第二章作业

第一题

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

ROS 和公司的中间件（华为 CM）对冲 话说中间件都是基于 ros 写的 对冲是一定的。。 笔记本和台式机都是这样 自己的笔记本实在没地方装了 C 盘剩不到 1 个 g D 盘都满了 虚拟机都装不下 双系统。。。 不过正好看见群里有人发了转换好的 mat 文件 直接在 matlab 上运行了 使用的是 kalibr_allan 尽力了 希望别影响成绩。。

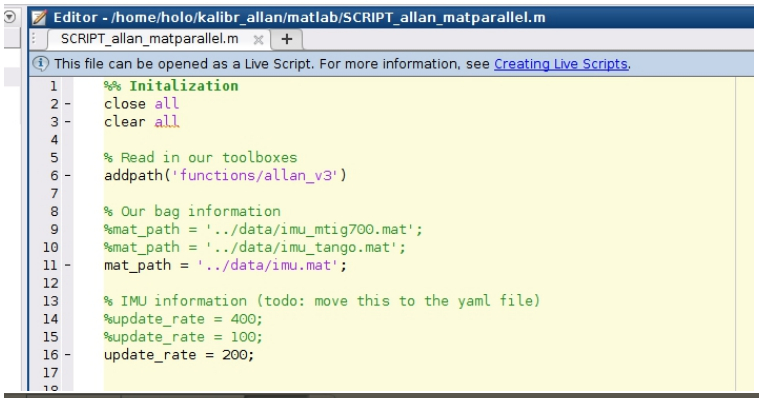
第一题的意思就是进行 imu 误差标定嘛 但是误差又是我们自己制造的 制造位置如下：

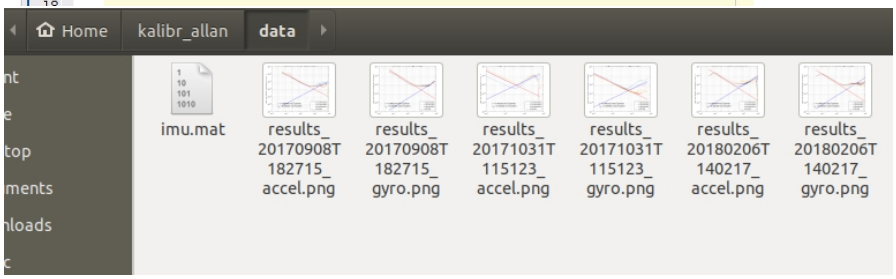
```
Param();

// time
int imu_frequency = 200;
int cam_frequency = 30;
double imu_timestep = 1./imu_frequency;
double cam_timestep = 1./cam_frequency;
double t_start = 0.;
double t_end = 20;  //  20 s

// noise
double gyro_bias_sigma = 1.0e-5;
double acc_bias_sigma = 0.0001;

double gyro_noise_sigma = 0.015;    // rad/s * 1/sqrt(hz)
double acc_noise_sigma = 0.019;      //  m/(s^2) * 1/sqrt(hz)

double pixel_noise = 1;              // 1 pixel noise

// cam f
double fx = 460;
double fy = 460;
double cx = 255;
double cy = 255;
double image_w = 640;
double image_h = 640;


// 外参数
Eigen::Matrix3d R_bc;   // cam to body
Eigen::Vector3d t_bc;     // cam to body
```

可以在这里修改误差 但是由于我无法使用 ros 直接用的群里生成好的 mat 文件

那就来验证一下标定结果

他还把文件名字改了 改了我也改一下吧。。 把文件放入对应路径：





这就跑完了 生成 result 文件了

```
>> cd ../../..
>> cd kalibr_allan/
>> cd matlab/
>> SCRIPT_allan_matparallel
opening the mat file.
loading timeseries.
imu frequency of 200.00.
sample period of 0.00500.
calculating allan deviation.
Elapsed time is 595.904973 seconds.
saving to: results_20220225T143756.mat
done saving!
>>
```

对生成的 result 文件进行下一步处理

Editor - /home/holo/kalibr_allan/matlab/SCRIPT_process_results.m*

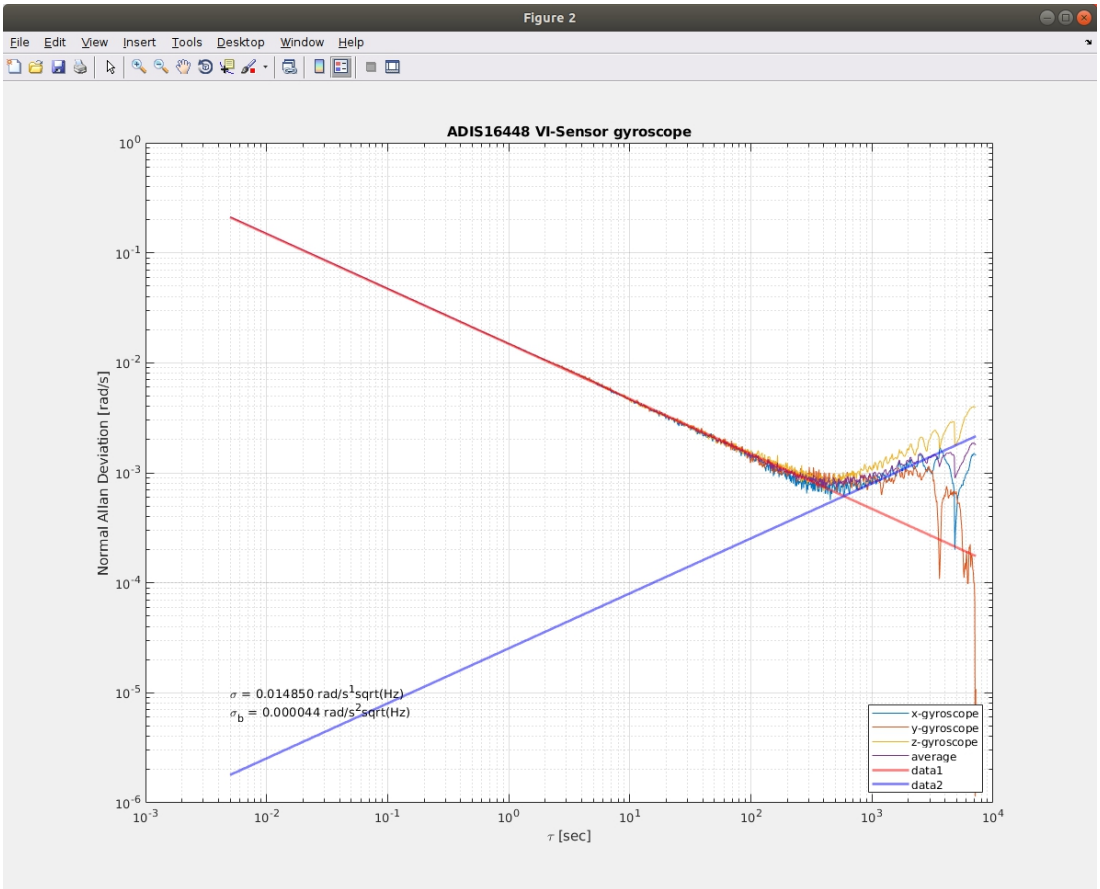
SCRIPT_process_results.m*

This file can be opened as a Live Script. For more information, see Creating Live Scripts.

```
%% Initalization
close all
clear all

% Read in our toolboxes
addpath('functions')
addpath('functions/allan_v3')

% Our bag information
%titlestr = 'XSENS MTi-G-710';
%mat_path = '../data/bags/results_20170908T182715.mat';

%titlestr = 'Tango Yellowstone #1';
%mat_path = '../data/bags/results_20171031T115123.mat';

titlestr = 'ADIS16448 VI-Sensor';
mat_path = '../data/results_20220225T143756.mat';

% Load the mat file (should load "data_imu" matrix)
fprintf('=> opening the mat file.\n')
load(mat_path);
```

也是修改路径和对应文件名 在运行一次 出结果了

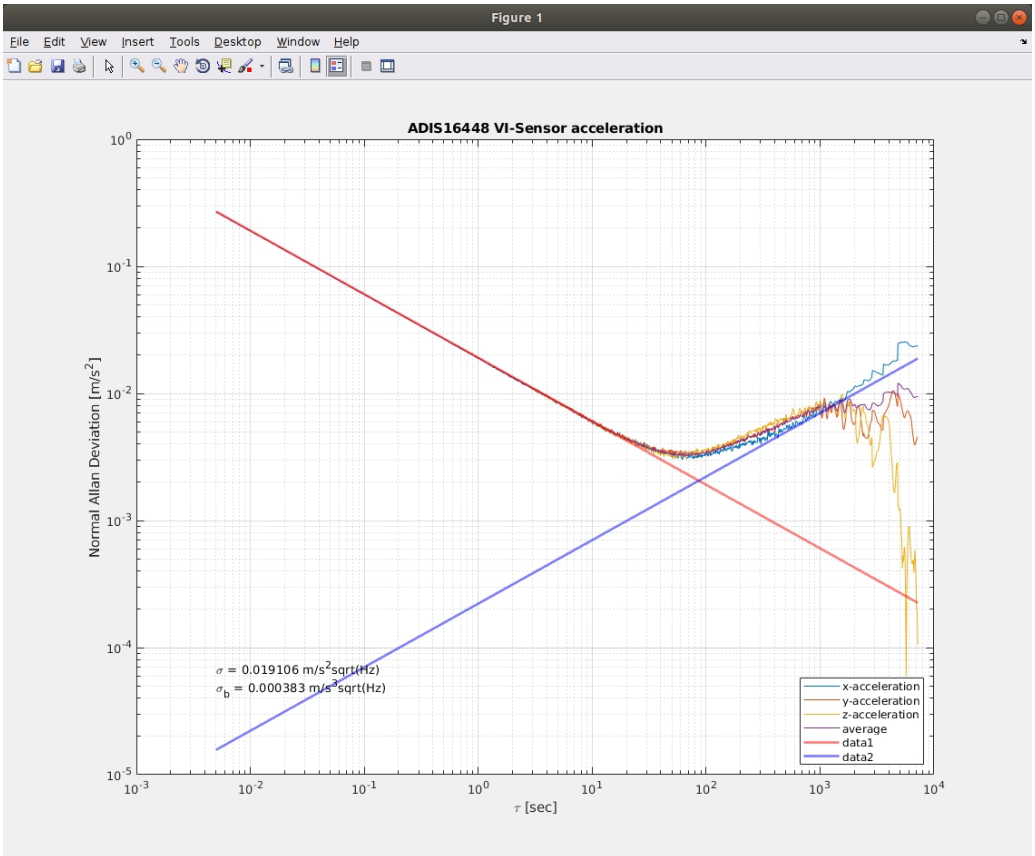
gyro 的：



高斯白噪声：0.014 设的是 0.015 很接近

bias 随机游走：0.000044 设的是 0.00001 这也还可以

acc 的：



高斯白噪声：0.019 设的是 0.019 不考虑后面的位数是一样的

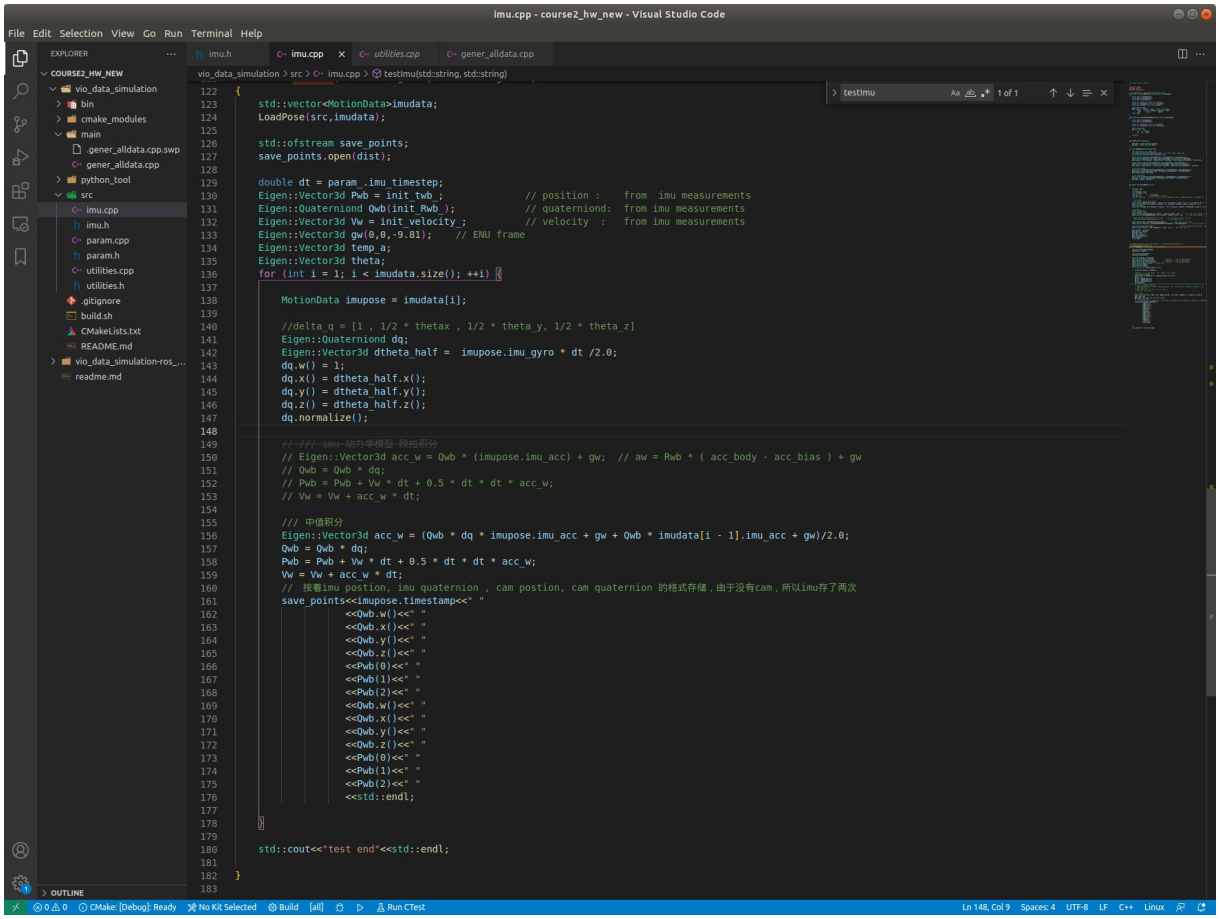
bias 随机游走：0.000383 设的是 0.0001 也还 ok

标定的很准

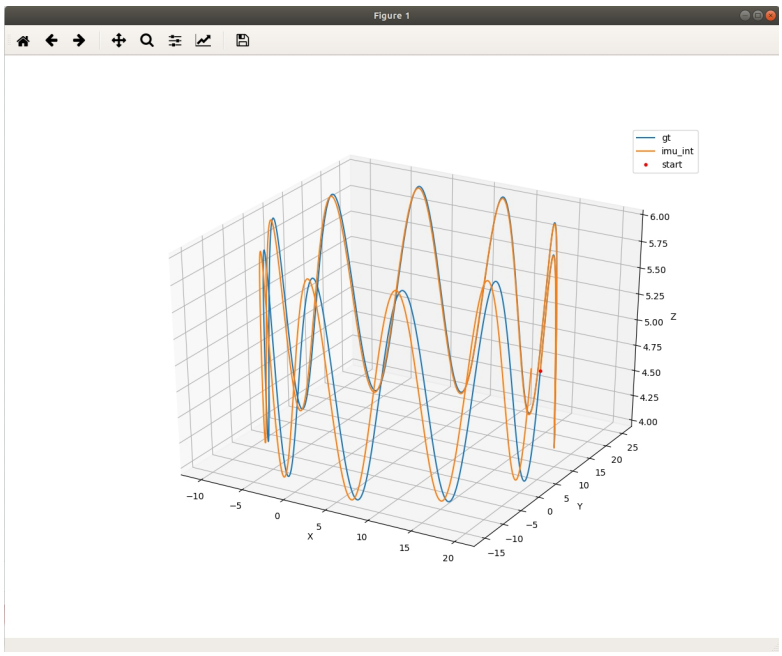
第二题

❷ 将 IMU 仿真代码中的欧拉积分替换成中值积分。

通读了一遍源码 发现在这里进行修改：



先运行一遍：

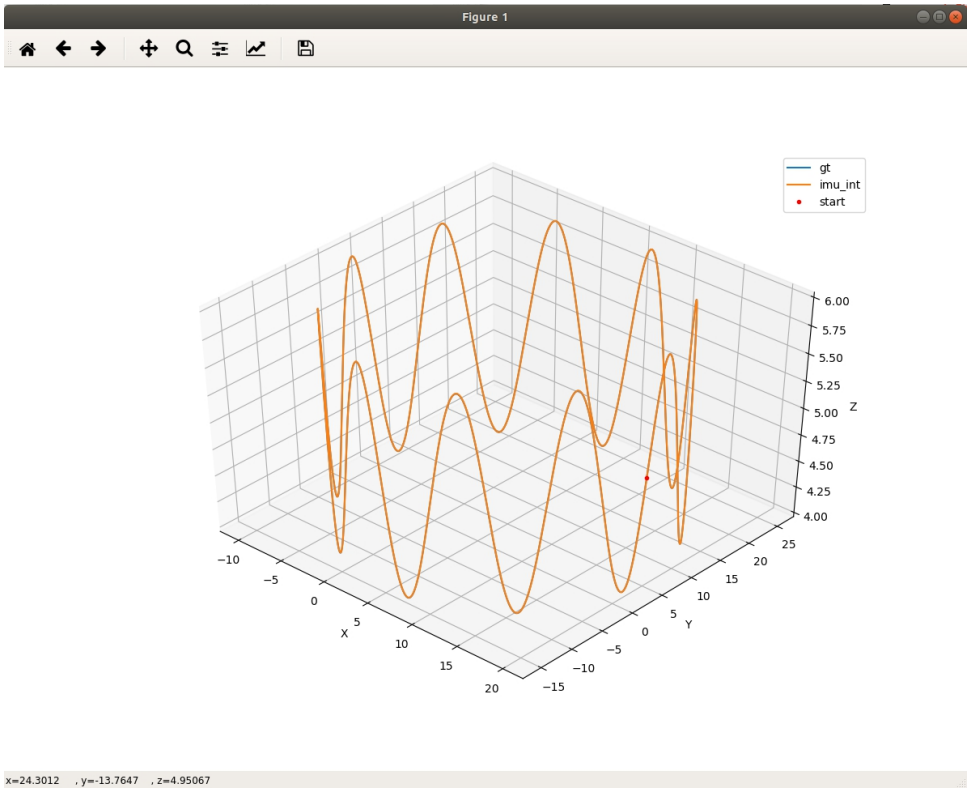


可以看出欧拉积分的特点：越来越拉跨

改成中值积分：

```cpp
//delta_q = [1 , 1/2 * thetax , 1/2 * theta_y, 1/2 * theta_z]
Eigen::Quaterniond dq;
Eigen::Vector3d dtheta_half =  (imupose.imu_gyro + imudata[i - 1].imu_gyro)/2.0 * dt /2.0;
dq.w() = 1;
dq.x() = dtheta_half.x();
dq.y() = dtheta_half.y();
dq.z() = dtheta_half.z();
dq.normalize();

// /// imu 动力学模型 欧拉积分
// Eigen::Vector3d acc_w = Qwb * (imupose.imu_acc) + gw;  // aw = Rwb * ( acc_body - acc_bias ) + gw
// Qwb = Qwb * dq;
// Pwb = Pwb + Vw * dt + 0.5 * dt * dt * acc_w;
// Vw = Vw + acc_w * dt;

/// 中值积分
Eigen::Vector3d acc_w = (Qwb * dq * imupose.imu_acc + gw + Qwb * imudata[i - 1].imu_acc + gw)/2.0;
Qwb = Qwb * dq;
Pwb = Pwb + Vw * dt + 0.5 * dt * dt * acc_w;
Vw = Vw + acc_w * dt;
```

再运行一遍：



基本完美的重合了~

选做题：

A continuous-time representation for visual-inertial fusion with application to rolling shutter cameras

本文通篇讲解了一种利用卷帘式相机和 imu 传感器进行视觉和惯性融合的方法

B 样条标准基函数公式：

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

然后重写成累加的形式：

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

利用这个公式：

$$\Omega_i = \log(\mathbf{T}_{w,i-1}^{-1}\mathbf{T}_{w,i}) \in \mathfrak{se}3$$

将 B 样条标准基函数公式重写：

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

其中 T 为李代数模式 写成这种形式的原因是方便后面的求导 如果可以求导就直接对应了 imu 的测量模型保证了连续性

本文以三次 B 样条为例 假设四个控制点分布在一段时间间隔内 采用本文自行定义的类似归一化的方法将控制时间点转换为同一时间序列 为的是将累积基函数进行时间的求导

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

C 就是本文随便举的一个系数的例子 u 代表不同次项

位姿变换就是这样：

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

于是可以求出位姿对于时间的一阶和二阶导数

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

接下来就是构建误差函数了

构建之前先建立两个相机之间的投影对应关系 通过图像坐标进行对比

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

加速度计和陀螺仪的测量 就是老师上课讲的公式：

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

误差函数 最小化这个误差函数估计出待求参数

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

得到的是样条插值结果 更好的相机运动轨迹