深蓝学院
shenlanxueyuan.com

第十一章：IMU传感器作业思路提示

主讲人 于子平

【作业内容】

深蓝学院
shenlanxueyuan.com

① 设置IMU仿真代码中不同的参数，生成Allan方差标定曲线。

https://github.com/gaowenliang/imu_utils

https://github.com/rpng/kalibr_allan

② 将IMU仿真代码中欧拉积分替换成中值积分。

③ 提升作业：阅读文献，撰写总结推导

Lovegrove S, Patron-Perez A, Sibley G. Spline Fusion: A continuous-time representation for visual-inertial fusion with application to rolling shutter cameras[C]//BMVC. 2013, 2(5): 8.

Allan方差标定曲线

深蓝学院
shenlanxueyuan.com

- Allan方差标定的原因：IMU采集数据的时候会产生两种误差：确定性误差和随机误差。确定性误差可以通过六面法和最小二乘法求解，而随机误差是随时间变化的，主要包含高斯白噪声和bias随机游走，通过标定方差来减小随机误差的干扰。

- Allan方差标定的方法：

有两种工具：imu_utils和kalibr_allan

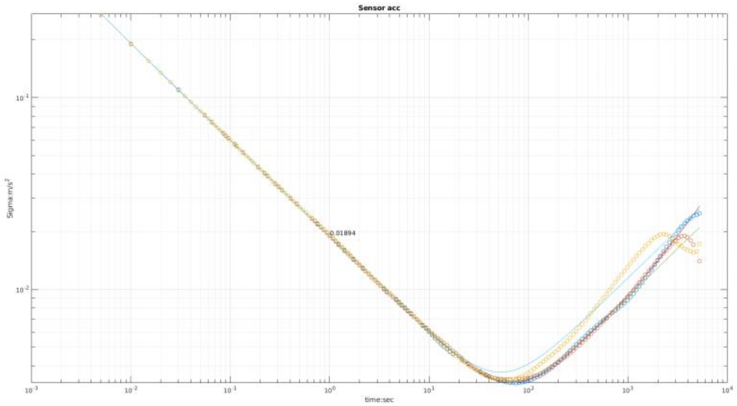
Allan方差标定

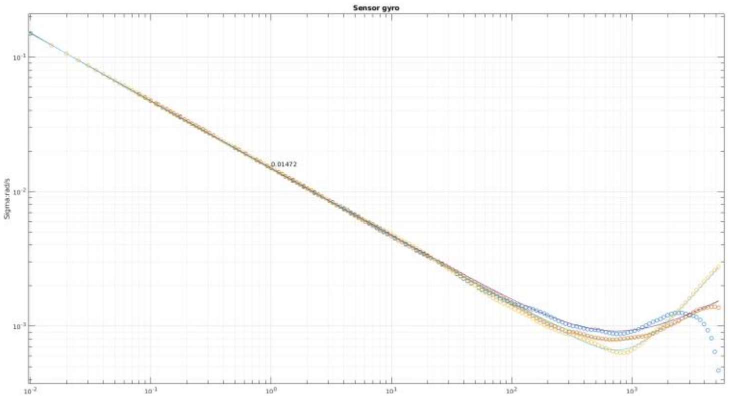
仿真数据的产生：
在vio_data_simulation-ros_version和vio_data_simulation的代码中可以产生imu仿真数据。设定每一个时刻的位置的运动方程以及旋转的运动方程，对方程进行求导，得到每一个timestamp的acc和gyro。基于ros版本的会生成一个imu.bag，即记录了imu数据的rosbag。

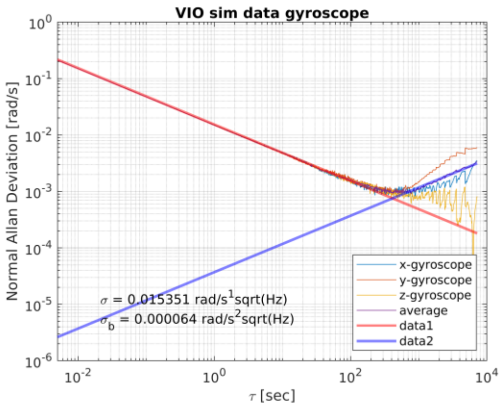
方差标定：
Imu_utils：启动launch文件开始接受并分析imu话题数据，然后开始回放rosbag，最后通过matlab绘制allan曲线。

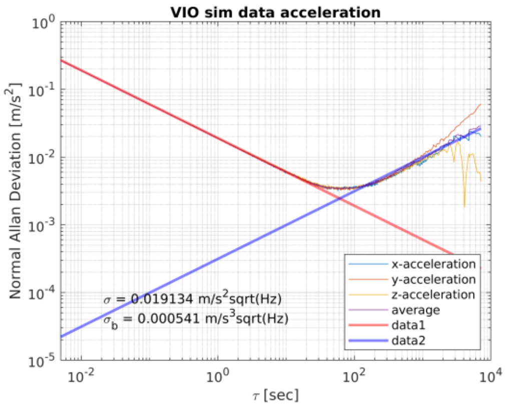
Kalibr_allan：直接将rosbag转换成mat文件，然后用matlab进行分析。

Allan方差标定


Sensor acc
0.01894
time:sec


Sensor gyro
0.01472


VIO sim data gyroscope
Normal Allan Deviation [rad/s]
τ [sec]
σ = 0.015351 rad/s^{1}sqrt(Hz)
σ_b = 0.000064 rad/s^{2}sqrt(Hz)
x-gyroscope
y-gyroscope
z-gyroscope
average
data1
data2


VIO sim data acceleration
Normal Allan Deviation [m/s^2]
τ [sec]
σ = 0.019134 m/s^{2}sqrt(Hz)
σ_b = 0.000541 m/s^{3}sqrt(Hz)
x-acceleration
y-acceleration
z-acceleration
average
data1
data2

Allan方差标定

进行imu_utils与kalibr_allan两种不同标定方法的结果对比，以及设定不同大小的误差，进行多次实验比较。

误差类型	仿真设定值	imu_utils	kalibr_allan
加速度计白噪声($\frac{m}{s^2}\frac{1}{\sqrt{Hz}}$)	0.019	0.0190	0.019169
加速度计随机游走($\frac{m}{s^3}\frac{1}{\sqrt{Hz}}$)	0.0005	0.00024269（估计）	0.000571
陀螺仪白噪声($\frac{rad}{s}\frac{1}{\sqrt{Hz}}$)	0.015	0.0148	0.015028
陀螺仪随机游走($\frac{rad}{s^2}\frac{1}{\sqrt{Hz}}$)	0.00005	0.00005596（估计）	0.000041

Allan方差标定

加速度计高斯白噪声大小	实际大小	标定结果	标定结果连续化后结果
PPT默认噪声	0.019	2.6764053623067113e-01	0.018925043808911142

加速度计Bias随机游走大小	实际大小	标定结果	标定结果连续化后结果
PPT默认噪声	5e-4	3.6174135533268148e-03	0.051157953078270306

高斯白噪声：

$$\sigma = \sigma_d \cdot \sqrt{\triangle t}$$

bias随机游走：

$$\sigma_b = \sigma_{bd} \frac{1}{\sqrt{\triangle t}}$$

IMU中值积分

IMU传感器可以测量当前Body的角速度和加速度，
因此对IMU测量进行积分，可以得到Body的当前P、V、Q

$$\tilde{\boldsymbol{\omega}}^b = \boldsymbol{\omega}^b + \mathbf{b}^g + \mathbf{n}^g$$
$$\tilde{\mathbf{a}}^b = \mathbf{q}_{bw}(\mathbf{a}^w + \mathbf{g}^w) + \mathbf{b}^a + \mathbf{n}^a$$

测量模型

$$\mathbf{p}_{wb_{k+1}} = \mathbf{p}_{wb_k} + \mathbf{v}_k^w \Delta t + \frac{1}{2}\mathbf{a}\Delta t^2$$
$$\mathbf{v}_{k+1}^w = \mathbf{v}_k^w + \mathbf{a}\Delta t$$
$$\mathbf{q}_{wb_{k+1}} = \mathbf{q}_{wb_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\boldsymbol{\omega}\delta t \end{bmatrix}$$

运动模型

$$\mathbf{a} = \mathbf{q}_{wb_k}\left(\mathbf{a}^{b_k} - \mathbf{b}_k^a\right) - \mathbf{g}^w$$
$$\boldsymbol{\omega} = \boldsymbol{\omega}^{b_k} - \mathbf{b}_k^g$$

欧拉积分

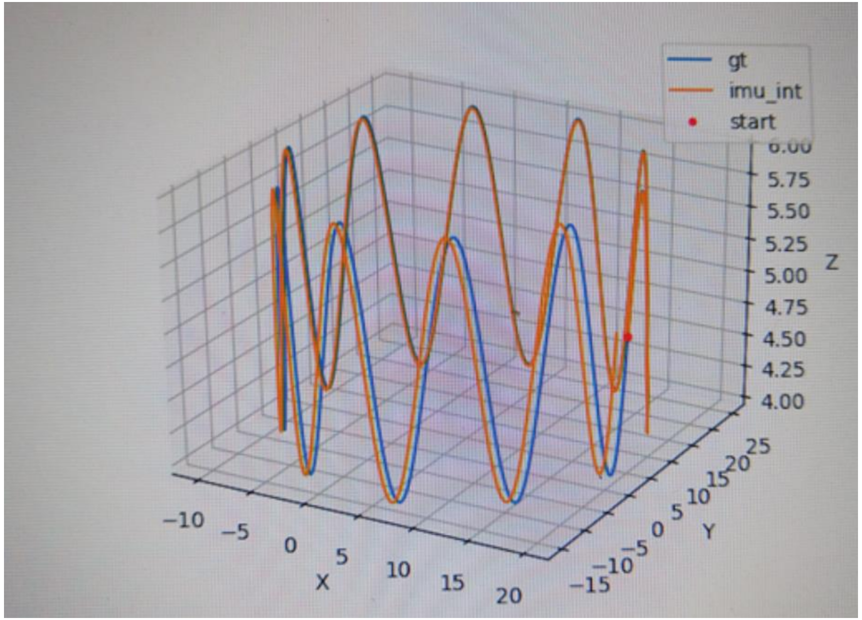
中值积分

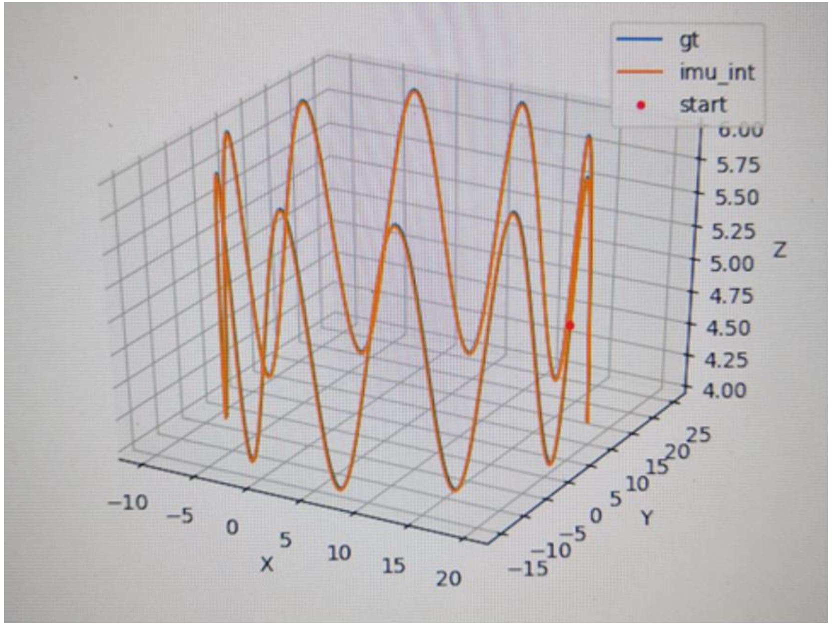
$$\mathbf{a} = \frac{1}{2}\left[\mathbf{q}_{wb_k}\left(\mathbf{a}^{b_k} - \mathbf{b}_k^a\right) - \mathbf{g}^w + \mathbf{q}_{wb_{k+1}}\left(\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a\right) - \mathbf{g}^w\right]$$
$$\boldsymbol{\omega} = \frac{1}{2}\left[\left(\boldsymbol{\omega}^{b_k} - \mathbf{b}_k^g\right)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

```
// ***************** 欧拉积分 ********************************************
// delta_q = [1 , 1/2 * thetax , 1/2 * theta_y, 1/2 * theta_z]
// Eigen::Quaterniond dq;
// Eigen::Vector3d dtheta_half = imupose.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::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 ********************************************
MotionData imupose_pre = imudata[i - 1];
Eigen::Vector3d mid_omega = (imupose_pre.imu_gyro + imupose.imu_gyro) * 0.5;
Eigen::Vector3d dtheta_half = mid_omega * dt / 2.0;
Eigen::Quaterniond dq(1, dtheta_half.x(), dtheta_half.y(), dtheta_half.z());
dq.normalize();

Eigen::Quaterniond Qwb_newx = Qwb * dq;

Eigen::Vector3d acc_w = 0.5 * (Qwb * (imupose.imu_acc) + gw + Qwb_newx * (imupose_pre.imu_acc) + gw);
Vw = Vw + acc_w * dt;
Pwb = Pwb + Vw * dt + 0.5 * dt * dt * acc_w;
Qwb = Qwb_newx;
```

IMU中值积分

深蓝学院
shenlanxueyuan.com





基于累积3次B-Spline的SE3插值

深蓝学院 shenlanxueyuan.com

累计基函数的 B 次样条曲线如下：

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

其中，$\mathbf{p}_i \in \mathbb{R}^N$，表示 t_i 时刻的控制点，$B_{i,k}(t)$ 为基函数，公式如下：

$$B_{i,0}(x)=\begin{cases}1, & t_i \le x < t_{i+1}\\ 0, & otherwise\end{cases} \tag{2}$$

$$B_{i,p}(x)=\frac{x-t_i}{t_{i+p}-t_i}B_{i,p-1}(x)+\frac{t_{i+p+1}-x}{t_{i+p+1}-t_{i+1}}B_{i+1,p-1}(x) \tag{3}$$

公式(1)可以改写为：

$$\mathbf{p}(t)=\mathbf{p}_0\tilde{B}_{0,k}(t)+\sum_{i=0}^{n}(\mathbf{p}_i-\mathbf{p}_{i-1})\tilde{B}_{i,k}(t) \tag{4}$$

其中，$\tilde{B}_{i,k}(t)=\sum_{j=i}^{n}B_{j,k}(t)$ 为累计基函数，然后通过用控制点之间的对数变换 $\Omega_i=\log\left(\mathbf{T}_{w,i-1}^{-1},\mathbf{T}_{w,i}\right)$ 操作代替控制点之差来描述 SE3 中的轨迹，公式(4)可以进一步变换成：

$$\mathbf{T}_{w,s}(t)=\exp\left(\tilde{B}_{0,k}(t)\log\mathbf{T}_{w,0}\right)\prod_{i=1}^{n}\exp\left(\tilde{B}_{0,k}(t)\Omega_i\right) \tag{5}$$

其中，$\mathbf{T}_{w,s}(t)\in SE3$ 是样条曲线上 t 时刻的位姿，而 $\mathbf{T}_{w,i}\in SE3$ 就是世界坐标系下控制点位姿。

基于累积3次B-Spline的SE3插值

深蓝学院 shenlanxueyuan.com

IMU的误差表达式如下：

$$\boldsymbol{\omega}_m = \boldsymbol{\omega} + \mathbf{b}_\omega + \mathbf{n}_\omega$$

$$\mathbf{a}_m = \mathbf{a} + \mathbf{g} + \mathbf{b}_a + \mathbf{n}_a$$

用B样条可以得到连续时间下的陀螺仪角速度和加速度计的加速度值，如下：

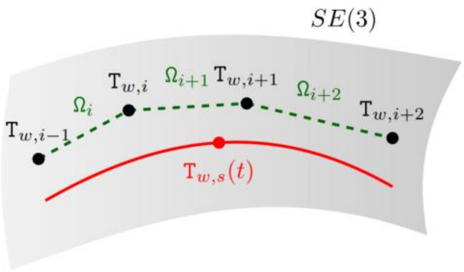
$$\boldsymbol{\omega}_m(u(t)) = \left({}_I^G\mathbf{R}(u(t))^\top \, {}_I^G\dot{\mathbf{R}}(u(t))\right)^\vee + \mathbf{b}_\omega + \mathbf{n}_\omega$$

$$\mathbf{a}_m(u(t)) = {}_I^G\mathbf{R}(u(t))^\top \left({}^G\ddot{\mathbf{p}}_I(u(t)) + {}^G\mathbf{g}\right) + \mathbf{b}_a + \mathbf{n}_a$$

其中，R，P为矩阵T的子集

基于累积3次B-Spline的SE3插值

对于四次累计 B 样条曲线，$t\in[t_i,t_{i+1}]$这段时间中一共有$[t_{i-1},t_i,\ t_{i+1},\ t_{i+2}]$四个控制点来确定时刻 t 的样条曲线上的值，使用 $s(t)=(t-t_0)/\Delta t$ 来表示平均时间的函数，控制点的时刻 t_i 就可以由平均时间函数 $s_i\in[0,1,\cdots,n]$来表示，对于时间 $s_i\leq s(t)<s_{i+1}$，定义 $u(t)=s(t)-s_i$ 来表示。重写矩阵形式的 B 样条曲线以及它的一阶二阶微分函数如下：



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

样条轨迹上的位姿可以定义为：

$$\mathbf{T}_{w,s}(u)=\mathbf{T}_{w,i-1}\prod_{j=1}^{3}\exp\left(\tilde{\mathbf{B}}(u)_j\,\Omega_{i+j}\right) \tag{7}$$

其中 i 下标表示时间 t 所在的时间间隔区间，求上式的一阶导数和二阶导数，也就是对应的速度和加速度，如下：

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

深蓝学院
shenlanxueyuan.com

感谢各位聆听！

Thanks for Listening