第二章作业：

# 1.设置IMU仿真代码中的不同参数，生成allen方差标定曲线：

使用的工具为imu\_utils,通过在网上查阅相关资料的。仿真程序的的参数设置在vio\_data\_simulation-ros\_version的param.h文件中。设置如下

图形用户界面, 文本, 应用程序

描述已自动生成

标定后的allen方差曲线为：

图表, 折线图, 直方图

描述已自动生成

图 1加速度的方差曲线

图表, 直方图

描述已自动生成

图 2 角速度的方差曲线

标定结果

表格 1 陀螺仪的标定结果

|  |  |  |
| --- | --- | --- |
| 陀螺仪 | 设定值 | 标定值 |
| 白噪声 | 0.015 | 0.0148 |
| 偏置 | 0.00005 | 0.000056 |

表格 2 加速度计标定结果

|  |  |  |
| --- | --- | --- |
| 加速度计 | 设定值 | 标定值 |
| 白噪声 | 0.019 | 0.0183 |
| 偏置 | 0.0005 | 0.0002 |

# 2.用欧拉积分替换中值积分

代码如下：

for (int i = 1; i < imudata.size(); ++i) {

MotionData imupose = imudata[i];

MotionData imu1 = imudata[i-1]; //上一个时刻的IMU信息

Eigen::Quaterniond dq;

Eigen::Vector3d dtheta\_half = 0.5\*(imupose.imu\_gyro+imu1.imu\_gyro) \* dt /2.0; //w = 1/2(w1+w2)

dq.w() = 1;

dq.x() = dtheta\_half.x();

dq.y() = dtheta\_half.y();

dq.z() = dtheta\_half.z();

dq.normalize();

/// 中值积分

Eigen::Quaterniond Qwb1 = Qwb \* dq; //Qwb1当前时刻的姿态四元数，Qwb为上一时刻的

Eigen::Vector3d acc\_w = 0.5\*(Qwb1\*imupose.imu\_acc+Qwb\*imu1.imu\_acc)+gw; //中值法的公式，这里gw取符号选取根定义有关

Pwb = Pwb + Vw \* dt + 0.5 \* dt \* dt \* acc\_w;

Vw = Vw + acc\_w \* dt;

Qwb=Qwb1;

结果：

图形用户界面, 图表

描述已自动生成

图 3欧拉法

图形用户界面, 图表

描述已自动生成

图 4中值法

对比上述两个轨迹可得，中值法得到的轨迹更加贴近真实轨迹。

# 3.阅读从已有轨迹生成imu数据的论文，撰写总结推导：

总的来讲，b样条法可以让我们得到所需要的轨迹的方程，得到轨迹方程之后进行微分操作得到需要的IMU数据。