



深蓝学院
shenlanxueyuan.com

第二章作业分享

主讲人 晋炎彬



使用imu_utils完成标定

具体流程：

1. 下载和编译code_utils
2. 下载和编译imu_utils
3. 编译vio_data_simulation-ros_version
4. 执行生成的vio_data_simulation_node节点，生成bag文件

使用imu_utils完成标定

5.去data目录下的yaml文件下查看结果。结果如右图。

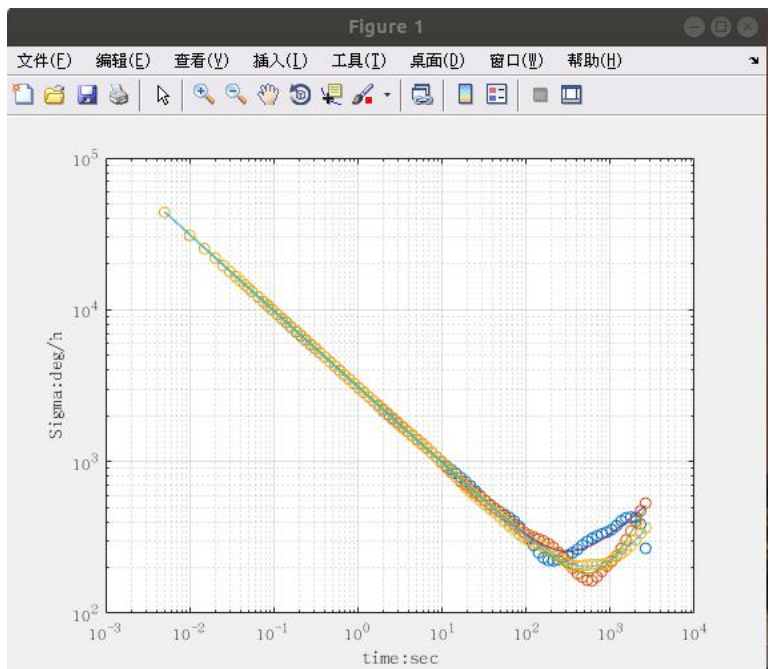
(这里的结果需要做一个单位转换,
 $1/\sqrt{\text{Hz}}$)

6.用matlab绘图

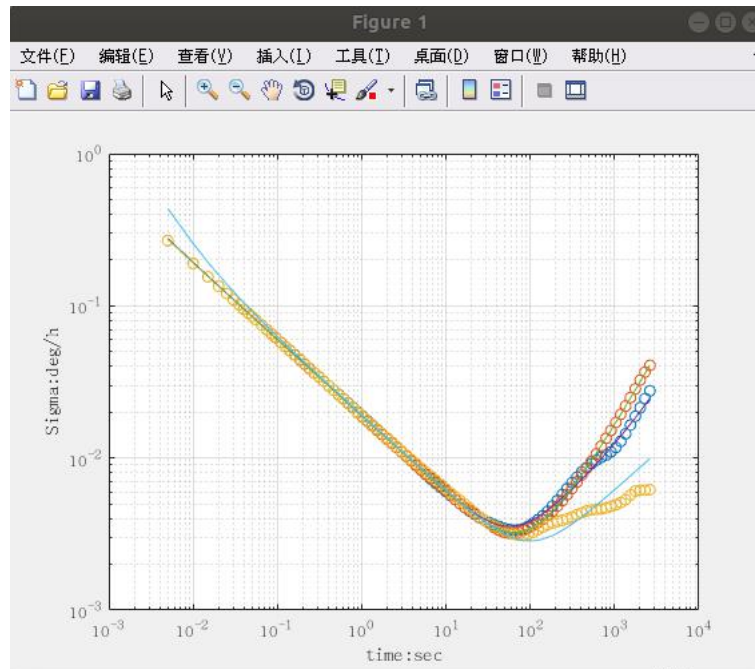
这一步执行完成之后就可以生成最后的Allan
曲线图。

```
%YAML:1.0
---
type: IMU
name: mytest
Gyr:
  unit: " rad/s"
  avg-axis:
    gyr_n: 2.1211921446409021e-01
    gyr_w: 1.0370031139236167e-03
  x-axis:
    gyr_n: 2.1088406551640301e-01
    gyr_w: 1.2412994011512023e-03
  y-axis:
    gyr_n: 2.1551626327376416e-01
    gyr_w: 8.8845671686073400e-04
  z-axis:
    gyr_n: 2.0995731460210346e-01
    gyr_w: 9.8125322375891381e-04
Acc:
  unit: " m/s^2"
  avg-axis:
    acc_n: 2.6915388200864510e-01
    acc_w: 3.2162969876312216e-03
  x-axis:
    acc_n: 2.6846985320623740e-01
    acc_w: 3.5510405443371232e-03
  y-axis:
    acc_n: 2.6720750921858061e-01
    acc_w: 3.2426070876615731e-03
  z-axis:
    acc_n: 2.7178428360111728e-01
    acc_w: 2.8552433308949680e-03
```

使用imu_utils完成标定



陀螺仪



加速度计

使用kalibr_allan 完成 allan 标定



深蓝学院
shenlanxueyuan.com

具体流程：

- 1.生成bag文件的过程和前面一样
- 2.把bag文件转换成.m文件

```
roslaunch bagconvert bagconvert imu.bag imu
```

3.运行SCRIPT_allan_matparallel.m脚本进行数据分析，具体运行情况如下：

使用kalibr_allan 完成 allan 标定



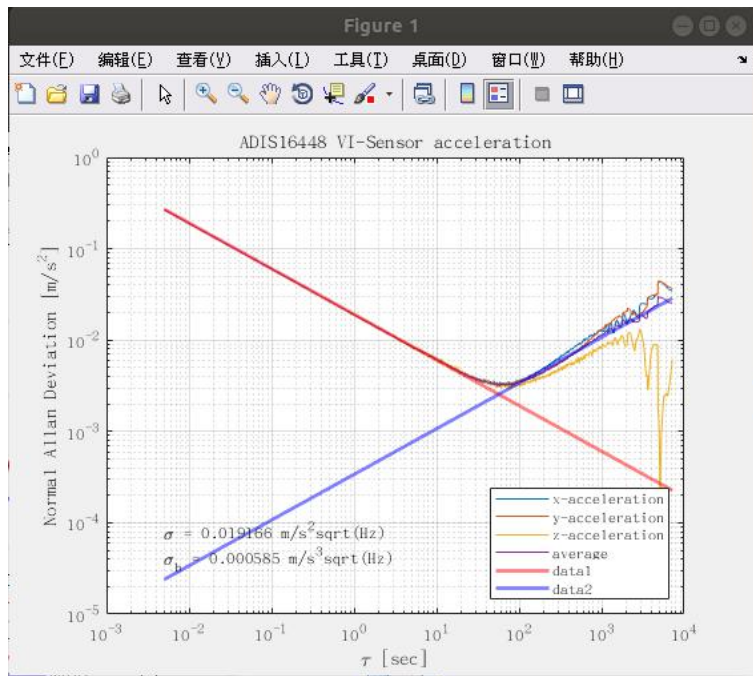
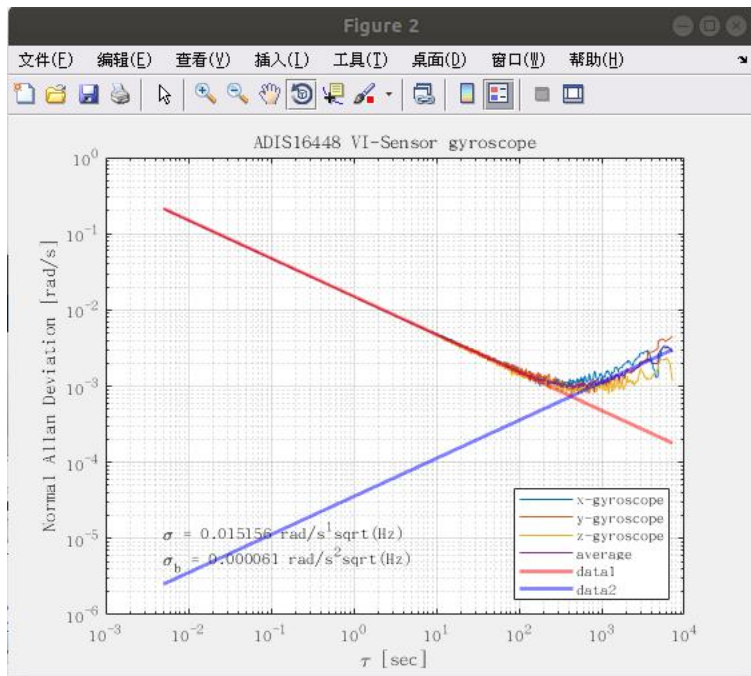
深蓝学院
shenlanxueyuan.com

```
>> 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 606.383703 秒。  
saving to: results_20210922T170855.mat  
done saving!
```

可以看出分析数据的时间大概为606s。

使用kalibr_allan 完成 allan 标定

4.运行SCRIPT_process_results.m脚本文件进行画图，最后的结果：



1.2 欧拉积分和中值积分

```
// 中值积分
MotionData imupose1 = imudata[i];
MotionData imupose2 = imudata[i-1];

//delta_q = [1, 1/2 * thetax, 1/2 * theta_y, 1/2 * theta_z]
Eigen::Quaterniond dq;
Eigen::Vector3d dtheta_half = (0.5 * (imupose1.imu_gyro + imupose2.imu_gyro)) * dt / 2.0;
dq.w() = 1;
dq.x() = dtheta_half.x();
dq.y() = dtheta_half.y();
dq.z() = dtheta_half.z();
dq.normalize();

Eigen::Quaterniond Qwb1 = Qwb * dq; //这里乘dp是为了算时刻 i 的四元数，以便算a的时候更准确
Eigen::Vector3d acc_w = 0.5 * (Qwb * (imupose1.imu_acc) + gw + Qwb1 * (imupose2.imu_acc) + gw);
Qwb = Qwb * dq;
Pwb = Pwb + Vw * dt + 0.5 * dt * dt * acc_w;
Vw = Vw + acc_w * dt;
```


1.2 欧拉积分和中值积分

结果对比：

Figure 1

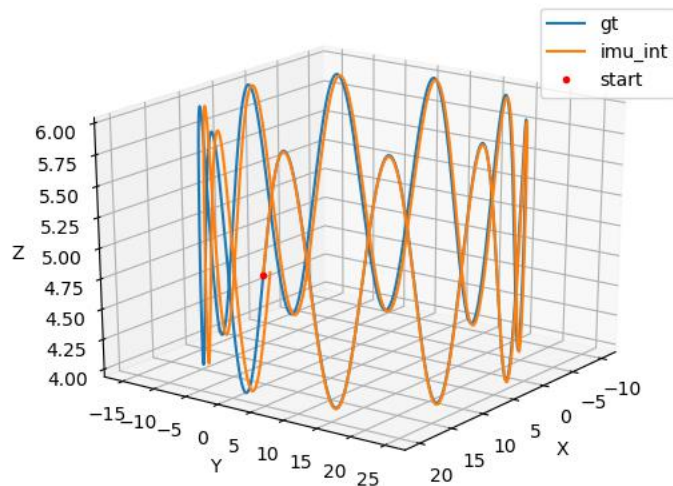
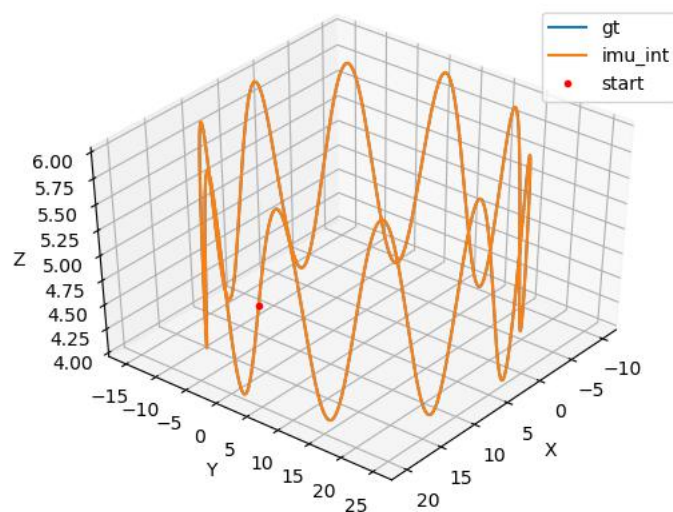


Figure 1



2、提升作业

贝塞尔曲线：

由其控制点决定其形状, n 个控制点对应着 $n - 1$ 阶的贝塞尔曲线，并且可以通过递归的方式来绘制。

缺点：

阶次 = 控制点个数 - 1

牵一发动全身，移动一个控制点，整段曲线都会变化

参考：<https://zhuanlan.zhihu.com/p/136647181>

2、提升作业

B样条：

B样条采用解决方案是贝塞尔曲线的拼接，也就是把一条曲线变为多段贝塞尔曲线的拼接。

优点：

- 1、可以指定阶次。
- 2、移动控制点只改变曲线的部分形状。

参考：<https://zhuanlan.zhihu.com/p/139759835>

<https://zhuanlan.zhihu.com/p/140921657>

2、提升作业

目的：

用离散的位姿利用B样条拟合成曲线方程，对曲线求导获得IMU数据

vio中相机和IMU时间戳同步

理论：将离散的位姿利用B样条拟合成连续的位姿轨迹曲线

1.B-Spline累积公式

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

2、提升作业

2.针对位姿，提出B-Spline累乘公式，写出离散位姿的BSpline曲线公式

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

3.k=4三次的基函数。累积基函数的一阶和二阶导数：

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

4.位姿曲线求导：

$$\begin{aligned} \mathbf{T}_{w,s}(u) &= \mathbf{T}_{w,i-1} (\dot{\mathbf{A}}_0 \mathbf{A}_1 \mathbf{A}_2 + \mathbf{A}_0 \dot{\mathbf{A}}_1 \mathbf{A}_2 + \mathbf{A}_0 \mathbf{A}_1 \dot{\mathbf{A}}_2), \\ \ddot{\mathbf{T}}_{w,s}(u) &= \mathbf{T}_{w,i-1} \begin{pmatrix} \ddot{\mathbf{A}}_0 \mathbf{A}_1 \mathbf{A}_2 + \mathbf{A}_0 \ddot{\mathbf{A}}_1 \mathbf{A}_2 + \mathbf{A}_0 \mathbf{A}_1 \ddot{\mathbf{A}}_2 + \\ 2(\dot{\mathbf{A}}_0 \dot{\mathbf{A}}_1 \mathbf{A}_2 + \dot{\mathbf{A}}_0 \mathbf{A}_1 \dot{\mathbf{A}}_2 + \mathbf{A}_0 \dot{\mathbf{A}}_1 \dot{\mathbf{A}}_2) \end{pmatrix}, \\ \mathbf{A}_j &= \exp(\Omega_{i+j} \tilde{\mathbf{B}}(u)_j), \quad \dot{\mathbf{A}}_j = \mathbf{A}_j \Omega_{i+j} \dot{\tilde{\mathbf{B}}}(u)_j, \\ \ddot{\mathbf{A}}_j &= \dot{\mathbf{A}}_j \Omega_{i+j} \dot{\tilde{\mathbf{B}}}(u)_j + \mathbf{A}_j \Omega_{i+j} \ddot{\tilde{\mathbf{B}}}(u)_j \end{aligned}$$

2、提升作业

5.对曲线求导获得IMU预测的角速度和加速度

$$\begin{aligned} Gyro(u) &= (R_{w,s}^T(u) \cdot \dot{R}_{w,s}(u))^V + bias \\ Accel(u) &= R_{w,s}^T(u) \cdot (\ddot{s}_w(u) + g_w) + bias \end{aligned}$$

Gyro公式有误

6.构建残差公式(测量-预测)，优化后可以得到B样条控制位姿、 bias、 相机外参内参等。

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

参考：https://udel.edu/~pgeneva/downloads/notes/2018_notes_mueffler2017arxiv.pdf

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

感谢各位聆听
Thanks for Listening

