



深蓝学院  
shenlanxueyuan.com

## vio第八期第二章作业讲解



主讲人 梁章科



# 作业内容

## 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/gaowenliang/imu_utils)  
[https://github.com/rpng/kalibr\\_allan](https://github.com/rpng/kalibr_allan)  
...
- ② 将 IMU 仿真代码中的欧拉积分替换成中值积分。

## 提升作业，选做

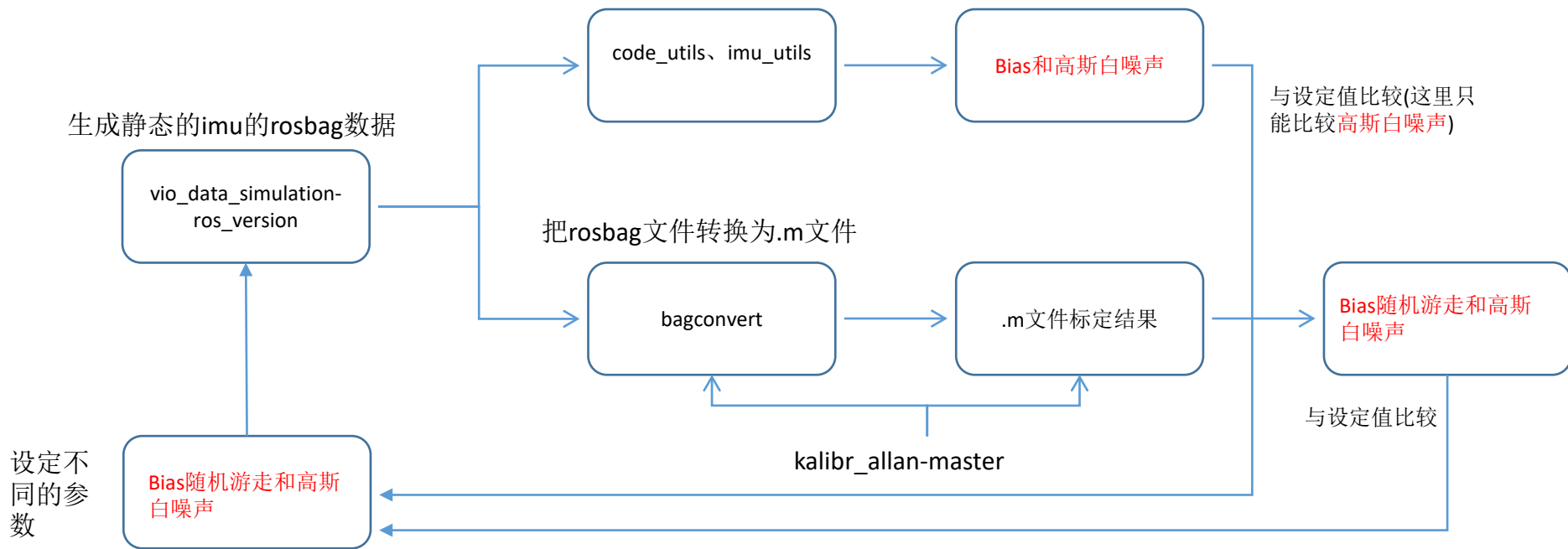
阅读从已有轨迹生成 imu 数据的论文，撰写总结推导：

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

# 基础作业1

生成静止 imu 数据，使用imu\_utils或者kalibr\_allan完成Allen方差标定

回放rosvbag文件，用imu\_utils标定allen方差      imuData\_imu\_param.yaml



# 基础作业1

## imu\_utils

### IMU Noise Values

Parameter	YAML element	Symbol	Units
Gyroscope "white noise"	gyr_n	$\sigma_g$	$\frac{rad}{s} \frac{1}{\sqrt{Hz}}$
Accelerometer "white noise"	acc_n	$\sigma_a$	$\frac{m}{s^2} \frac{1}{\sqrt{Hz}}$
Gyroscope "bias Instability"	gyr_w	$\sigma_{bg}$	$\frac{rad}{s} \sqrt{Hz}$
Accelerometer "bias Instability"	acc_w	$\sigma_{ba}$	$\frac{m}{s^2} \sqrt{Hz}$

## Kalibr\_allan

### IMU Noise Values

Parameter	YAML element	Symbol	Units
Gyroscope "white noise"	gyroscope_noise_density	$\sigma_g$	$\frac{rad}{s} \frac{1}{\sqrt{Hz}}$
Accelerometer "white noise"	accelerometer_noise_density	$\sigma_a$	$\frac{m}{s^2} \frac{1}{\sqrt{Hz}}$
Gyroscope "random walk"	gyroscope_random_walk	$\sigma_{bg}$	$\frac{rad}{s^2} \frac{1}{\sqrt{Hz}}$
Accelerometer "random walk"	accelerometer_random_walk	$\sigma_{ba}$	$\frac{m}{s^3} \frac{1}{\sqrt{Hz}}$

IMU仿真代码设定了陀螺仪和加速度计的**高斯白噪声**和**bias随机游走**，imu\_utils标定得到陀螺仪和加速度计的**高斯白噪声**和**bias**，kalibr\_allan标定得到陀螺仪和加速度计的**高斯白噪声**和**bias随机游走**，因此如果选用imu\_utils标定，只需对比其高斯白噪声与设定值；如果选用kalibr\_allan标定，则需要对比高斯白噪声和bais随机游走。

# 基础作业1

## 1. 下载和编译code\_utils

编译前需要安装好Ceres：新建一个workspace文件夹，在其目录下新建src文件夹，输入下面命令下载和编译code\_utils

```
cd ~/workspace/src
```

```
git clone https://github.com/gaowenliang/code_utils
```

```
cd ~/workspace
```

```
catkin_make
```

编译过程如果遇到 `fatal error: backward.hpp: No such file or directory`，请在code\_utils的CMakeLists.txt加入这句代码 `include_directories("include/code_utils")`

# 基础作业1-imu\_utils

## 2. 下载和编译imu\_utils

```
cd ~/workspace/src
```

```
git clone https://github.com/gaowenliang/imu_utils
```

```
cd ~/workspace
```

```
catkin_make
```

编译过程如果出现错误 `error: aggregate 'std::ofstream out_t' has incomplete type and cannot be defined, std::ofstream out_t;`

请在imu\_utils/src/imu\_an.cpp 添加头文件 `#include <fstream>`

# 基础作业1-imu\_utils

3. 编译vio\_data\_simulation-ros\_version, 生成IMU静态rosvag数据

- 把文件夹vio\_data\_simulation-ros\_version拷到workspace/src目录下
- 修改vio\_data\_simulation-ros\_version/src/gener\_alldata.cpp 里面的bag\_path (IMU rosvag保存的路径)
- 编译
- 生成IMU静态rosvag数据, 输入

```
cd ~/workspace
```

```
source devel/setup.bash
```

```
rosvun vio_data_simulation vio_data_simulation_node
```

# 基础作业1-imu\_utils

使用imu\_utils标定Allen方差

- 新开一个terminal, 输入roscore
- 新开一个terminal, 输入下面命令, 进行Allen方差标定

```
cd ~/workspace
```

```
source devel/setup.bash
```

```
roslaunch imu_utils imuData.launch
```

- 新开一个terminal, 如果imu.bag在workspace目录下(如果是其他文件夹, 请修改下面的第一条命令), 输入下面命令, 播放rosbag

```
cd ~/workspace
```

```
rosbag play -r 500 imu.bag
```



# 基础作业1-imu\_utils

imu\_utils标定Allen方  
差，结果分析

- 在imu\_utils/data目录下的  
imuData\_imu\_param.yaml有加速度计和陀螺仪的bias和高斯白噪声标定结果，请对比  
vio\_data\_simulation-  
ros\_version/src/param.h 参数设置，请注意  
两者结果的单位转换

```
%YAML:1.0
---
type: IMU
name: imuData
Gyr:
  unit: " rad/s"
  avg-axis:
    gyr_n: 2.1116397962239117e-01
    gyr_w: 1.0778209496476461e-03
  x-axis:
    gyr_n: 2.1125251766122802e-01
    gyr_w: 1.1575713207586493e-03
  y-axis:
    gyr_n: 2.0828542289387936e-01
    gyr_w: 9.3308792723813781e-04
  z-axis:
    gyr_n: 2.1395399831206610e-01
    gyr_w: 1.1428036009461509e-03
Acc:
  unit: " m/s^2"
  avg-axis:
    acc_n: 2.6934224907886045e-01
    acc_w: 3.5219409959472473e-03
  x-axis:
    acc_n: 2.7028927704466654e-01
    acc_w: 4.2242321188553245e-03
  y-axis:
    acc_n: 2.7227764084582862e-01
    acc_w: 3.1523777537398904e-03
  z-axis:
    acc_n: 2.6545982934608614e-01
    acc_w: 3.1892131152465261e-03
```

对比高斯白  
噪声，需除  
以 $\sqrt{\text{Hz}}$

```
// 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 = 3600 * 4; // 4 hours for allan

// noise
double gyro_bias_sigma = 0.00005;
double acc_bias_sigma = 0.0005;

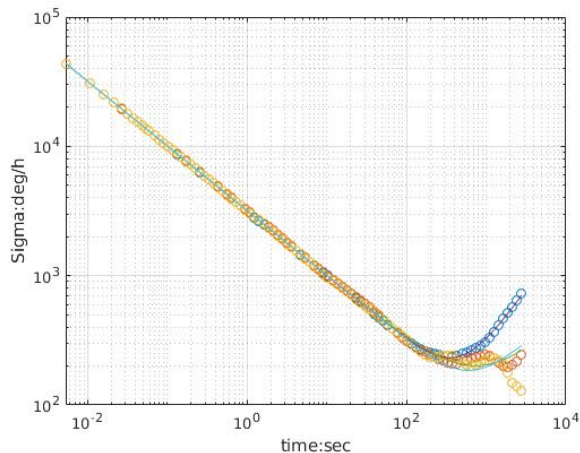
//double gyro_bias_sigma = 1.0e-5;
//double acc_bias_sigma = 0.0001;

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

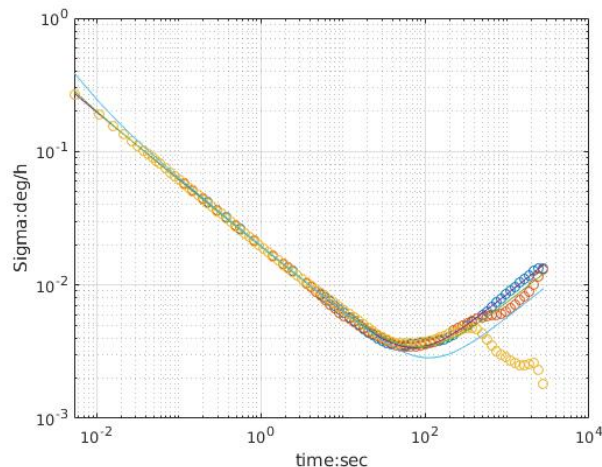
# 基础作业1-imu\_utils

利用imu\_utils/scripts/draw\_allan.m(注意修改里面的文件路径)分别画出加速度计和陀螺仪的Allen方差

陀螺仪



加速度计



关于Allen方差标定更详细的内容请参考这篇文章: <https://zhuanlan.zhihu.com/p/158927004>

# 基础作业1-kalibr\_allan

## kalibr\_allan安装与编译:

- 请提前安装好MATLAB，最好是2018版本，输入下面命令：

```
cd ~/workspace/src
```

```
git clone
```

```
https://github.com/rpng/kalibr_allan
```

```
cd ~/workspace
```

```
catkin_make
```

- 编译过程中如果出现找不到MATLAB，说明你的MATLAB文件没有安装在系统默认的地方 (/usr/local/bin)，

- 那么请打开

```
workspace/src/kalibr_allan/bagconvert/cmake/FindMatlab.cmake， 找到这句话
```

```
find_program(MATLAB_EXE_PATH matlab  
PATHS /usr/local/bin)
```

- 修改PATHS地址为你的MATLAB安装目录的bin地址，例如

```
find_program(MATLAB_EXE_PATH matlab  
PATHS
```

```
/usr/local/MATLAB/R2018a/bin)
```

- 保存，然后输入下面命令重新编译：

```
rm -rf build/ && catkin_make
```

# 基础作业1-kalibr\_allan

把rosvbag文件转换为.m文件

- 接一页ppt的命令，继续输入

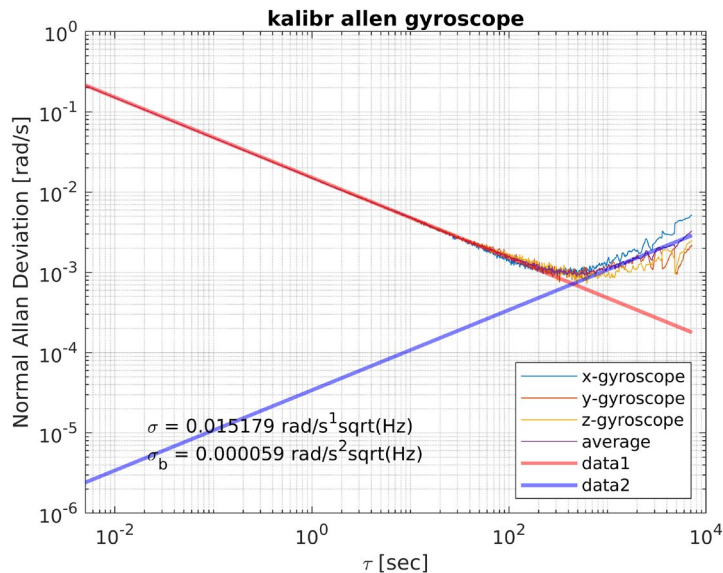
```
source devel/setup.bash
```

```
rosvrun bagconvert bagconvert imu.bag imu
```

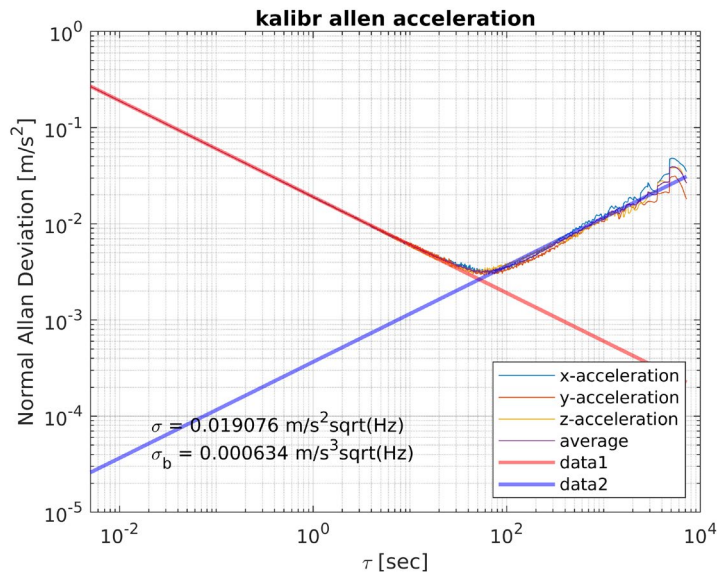
- 上面这条命令要求imu.bag在workspace目录下，生成的imu.mat也会在这个目录下
- 在src/kalibr\_allan/matlab目录下首先修改 SCRIPT\_allan\_matparallel.m里面的imu.mat文件路径mat\_path，运行，会在src/kalibr\_allan/data目录下生成一个result开头的mat文件，然后修改SCRIPT\_process\_results.m的mat\_path路径为result开头的文件路径，运行，就能得到kalibr\_allan标定的高斯白噪声和bias随机游走

# 基础作业1-kalibr\_allan

## 陀螺仪



## 加速度计



标定结果如果有NaN，适当增大对应的仿真参数

最后设置IMU不同的bias随机游走和高斯白噪声，用imu\_utils或者kalibr\_allan生成Allan方差标定曲线，对比结果

# 基础作业2-中值积分

中值积分替换代码原有的欧拉积分：

- 首先编译vio\_data\_simulation:

```
cd vio_data_simulation-master
```

```
mkdir build
```

```
cd build
```

```
cmake ..
```

```
make
```

```
cd ../bin
```

```
./data_gen
```

```
cd ../python_tool
```

```
python draw_trajcory.py
```

- 上述命令会画出欧拉积分下的轨迹图
- 参考第40页ppt，把vio\_data\_simulation/src/imu.cpp的欧拉积分替换成中值积分，重新编译，画出中值积分下的轨迹图

## 运动模型的离散积分——中值法

使用 mid-point 方法，即两个相邻时刻  $k$  到  $k+1$  的位姿是用两个时刻的测量值  $\mathbf{a}, \boldsymbol{\omega}$  的平均值来计算。

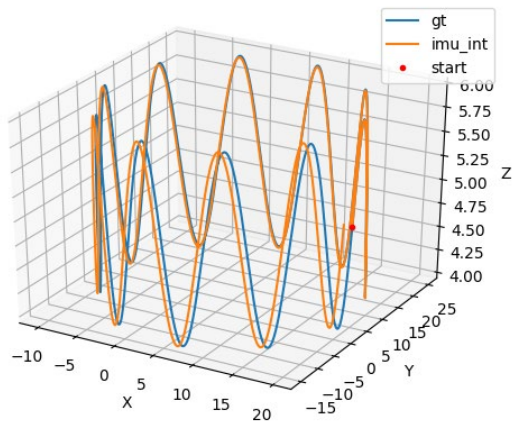
$$\begin{aligned}\mathbf{p}_{wb_{k+1}} &= \mathbf{p}_{wb_k} + \mathbf{v}_k^w \Delta t + \frac{1}{2} \mathbf{a} \Delta t^2 \\ \mathbf{v}_{k+1}^w &= \mathbf{v}_k^w + \mathbf{a} \Delta t \\ \mathbf{q}_{wb_{k+1}} &= \mathbf{q}_{wb_k} \otimes \left[ \frac{1}{2} \boldsymbol{\omega} \delta t \right]\end{aligned}\quad (32)$$

其中，

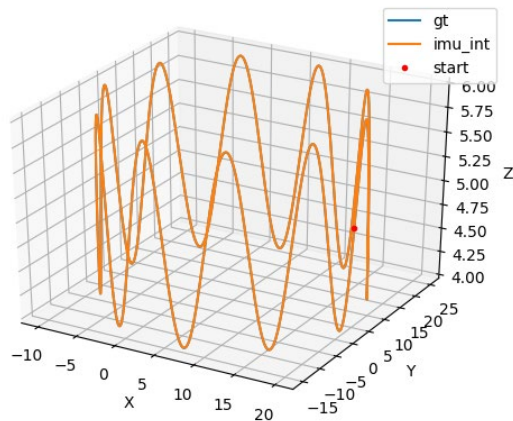
$$\begin{aligned}\mathbf{a} &= \frac{1}{2} \left[ \mathbf{q}_{wb_k} \left( \mathbf{a}^{b_k} - \mathbf{b}_k^a \right) - \mathbf{g}^w + \mathbf{q}_{wb_{k+1}} \left( \mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a \right) - \mathbf{g}^w \right] \\ \boldsymbol{\omega} &= \frac{1}{2} \left[ \left( \boldsymbol{\omega}^{b_k} - \mathbf{b}_k^g \right) + \left( \boldsymbol{\omega}^{b_{k+1}} - \mathbf{b}_k^g \right) \right]\end{aligned}\quad (33)$$

# 基础作业2-中值积分

欧拉积分



中值积分



# 提升作业-阅读论文

阅读论文之前，需要先了解贝塞尔曲线和B样条(B-Spline)

- **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

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

- **贝塞尔曲线缺点**：

- ☐ 阶次=控制点个数-1

- ☐ 移动一个控制点，整段曲线都会变化B样条

B样条(B-Spline)可解决上述的缺点

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

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



# 提升作业-阅读论文

●目的:

□用离散的位姿利用B样条拟合曲线方程，对曲线求导获得IMU数据

□Vio中相机和IMU时间戳同步

●核心理论：将离散的位姿利用B样条拟合成连续的位姿轨迹曲线

B-Spline累积公式

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

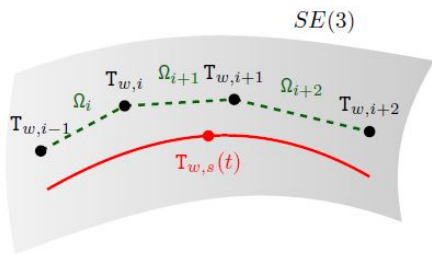


Fig. 3: Geometric interpretation of the cubic spline interpolation given by formula (4). The cumulative formulation uses one absolute control pose  $\mathbf{T}_{w,i-1}$  and three incremental control poses  $\Omega_i, \Omega_{i+1}, \Omega_{i+2}$  to compute the interpolated pose  $\mathbf{T}_{w,s}$ .

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

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

K=4的基函数  $\tilde{\mathbf{B}}(u) = \mathbf{C} \begin{bmatrix} 1 \\ u \\ u^2 \\ u^3 \end{bmatrix}$ ,  $\dot{\tilde{\mathbf{B}}}(u) = \frac{1}{\Delta t} \mathbf{C} \begin{bmatrix} 0 \\ 1 \\ 2u \\ 3u^2 \end{bmatrix}$ ,  $\ddot{\tilde{\mathbf{B}}}(u) = \frac{1}{\Delta t^2} \mathbf{C} \begin{bmatrix} 0 \\ 0 \\ 2 \\ 6u \end{bmatrix}$ ,  $\mathbf{C} = \frac{1}{6} \begin{bmatrix} 6 & 0 & 0 & 0 \\ 5 & 3 & -3 & 1 \\ 1 & 3 & 3 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

# 提升作业-阅读论文

位姿曲线求导

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

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

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

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

对曲线求导获取IMU预测的角速度和加速度

$$\text{Gyro}(u) = (\mathbf{R}_{w,s}^T(u) \cdot \dot{\mathbf{R}}_{w,s}(u))^V + \text{bias} \quad \leftarrow \text{Gyro公式有误} \quad \text{Gyro}(u) = \mathbf{R}_{w,s}^T(u) \cdot \dot{\mathbf{R}}_{w,s}(u) + \text{bias}, \quad (8)$$

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

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

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

这篇文章对论文公式有比较详细的推导:

[https://udel.edu/~pgeneva/downloads/notes/2018\\_notes\\_mueffler2017arxiv.pdf](https://udel.edu/~pgeneva/downloads/notes/2018_notes_mueffler2017arxiv.pdf)





深蓝学院  
shenlanxueyuan.com

感谢各位聆听 !  
Thanks for Listening

