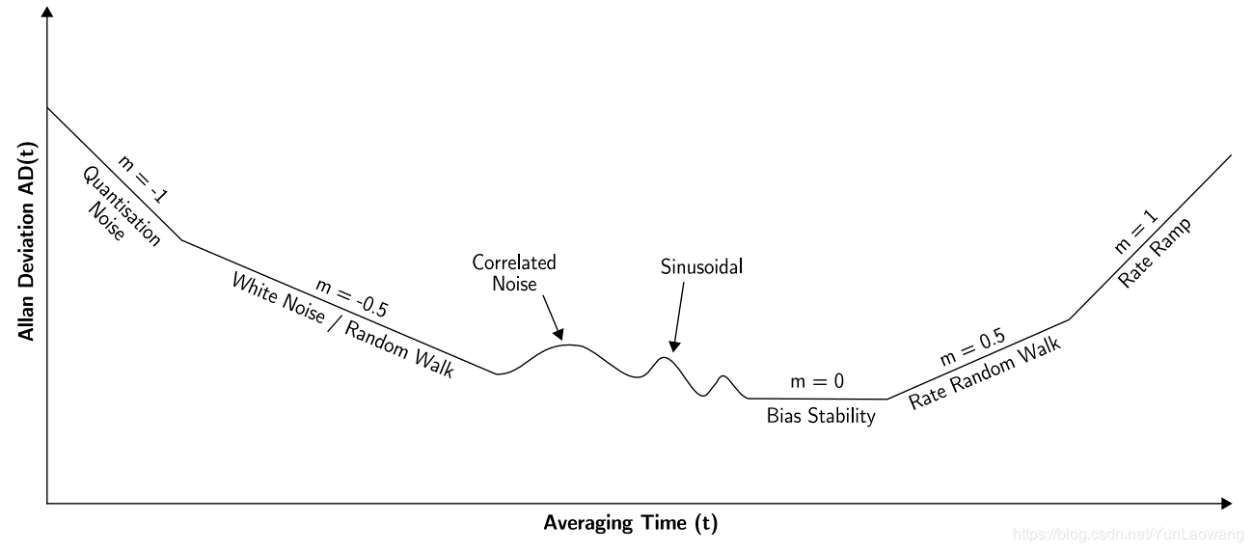
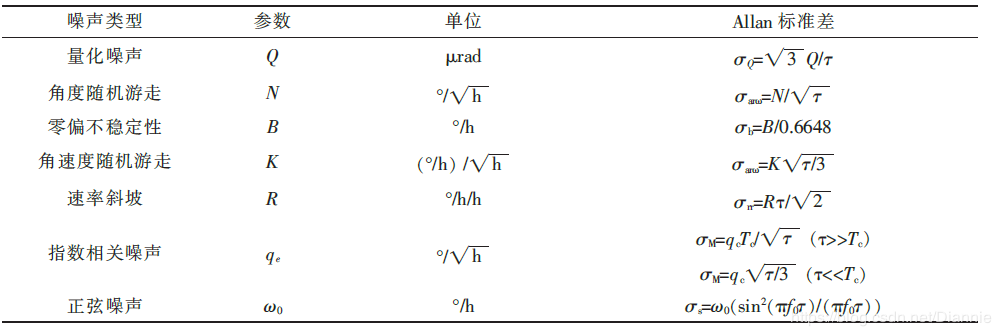
**量化噪声(Quantization Noise)**：误差系数为Q，Allan方差双对数曲线上斜率为-1的直线延长线与t=10°的交点的纵坐标读数为；

**角度随机游走(Angle Random Walk)**：其误差系数N，Allan方差双对数曲线上斜率为-1/2的线的延长线与t=10°交点的纵坐标读数即为N；

**零偏不稳定性(Bias Instability)**：其误差系数B，Allan方差双对数曲线上斜率为0的线的延长线与t=10°交点的纵坐标读数为，一般常取底部平坦区的最小值或取t=101或t=102 处的值；

**角速率随机游走(Rate Random Walk)**：其误差系数K，斜率为+1/2的线的延长线与t=10° 交点的纵坐标读数为；

**角速率斜坡(Rate Ramp)**：其误差系数R，斜率为+1的线的延长线与t=10°交点的纵坐标读数为；

IMU标定Allan方差工具

常用的Allan方差工具，主要有以下两种：

* https://github.com/gaowenliang/imu\_utils
* https://github.com/rpng/kalibr\_allan

## imu\_utils

使用Ubuntu 18.04 + ros-melodic-desktop-full

1. 安装ROS

初始化catkin工作空间

* mkdir –p ~/catkin\_ws/src
* cd ~/catkin\_ws/src
* catkin\_init\_workspace //初始化工作空间
* cd ..
* catkin\_make
* source devel/setup.bash //设置ros环境

2、安装ceres-solver

注意：<https://github.com/ceres-solver/ceres-solver.git>，编译之前需要安装一些基础库：

* sudo apt-get install -y liblapack-dev libsuitesparse-dev libcxsparse3 libgflags-dev libgoogle-glog-dev libgtest-dev

3、编译code\_utils

在catkin工作空间中：

* cd ~/catkin\_ws/src
* git clone https://github.com/gaowenliang/code\_utils.git
* cd ~/catkin\_ws
* catkin\_make

在编译之前需要对源码进行修改两种方法：

1. 在~/catkin\_ws/src/code\_utils/CMakeLists.txt中，添加：include\_directories("include/code\_utils")；
2. 修改~/catkin\_ws/src/code\_utils/src/sumpixel\_test.cpp文件中的#include "backward.hpp"为#include "；

4、编译imu\_utils

* cd ~/catkin\_ws/srcgit clone https://github.com/gaowenliang/imu\_utils.git
* cd ~/catkin\_ws
* catkin\_make

5、生成imu.bag

* roscore
* source devel/setup.bash
* rosrun vio\_data\_simulation vio\_data\_simulation\_node

默认位置在～/根目录中

6、写launch文件

进入 catkin\_ws/src/imu\_utils/launch文件夹，新建imu.launch文件：

*<launch>*

*<node pkg="imu\_utils" type="imu\_an" name="imu\_an" output="screen">*

*<param name="imu\_topic" type="string" value= "/imu"/>*

*<param name="imu\_name" type="string" value= "imutest"/>*

*<param name="data\_save\_path" type="string" value= "$(find*

*imu\_utils)/data/"/>*

*<param name="max\_time\_min" type="int" value= "120"/>*

*<param name="max\_cluster" type="int" value= "100"/>*

*</node>*

*</launch>*

7、重新编译：

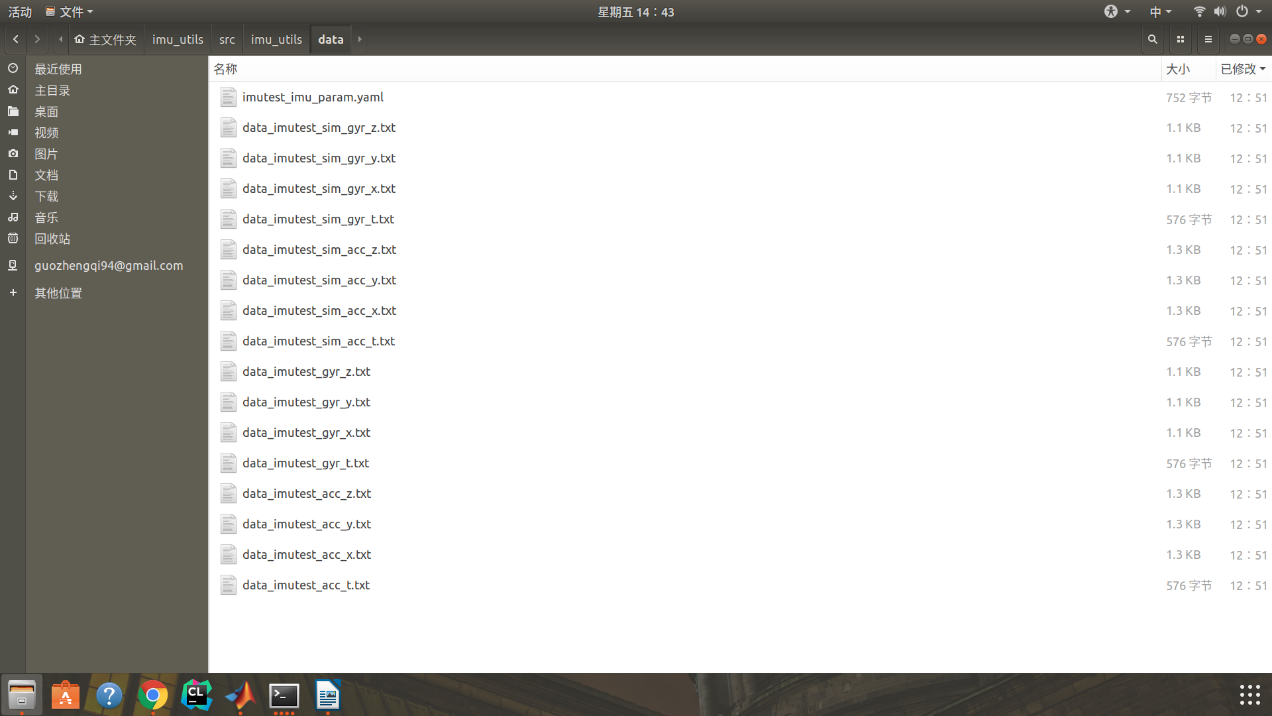
* cd ~/catkin\_ws
* catkin\_make
* source ./devel/setup.bash

8、生成Allan方差

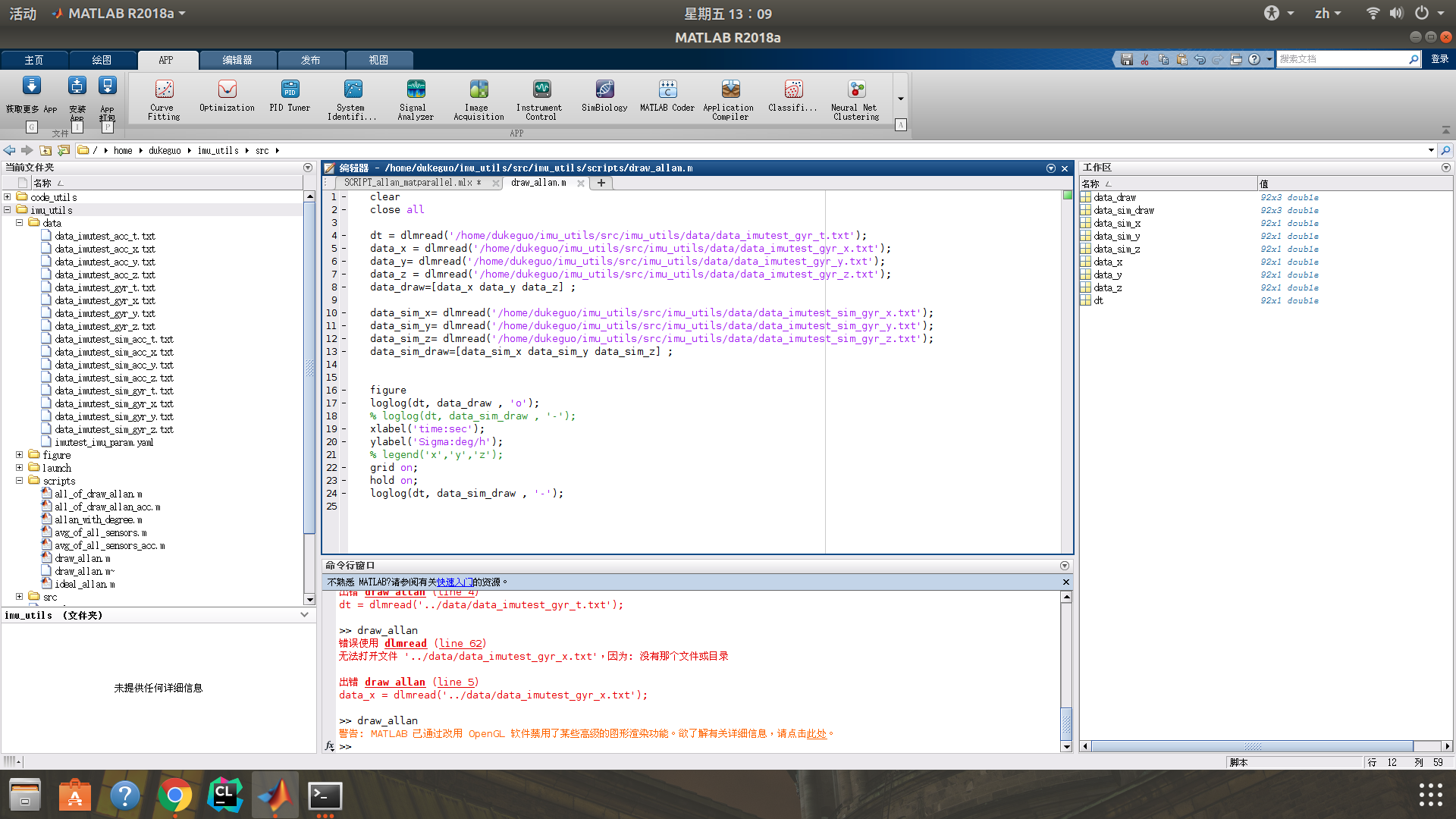
rosbag倍速回放imu.bag信息，并运行launch文件：

* rosbag play -r 200 imu.bag
* roslaunch imu\_utils imu.launch

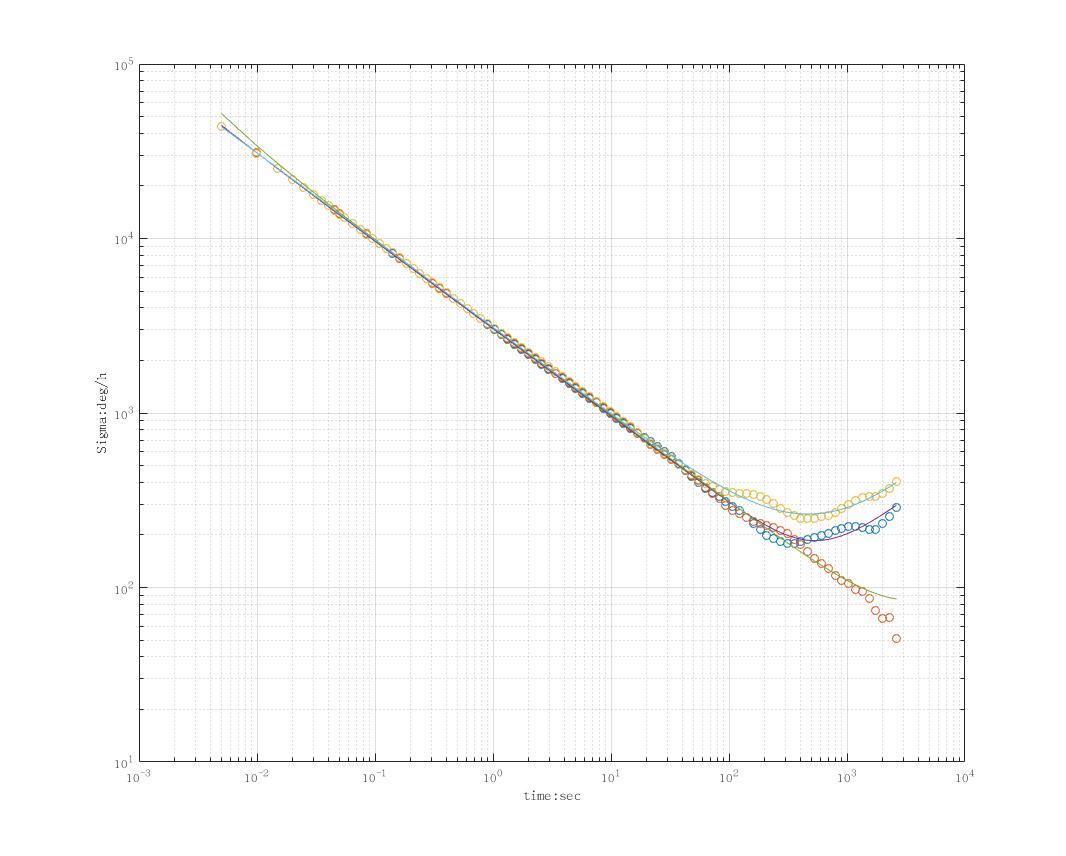
在imu\_utils/data文件夹下，会生成16个txt文件：

9. 绘制Allan方差图

修改 draw\_allan.m中文件路径：



运行结果：



根据Allan方差图即可读出相应的误差。

## kalibr\_allan

使用Matlab2018

1、编译kalibr\_allan

* cd ~/catkin\_ws/src
* git clone https://github.com/rpng/kalibr\_allan.git
* cd ..
* catkin\_make

2、bag文件转换成mat

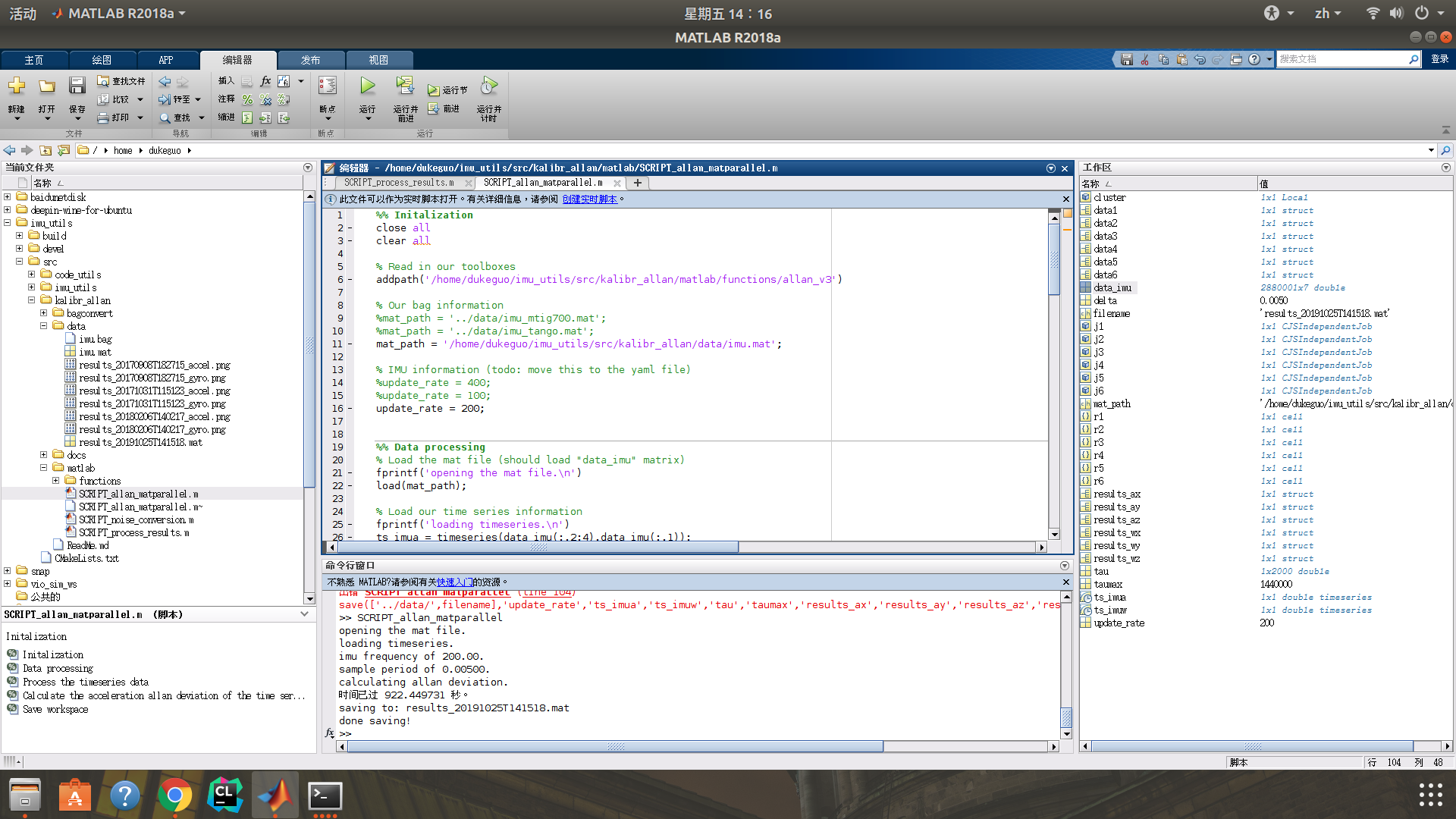
拷贝上述过程生成的imu.bag拷贝到~/catkin\_ws/src/kalibr\_allan/data文件夹中，参考：https://github.com/rpng/kalibr\_allan ，使用bagconvert将.bag转换成.mat文件：

* rosrun bagconvert bagconvert /home/dukeguo/imu\_utils/src/kalibr\_allan/data/imu.bag imu

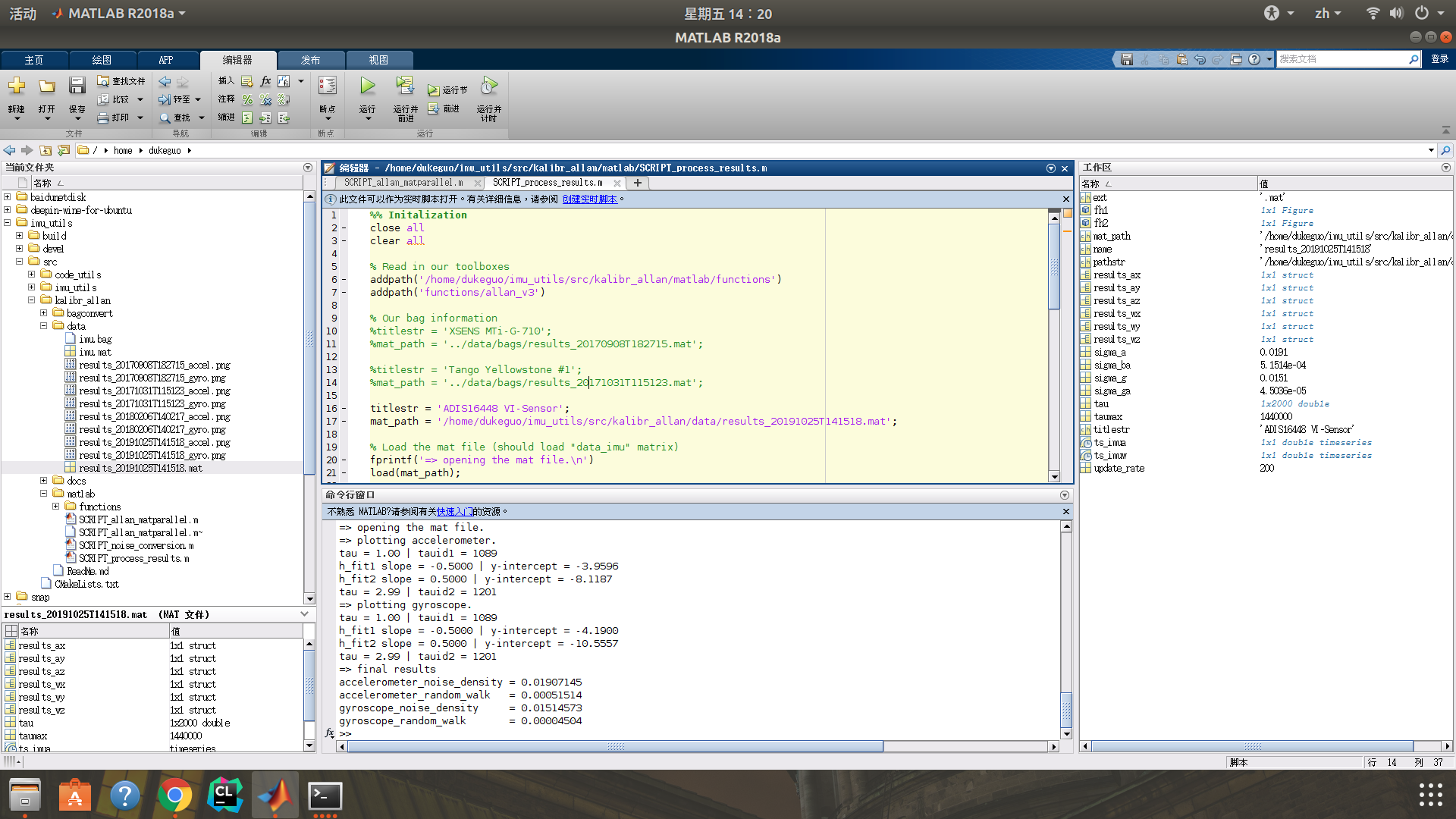
在imu.bag文件的位置生成转换的结果imu.mat，许多文件是“/imu0”，记得修改。

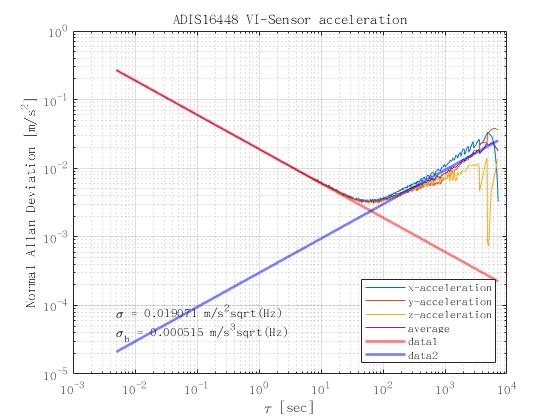
3、生成曲线参数文件

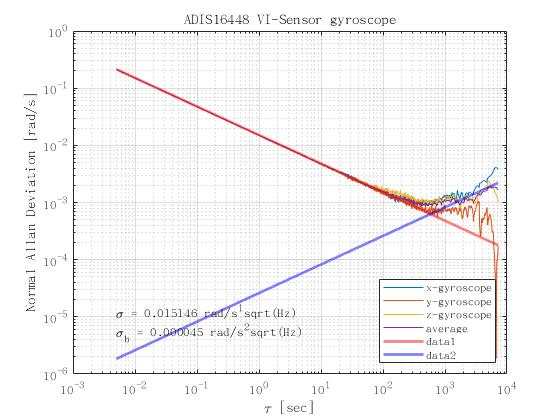
修改~/catkin\_ws/src/kalibr\_allan/matlab文件夹下的SCRIPT\_process\_results.m中.mat路径，即可画出allan曲线。

运行SCRIPT\_allan\_matparallel.m生成results\_20191025T141518.mat文件，用时922s。

4、绘制Allan曲线：

运行SCRIPT\_process\_results.m生成Allan结果

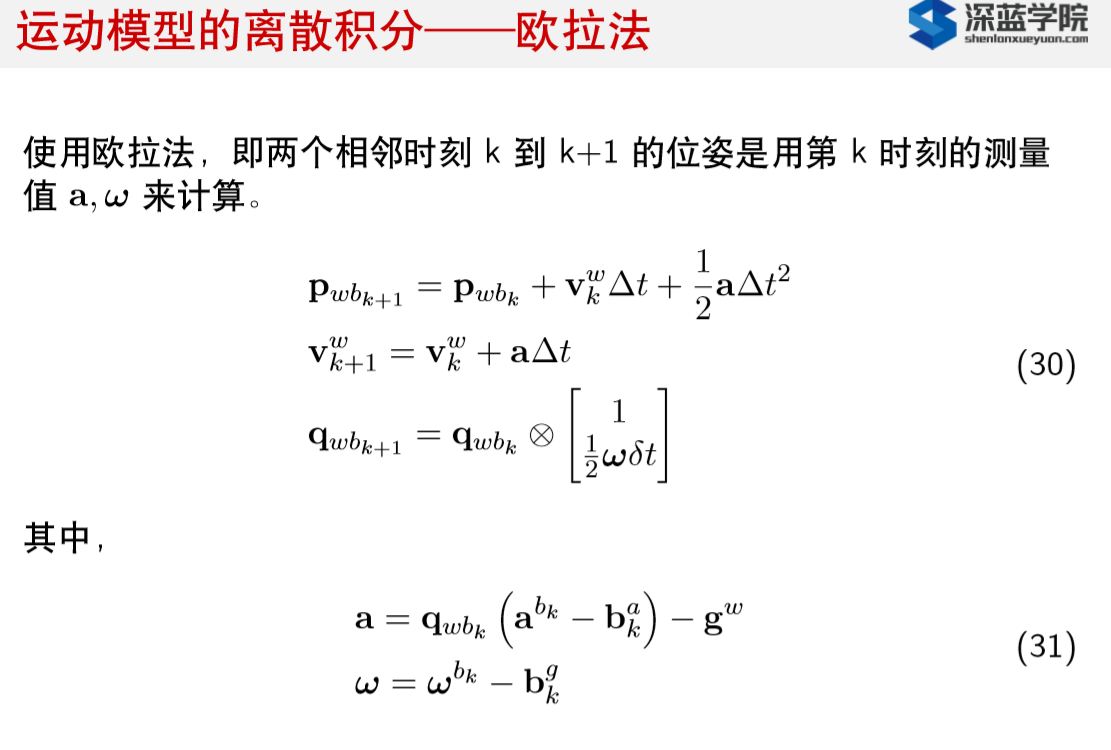




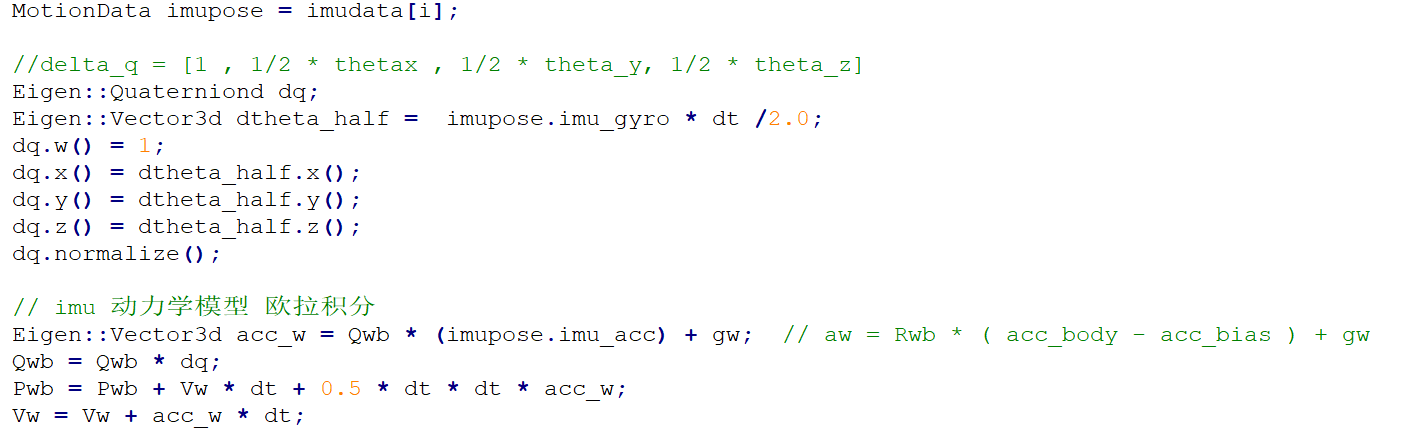
# 使用欧拉积分和中值积分进行IMU模拟

在进行IMU数据仿真的时候，主要的积分方法有欧拉积分和中值积分两种。

## 欧拉积分

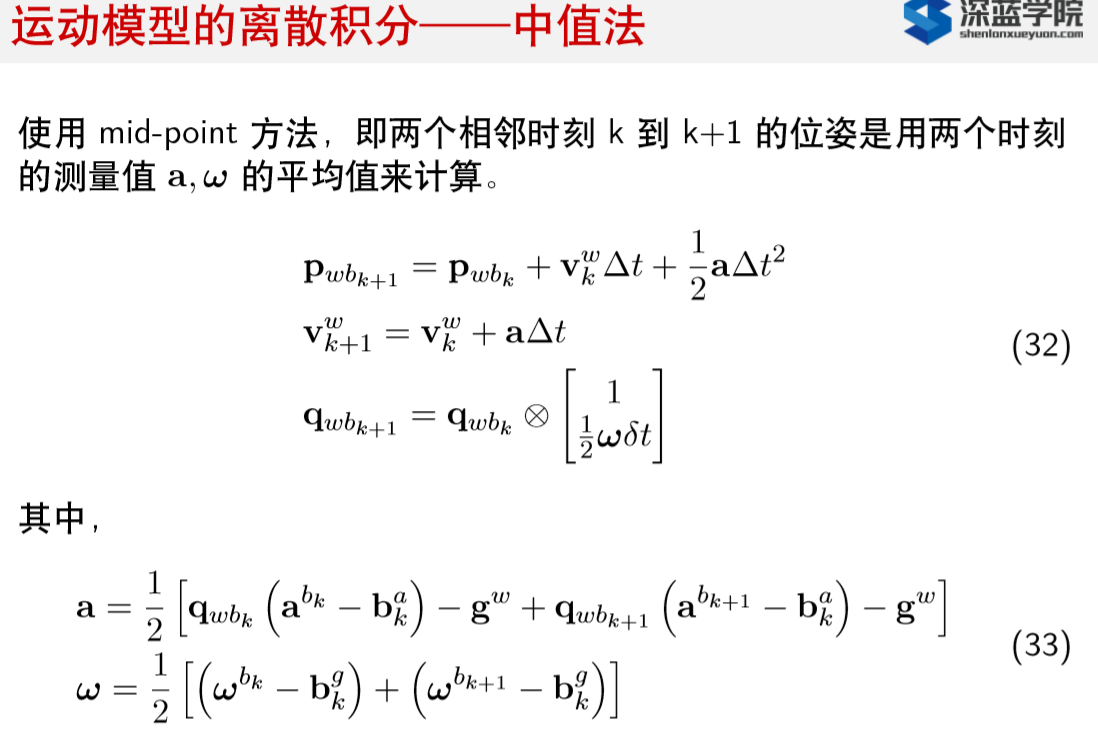


代码实现为：

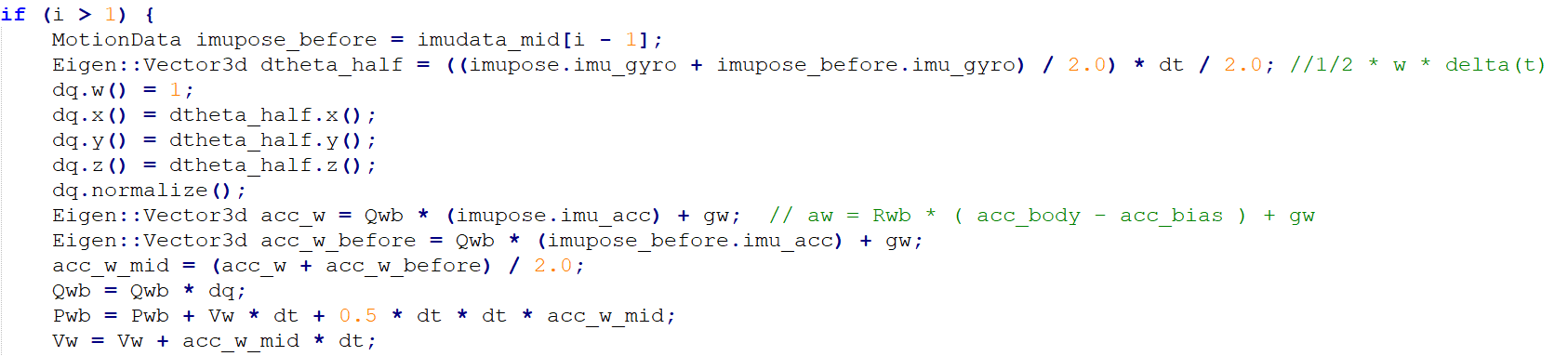


## 中值积分

中值积分与欧拉积分不同的是，从第二项开始，每一项都是与前项的平均值：



代码实现为：



如果将两个结果放在一起的话，可以明显看出，中值积分的结果更接近groundtruth：

