

深蓝学院 VIO 课程第二课作业

基础作业，必做

- ① 设置 IMU 仿真代码中的不同的参数，生成 Allen 方差标定曲线。
allan 方差工具：
https://github.com/gaowenliang/imu_utils
https://github.com/rpng/kalibr_allan

1. ALLAN 方差标定曲线

使用 ROS 版本的 vio_data_simulation 包，设置了陀螺仪（gyr）和加速度计（acc）三组不同噪声参数，单位与包内设置的一致，如下表所示：

表 1 设置的 IMU 三组噪声参数

IMU Sequence	White Noise	Random Walk
Gyr_1	0.015	0.00005 (5e-5)
Acc_1	0.019	0.0005 (5e-4)
Gyr_2	0.35	0.001 (1e-3)
Acc_2	0.29	0.03 (3e-2)
Gyr_3	0.0075	0.000008(8e-6)
Acc_3	0.0042	0.00009 (9e-5)

生成 3 个 rosbag 文件，分别用 imu_utils 工具和 kalibr_allan 工具进行标定，标定过程和结果如下：

1.1 使用 kalibr_allan 工具进行标定

1.1.1 标定过程及结果：

1) 使用可执行程序 bagconvert 将 IMU 数据 rosbag 转成 mat 格式；

```

应用程序输出
Custom Executable ✕ Custom Executable ✕
13:54:42: Starting /home/vance/vio_ws/devel/lib/bagconvert/bagconvert...
[ INFO] [1561614882.972123694]: Starting up
[ INFO] [1561614883.545942097]: BAG Path is: /home/vance/dataset/rosbags/vio_imu_3.bag
[ INFO] [1561614883.546681028]: MAT Path is: /home/vance/dataset/rosbags/vio_imu_3.mat
[ INFO] [1561614883.546693942]: IMU Topic is: /imu
[ INFO] [1561614883.546698981]: Reading in rosbag file...
[ INFO] [1561614885.356632500]: Done processing bag
[ INFO] [1561614889.058592812]: Done processing IMU data
[ INFO] [1561614889.058616471]: Size of IMU message: 20160007
13:54:49: /home/vance/vio_ws/devel/lib/bagconvert/bagconvert exited with code 0
    
```

图 1 bagconvert 程序控制台输出结果

2) 使用脚本 allan_matparallel.m 分析 IMU 数据;

```

命令窗口
不熟悉 MATLAB? 请参阅有关快速入门的资源。

>> SCRIPT_allan_matparallel
opening the mat file.
loading timeseries.
imu frequency of 200.00.
sample period of 0.00500.
calculating allan deviation.
时间已过 616.770076 秒。
saving to: results_20190627T123713.mat
done saving!
    
```

图 2 脚本 allan_matparallel.m 输出结果

3) 使用脚本 process_results.m 获取标定结果并画图。

```

>> SCRIPT_process_results
=> opening the mat file.
=> plotting accelerometer.
tau = 1.00 | tauid1 = 1089
h_fit1 slope = -0.5000 | y-intercept = -3.9549
h_fit2 slope = 0.5000 | y-intercept = -8.1581
tau = 2.99 | tauid2 = 1201
=> plotting gyroscope.
tau = 1.00 | tauid1 = 1089
h_fit1 slope = -0.5000 | y-intercept = -4.2015
h_fit2 slope = 0.5000 | y-intercept = -10.3266
tau = 2.99 | tauid2 = 1201
=> final results
accelerometer_noise_density = 0.01916032
accelerometer_random_walk = 0.00049525
gyroscope_noise_density = 0.01497353
gyroscope_random_walk = 0.00005663
    
```

图 3 脚本 process_results.m 输出结果

从以上标定过程和图可以看出生成的 IMU 数据有 2016 万个，IMU 的频率设置为 200Hz，数据的分析总共花了约 10 分钟时间。三组数据的标定结果如图 4 至图 6 所示：

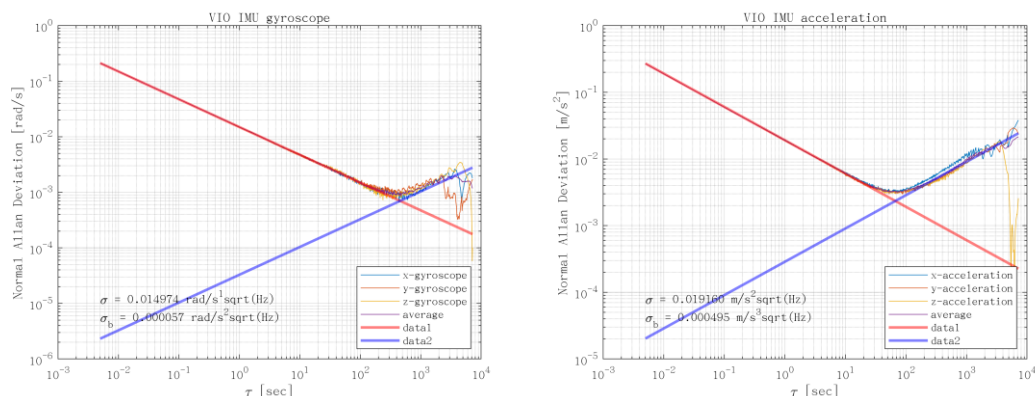


图 4 第一组数据的陀螺仪（左）和加速度计（右）的 Allan 曲线

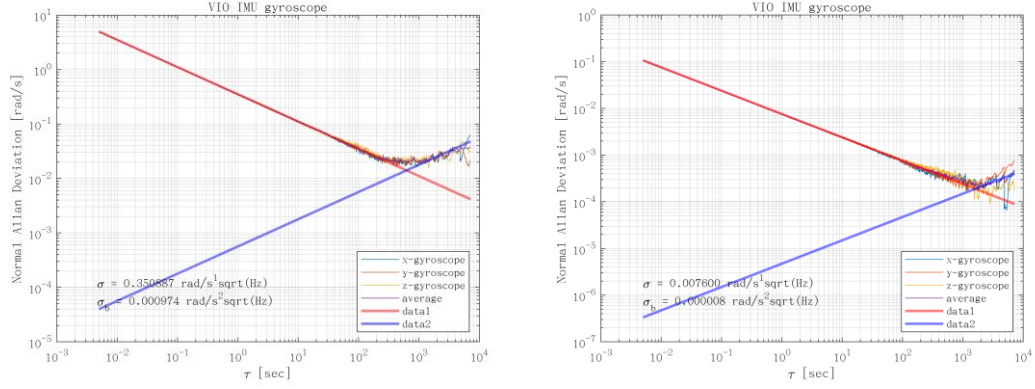


图 5 第二组数据的陀螺仪（左）和加速度计（右）的 Allan 曲线

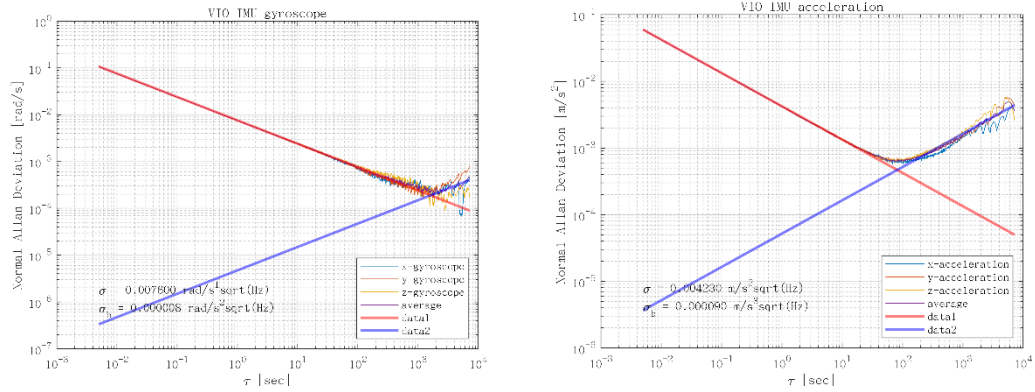


图 6 第三组数据的陀螺仪（左）和加速度计（右）的 Allan 曲线

1.1.2 结果分析:

从图 4 至图 6 可以得到三组数据的标定结果，将结果记于表 2，并进行分析：

表 2 kalibr_allan 工具标定结果分析

IMU Sequence	White Noise Truth	White Noise Calibration	White Noise Error Percentage	Random Walk Truth	Random Walk Calibration	Random Walk Error Percentage
Gyr_1	0.015	0.014974	0.17%	0.00005	0.000057	14%
Acc_1	0.019	0.019160	0.84%	0.0005	0.000495	1%
Gyr_2	0.35	0.350887	0.25%	0.001	0.000974	2.6%
Acc_2	0.29	0.293729	1.28%	0.03	0.030643	2.14%
Gyr_3	0.0075	0.007600	1.33%	0.000008	0.000008	0%
Acc_3	0.0042	0.004230	0.71%	0.00009	0.000090	0%

从表 2 中可以看出标定的整体结果较为准确，标定出的误差基本可以保持在真实值的 3%以内，虽然第一组数据的陀螺仪随机游走误差与真实值的相对误差比较高，但由于其绝对数值较小且数量级一致，因此标定结果较为可靠。

1.1.3 其他:

- kalibr_allan 包的编译依赖 MATLAB，需要安装 MATLAB；
- bagconvert 程序运行需要 R2017a 以上版本，本人使用 R2016a 版本运行时出现 Segmentation fault 错误，IDE Debug 定位还不准确（如图 7 所示），经过排查发现是 MATLAB 的问题，换成 R2018a 版本后解决问题（群里有人说 2017 版本可以使用）；
- bagconvert 程序运行结果可能只生成一个几百 KB 的 mat 数据，这说明转换失败，原因在于如果输入的第二个 topic 参数以 '/' 字符开头时（如 '/imu'），main.cpp 代码中第 77 行的 m.getTopic() 函数并不会获取首字符 '/'，因此造成了判断语句的值为 false，故数据并没有被成功转换。解决方式一是输入 topic 参数时不要以 '/' 字符开头，二是在代码中删除首字符 '/'，如图 8 所示。

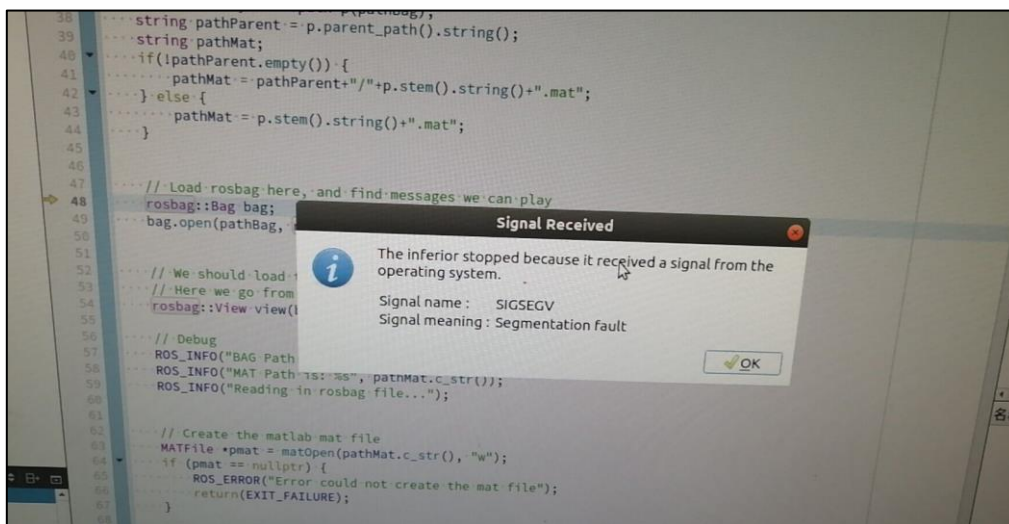


图 7 低 MATLAB 版本原因造成 bagconvert 程序运行错误

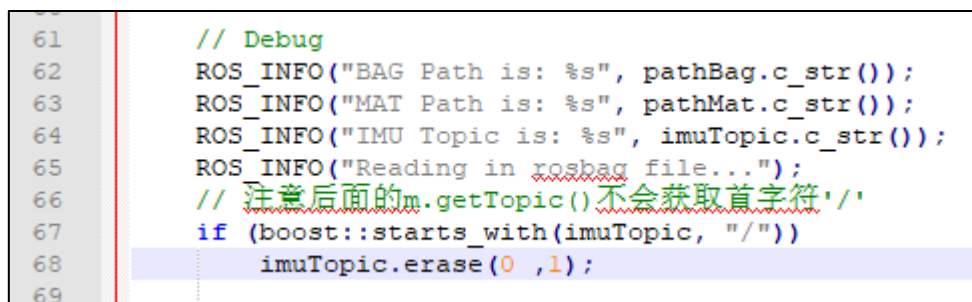


图 8 bagconvert 源代码中加入删除首字符'/'的字段

1.2 使用 imu_utils 工具进行标定

1.2.1 标定过程及结果：

1) 运行 imu_an 节点，分别播放三组 IMU 数据的记录包（rosbag），在 data/ 文件夹下生成一系列数据；

2) 参考 scripts/ 文件夹下的脚本，修改并生成 Allan 曲线，如图 9 至图 11 所示：

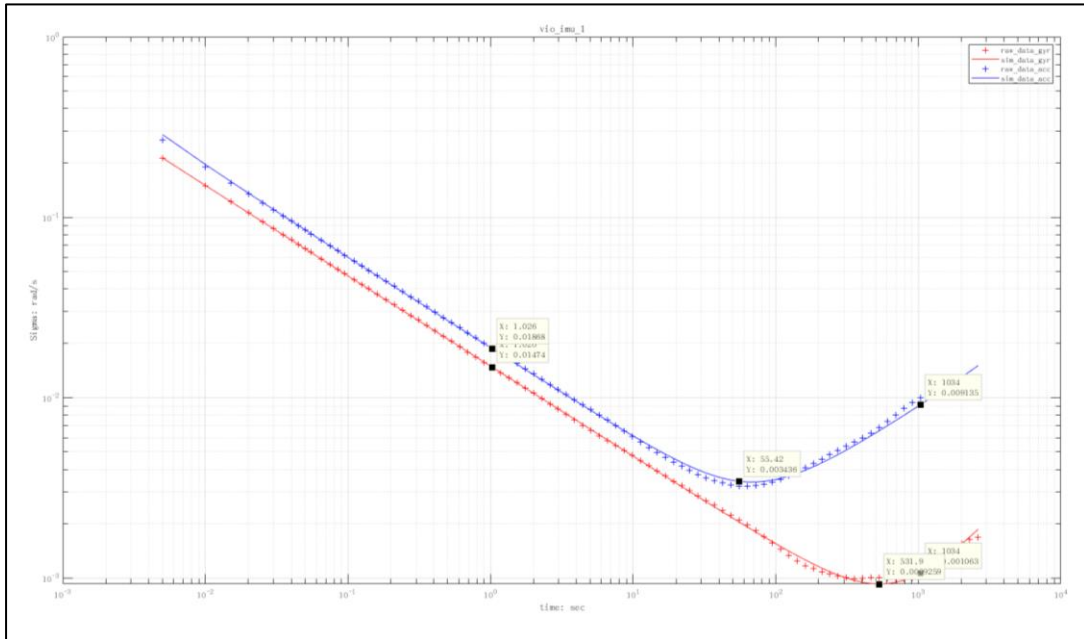


图 9 第一组数据陀螺仪（红）和加速度计（蓝）的 Allan 曲线

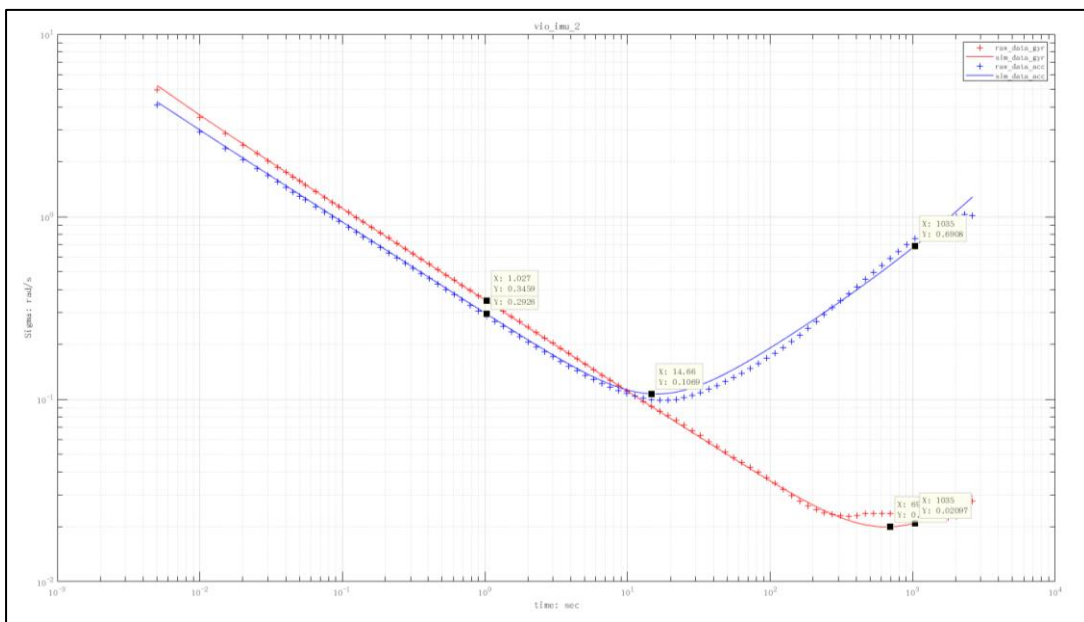


图 10 第二组数据陀螺仪（红）和加速度计（蓝）的 Allan 曲线

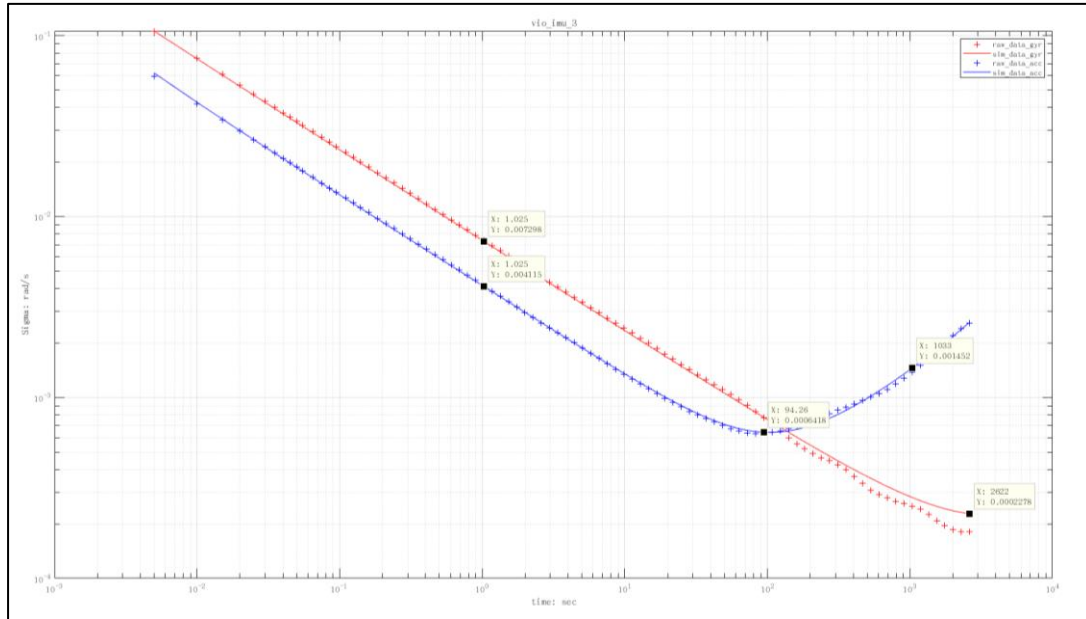


图 11 第三组数据陀螺仪（红）和加速度计（蓝）的 Allan 曲线

根据资料¹，斜率为-1/2 直线上 $\tau = 1$ 的点为高斯白噪声，斜率为 1/2 直线上 $\tau = 3$ 的点为随机游走噪声。从图 9 至图 11 可以看出，此工具标定出的高斯白噪声数值与真实值相近，但随机游走噪声值标定不准确。

附件里包含了本次程序运行的 log 文件，从 log 文件里也可以看出标定出的值误差很大。

1.1.2 其他：

- imu_utils 工具包依赖 code_utils 包，需要到作者的 github 主页上下载此包到工作空间，首先编译 code_utils 包，成功后再将 imu_utils 工具包放入工作空间编译。
- 标定陀螺仪和加速度计的单位数量级好像不一样，标定陀螺仪时化为 rad/s 的单位，标定加速度计时单位不变，才能有上图的效果。具体原因没有深入分析，MATLAB 代码不是很熟。

¹ <https://github.com/ethz-asl/kalibr/wiki/IMU-Noise-Model#from-the-allan-standard-deviation-ad>

② 将 IMU 仿真代码中的欧拉积分替换成中值积分。

2. IMU 数据仿真中欧拉积分替换为中值积分

2.1 中值积分源代码：

中值积分部分代码如图 12 所示：

```

152 // imu 动力学模型 欧拉积分
153 // Eigen::Vector3d acc_w = Qwb * (imupose.imu_acc) + gw; // aw = Rwb * (acc_body - acc_bias) + gw
154 // Qwb = Qwb * dq;
155 // Qwb.normalize();
156 // Fwb = Fwb + Vw * dt + 0.5 * dt * dt * acc_w;
157 // Vw = Vw + acc_w * dt;
158
159 // 中值积分
160 MotionData imupose1 = imudata[i-1]; // imupose1是第k帧数据
161 MotionData& imupose2 = imupose; // imupose2是第k+1帧数据
162 Eigen::Quaterniond dq_mid;
163 Eigen::Vector3d dtheta_half_mid = (imupose1.imu_gyro + imupose2.imu_gyro) * dt / 4.0;
164 dq_mid.w() = 1;
165 dq_mid.x() = dtheta_half_mid.x();
166 dq_mid.y() = dtheta_half_mid.y();
167 dq_mid.z() = dtheta_half_mid.z();
168 dq_mid.normalize();
169
170 Eigen::Quaterniond Qwb2(Qwb); // 这里Qwb2是Q_wbk+1, Qwb是Q_wbk
171 Qwb2 = Qwb * dq_mid;
172 Qwb2.normalize();
173
174 // aw = 0.5 * [Qwb * (acc_body - acc_bias) + gw + Qwb_k+1 * (acc_body_k+1 - acc_bias) + gw]
175 Eigen::Vector3d acc_w = ((Qwb * imupose1.imu_acc + gw) + (Qwb2 * imupose2.imu_acc + gw)) / 2.0;
176 Qwb = Qwb2;
177 Fwb = Fwb + Vw * dt + 0.5 * dt * dt * acc_w;
178 Vw = Vw + acc_w * dt;
    
```

图 12 中值积分部分代码

注意：

- 源代码中速度 V 的更新早于位置 P，这导致了计算位置 P 时使用的速度值为 V_{k+1} ，实际应使用 V_k ，故在此对两种变量的更新顺序进行了调换。
- 四元素需要归一化。

2.2 替换前后的效果对比：

使用 vio_data_simulation 工具（非 ROS 版）生成数据，使用欧拉积分时生成的数据如图 13 所示，使用中值积分时生成的数据如图 14 所示：

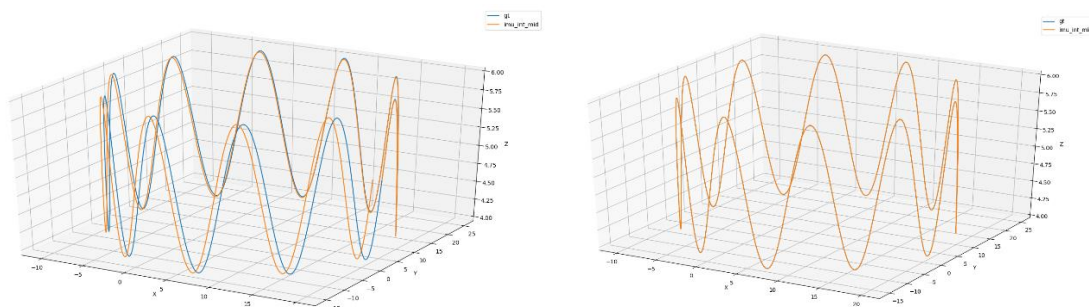


图 13 欧拉积分（左）和中值积分（右）模拟轨迹效果图

由图 13 可以看出，两种积分算出的 IMU 轨迹与真实值对比有挺大差别。欧拉积分方式的前期轨迹基本重合，后期轨迹均略有偏差；中值积分的轨迹基本与真实值重合，这说明使用中值积分计算 IMU 轨迹误差更小，更为精确。

为定量分析效果，对两种积分方式生成的轨迹分别与真实值进行误差计算，取均方根误差（RMSE）作为评价指标，分别计算得 0.472876（欧拉积分）与 0.030596（中值积分），证明中值积分比欧拉积分准确得多，精确度提升明显。

提升作业，选做

阅读从已有轨迹生成 imu 数据的论文，撰写总结推导：

- 2013 年 BMVC, Steven Lovegrove ,Spline Fusion: A continuous-time representation for visual-inertial fusion with application to rolling shutter cameras.

3. 阅读论文，撰写总结推导

3.1 介绍

本文介绍了一种利用卷帘式 CMOS 相机和 MEMS IMUs 等传感器进行 SLAM 和视觉惯性标定的方法。利用相机轨迹的连续时间模型，融合来自其它不同步的高速率传感器的信息，同时限制状态大小。对相机的卷帘快门进行建模，并能在惯性测量中产生误差。该模型不局限于视觉惯性 SLAM，还可以简化其他传感器的集成，如旋转 SICK 激光雷达。

3.2 连续时间表示

为了在刚体旋转和平移的流形空间中保持轨迹连续，在局部处理上用李代数 $se(3)$ 表示位姿。

3.2.1 相机位姿变换

$T_{w,a}$ 到 $T_{w,b}$ 在 Δt 时间内的变换用 $T_{b,a}$ 表示：

$$\mathbf{T}_{b,a} = \begin{bmatrix} \mathbf{R}_{b,a} & \mathbf{a}_b \\ \mathbf{0}^T & 1 \end{bmatrix}, \quad \mathbf{T}_{b,a} \in SE(3), \quad \mathbf{R}_{b,a} \in SO(3),$$

则其速度矩阵可以表示为：

$$\Omega = \frac{1}{\Delta} \log(\mathbf{T}_{b,a}), \Omega \in \mathbb{R}^{4 \times 4},$$

3.2.2 在 SE3 中的 C² 连续曲线

核心方法是一个连续的轨迹表示，可以：

- 本地控制，允许系统在线和批量运行。
- C² 的连续性，使能够预测 IMU 的测量值。
- 一个很好的近似最小扭矩轨迹

立方体样条曲线无法处理三维旋转，选择使用李代数形成的累积基函数来参数化连续轨迹。这种表示不仅是 C² 连续的，而且它还提供了一个非常简单的二阶导数公式。

累积 B 样条曲线参数化允许在样条曲线上的任意一点计算解析时间导数。可以很容易地合成加速度计和陀螺仪的测量值，反过来可以利用它们对观测到的测量值形成直接误差。

2.2.2.1 用累积基函数表示 B-样条

k-1 阶 B 样条曲线的标准基函数表示由下式给出：

$$\mathbf{p}(t) = \sum_{i=0}^n \mathbf{p}_i B_{i,k}(t) \quad (1)$$

$B_{i,k}(t)$ 是基函数，也叫调和函数，由 De Boor - Cox recursive 公式计算； \mathbf{p}_i 为 t_i 时刻的控制点。（1）式可改写为它的累积形式：

$$\mathbf{p}(t) = \mathbf{p}_0 \tilde{B}_{0,k}(t) + \sum_{i=1}^n (\mathbf{p}_i - \mathbf{p}_{i-1}) \tilde{B}_{i,k}(t) \quad (2)$$

其中 $\tilde{B}_{i,k}(t) = \sum_{j=i}^n B_{j,k}(t)$ 为累积基函数，利用对数和指数映射，重写（2）式，得到：

$$\mathbf{T}_{w,s}(t) = \exp(\tilde{B}_{0,k}(t) \log(\mathbf{T}_{w,0})) \prod_{i=1}^n \exp(\tilde{B}_{i,k}(t) \Omega_i), \quad (3)$$

其中， $\Omega_i = \log(\mathbf{T}_{w,i-1}^{-1} \mathbf{T}_{w,i}) \in \mathfrak{se3}$ ， $\mathbf{T}_{w,s}(t) \in \mathbb{SE3}$ 为控制点位姿。

2.2.2.2 累积立方 B 样条

3 阶立方 B 样条曲线，假设控制点在时间上平均分布，设置时间 $t \in [t_i, t_{i+1})$ 内 4 个控制点为 $[t_{i-1}, t_i, t_{i+1}, t_{i+2}]$ ，令 $s(t) = (t - t_0)/\Delta t$ ， $u(t) = s(t) - s_i$ ，则累积偏差 $\tilde{\mathbf{B}}(u)$ 和他的导数为：

$$\tilde{\mathbf{B}}(u) = \mathbf{C} \begin{bmatrix} 1 \\ u \\ u^2 \\ u^3 \end{bmatrix}, \quad \dot{\tilde{\mathbf{B}}}(u) = \frac{1}{\Delta t} \mathbf{C} \begin{bmatrix} 0 \\ 1 \\ 2u \\ 3u^2 \end{bmatrix}, \quad \ddot{\tilde{\mathbf{B}}}(u) = \frac{1}{\Delta t^2} \mathbf{C} \begin{bmatrix} 0 \\ 0 \\ 2 \\ 6u \end{bmatrix}, \quad \mathbf{C} = \frac{1}{6} \begin{bmatrix} 6 & 0 & 0 & 0 \\ 5 & 3 & -3 & 1 \\ 1 & 3 & 3 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

则样条曲线中的位姿可以表示为：

$$\mathbf{T}_{w,s}(u) = \mathbf{T}_{w,i-1} \prod_{j=1}^3 \exp(\tilde{\mathbf{B}}(u)_j \Omega_{i+j}), \quad (4)$$

$$\dot{\mathbf{T}}_{w,s}(u) = \mathbf{T}_{w,i-1} \left(\dot{\mathbf{A}}_0 \mathbf{A}_1 \mathbf{A}_2 + \mathbf{A}_0 \dot{\mathbf{A}}_1 \mathbf{A}_2 + \mathbf{A}_0 \mathbf{A}_1 \dot{\mathbf{A}}_2 \right), \quad (5)$$

$$\ddot{\mathbf{T}}_{w,s}(u) = \mathbf{T}_{w,i-1} \left(\ddot{\mathbf{A}}_0 \mathbf{A}_1 \mathbf{A}_2 + \mathbf{A}_0 \ddot{\mathbf{A}}_1 \mathbf{A}_2 + \mathbf{A}_0 \mathbf{A}_1 \ddot{\mathbf{A}}_2 + 2(\dot{\mathbf{A}}_0 \dot{\mathbf{A}}_1 \mathbf{A}_2 + \dot{\mathbf{A}}_0 \mathbf{A}_1 \dot{\mathbf{A}}_2 + \mathbf{A}_0 \dot{\mathbf{A}}_1 \dot{\mathbf{A}}_2) \right), \quad (6)$$

$$\mathbf{A}_j = \exp(\Omega_{i+j} \tilde{\mathbf{B}}(u)_j), \quad \dot{\mathbf{A}}_j = \mathbf{A}_j \Omega_{i+j} \dot{\tilde{\mathbf{B}}}(u)_j,$$

$$\ddot{\mathbf{A}}_j = \dot{\mathbf{A}}_j \Omega_{i+j} \dot{\tilde{\mathbf{B}}}(u)_j + \mathbf{A}_j \Omega_{i+j} \ddot{\tilde{\mathbf{B}}}(u)_j$$

3.3 生成 VI 数据的模型

3.3.1 参数化

使用逆深度（更好地表达无限远点，简化单目特征初始化），可以轻松求导，合成加速度计和陀螺仪的测量值，用以观测。

$$\mathbf{p}_b = \mathcal{W}(\mathbf{p}_a; \mathbf{T}_{b,a}, \rho) = \pi\left([\mathbf{K}_b | \mathbf{0}] \mathbf{T}_{b,a} [\mathbf{K}_a^{-1} \begin{bmatrix} \mathbf{p}_a \\ 1 \end{bmatrix}; \rho\right], \quad (7)$$

$$\text{Gyro}(u) = \mathbf{R}_{w,s}^\top(u) \cdot \dot{\mathbf{R}}_{w,s}(u) + \text{bias}, \quad (8)$$

$$\text{Accel}(u) = \mathbf{R}_{w,s}^\top(u) \cdot (\ddot{\mathbf{s}}_w(u) + g_w) + \text{bias}, \quad (9)$$

3.3.2 最小化误差（测量值与观测值）

重投影误差和惯性误差均匀校正，由信息矩阵赋予权重。

误差函数：

$$\begin{aligned}
 E(\theta) = & \sum_{\hat{\mathbf{p}}_m} \left(\hat{\mathbf{p}}_m - \mathcal{W}(\mathbf{p}_r; \mathbf{T}_{c,s} \mathbf{T}_{w,s}(u_m)^{-1} \mathbf{T}_{w,s}(u_r) \mathbf{T}_{s,c}, \rho) \right)_{\Sigma_p}^2 + \\
 & \sum_{\hat{\omega}_m} \left(\hat{\omega}_m - \text{Gyro}(u_m) \right)_{\Sigma_\omega}^2 + \sum_{\hat{\mathbf{a}}_m} \left(\hat{\mathbf{a}}_m - \text{Accel}(u_m) \right)_{\Sigma_a}^2, \quad (10)
 \end{aligned}$$

其中， u_r 表示路标点第一次被观测到的时间，测量值 m 发生在 u_m 时刻， θ 表示要优化的参数向量，包括样条控制位姿、相机内参、路标逆深度值 ρ ，相机到 IMU 的变换 $\mathbf{T}_{c,s}$ 和 IMU 误差。使用非线性最小二乘迭代计算。