VIO 第二课作业

靳波
2019/6/18

基础作业

1 设置 IMU 仿真代码中的不同的参数，生成 Allen 方差标定曲线

- IMU 仿真代码为 VIO_data_simulation-ros_version

仿真程序运行结果：

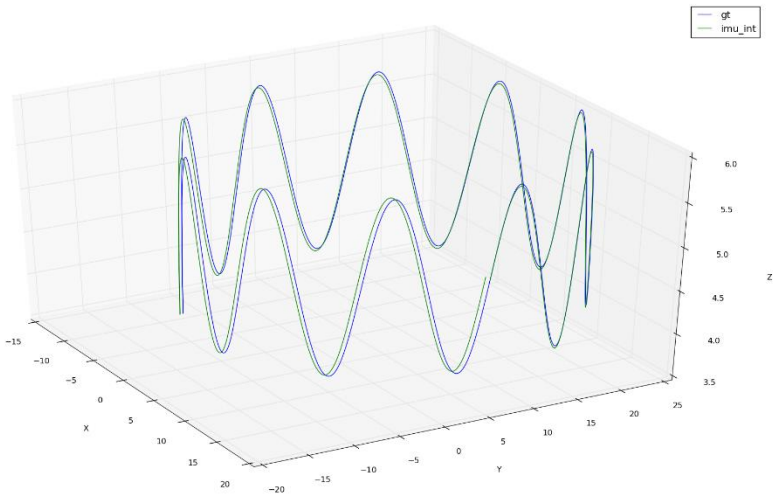


图 1 IMU 仿真曲线

为了得到不同的 Allen 方差曲线，设置两种不同的仿真参数，以得到两个不同的 IMU 数据，参数设置如下：

- 设置仿真参数 1：

加速度高斯白噪声：acc_nose_sigma=0.019(m/s^2)
陀螺仪的高斯白噪声:gyro_nose_sigma=0.015(rad/s)
加速度 bias 的随机游走噪声:acc_bias_sigma=0.0005
陀螺仪的 bias 随机游走噪声: gyro_bias_sigma=0.00005

- 设置仿真参数 2：

加速度高斯白噪声：acc_nose_sigma=0.02(m/s^2)
陀螺仪的高斯白噪声:gyro_nose_sigma=0.02(rad/s)
加速度 bias 的随机游走噪声:acc_bias_sigma=0.0001
陀螺仪的 bias 随机游走噪声: gyro_bias_sigma=0.00001

这里生成 allan 方差曲线的工具为 imu_utils

程序运行结果以及方差曲线图如图 2 至图 7 所示

- 仿真参数 1 下的运行结果:

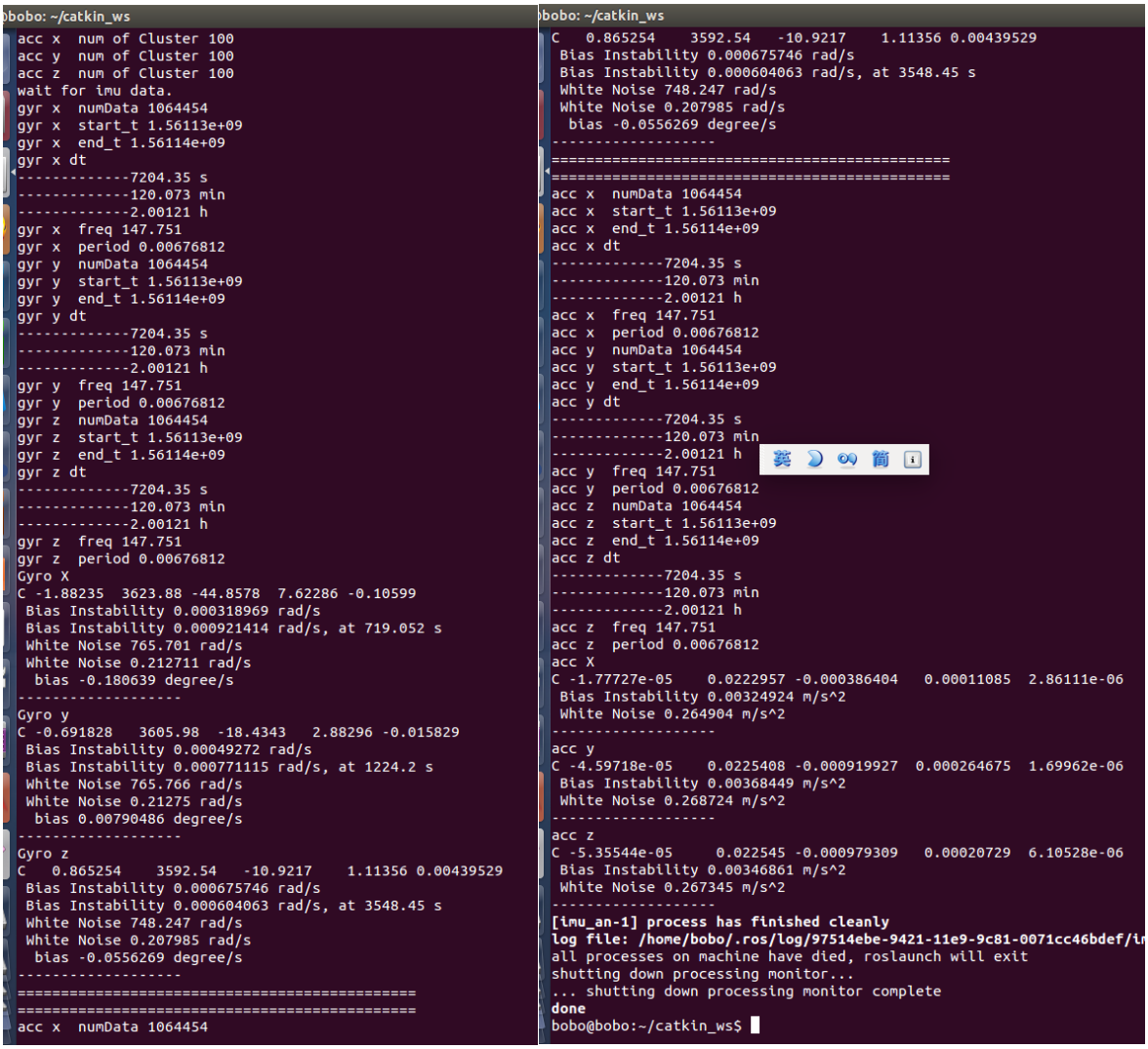


图 2 仿真参数 1 情况下的噪声估计结果

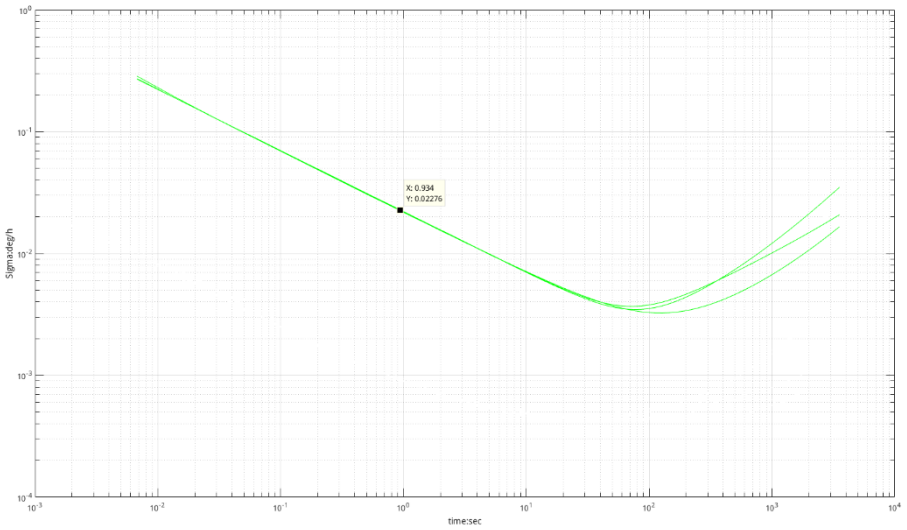


图 3 仿真参数 1 情况下加速度计 Allen 方差曲线

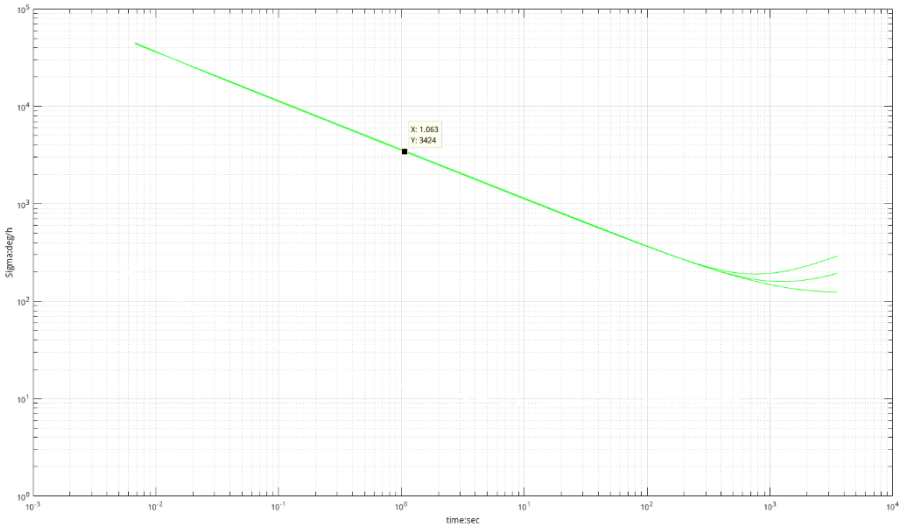


图 4 仿真参数 1 情况下陀螺仪 Allen 方差曲线

- 仿真参数 2 下的运行结果:

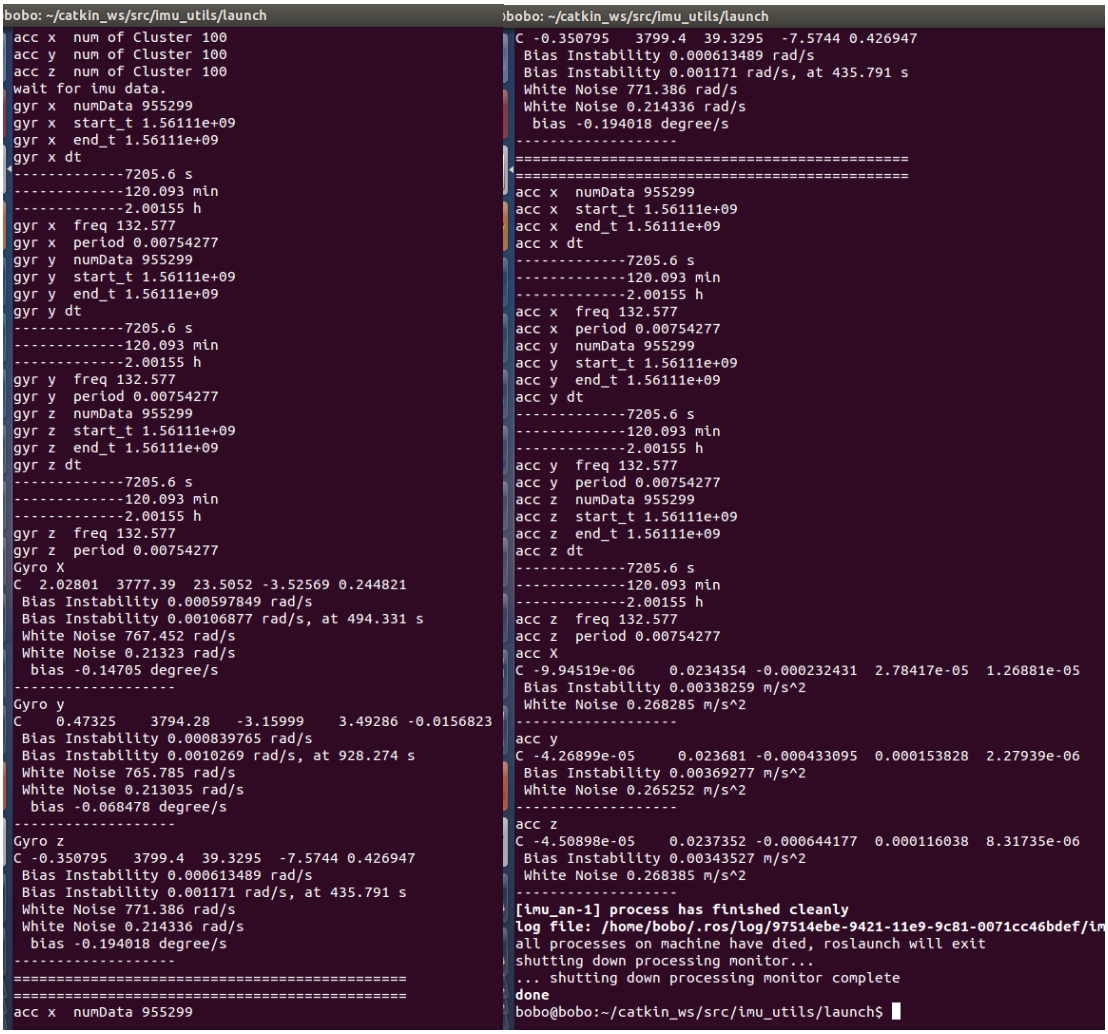


图 5 仿真参数 2 情况下的噪声估计结果

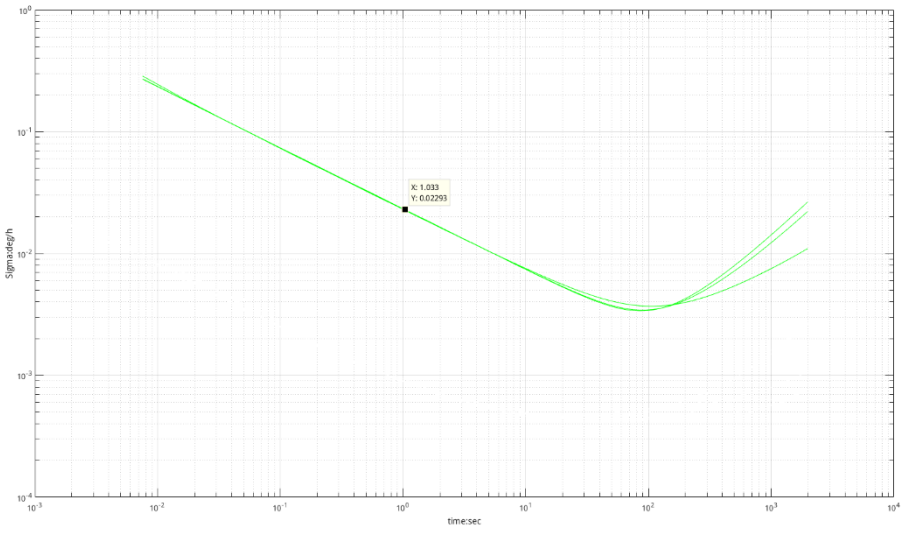


图 6 仿真参数 2 情况下加速度计 Allen 方差曲线

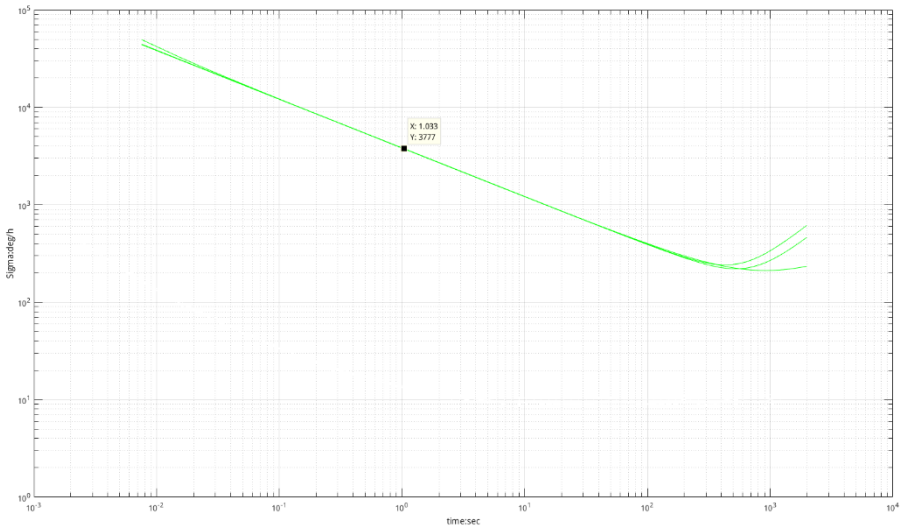


图 7 仿真参数 2 情况下陀螺仪 Allen 方差曲线

分析，在参数 1 下，标定的加速度为 0.02276 这与设定值 0.019 相差不大。标定的陀螺仪为 3424 deg/h 换算单位后为 0.01659 rad/s，这与设定值 0.015 接近。
在参数 2 下，标定的加速度为 0.02293 这与设定值 0.02 相差不大。标定的陀螺仪为 3777 deg/h 换算单位后为 0.0183 rad/s，这与设定值 0.02 接近。

2 将 IMU 仿真代码中的欧拉积分替换成中值积分

关键代码如下图所示：

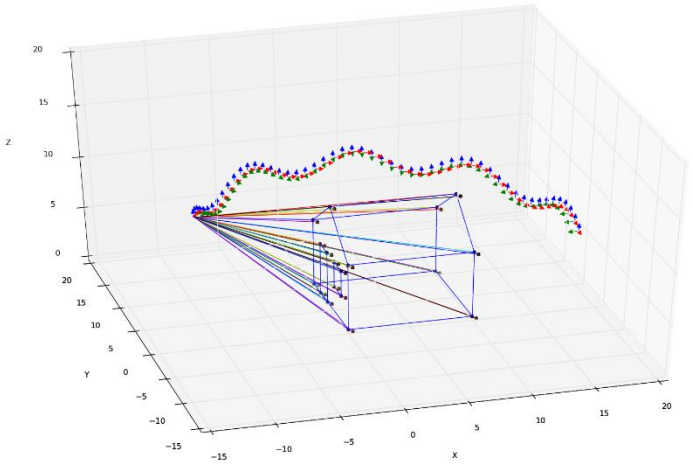
```
    Eigen::Quaterniond Qwb(init_Rwb_);           // quaterniond:  from imu measurements
    Eigen::Vector3d Vw = init_velocity_;         // velocity  :   from imu measurements
    Eigen::Vector3d gw(0,0,-9.81);    // ENU frame
    Eigen::Vector3d temp_a;
    Eigen::Vector3d theta;
    for (int i = 1; i < imudata.size()-1; ++i) {
        MotionData imupose = imudata[i];
        MotionData imupose1=imudata[i+1];
        //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();
        /// 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;
        //Vw = Vw + acc_w * dt;
        //Pwb = Pwb + Vw * dt + 0.5 * dt * dt * acc_w;

        /// 中值积分
        Eigen::Quaterniond dq;
        Eigen::Vector3d dtheta_half =  (imupose.imu_gyro+imupose1.imu_gyro) * dt /4.0;
        dq.w() = 1;
        dq.x() = dtheta_half.x();
        dq.y() = dtheta_half.y();
        dq.z() = dtheta_half.z();

        Eigen::Quaterniond Qwb1;
        Qwb1=Qwb*dq;
        Eigen::Vector3d acc_w = (Qwb * (imupose.imu_acc)+Qwb1*(imupose1.imu_acc))/2.0 + gw;

        Qwb=Qwb*dq;
        Vw = Vw + acc_w * dt;
        Pwb = Pwb + Vw * dt + 0.5 * dt * dt * acc_w;
```

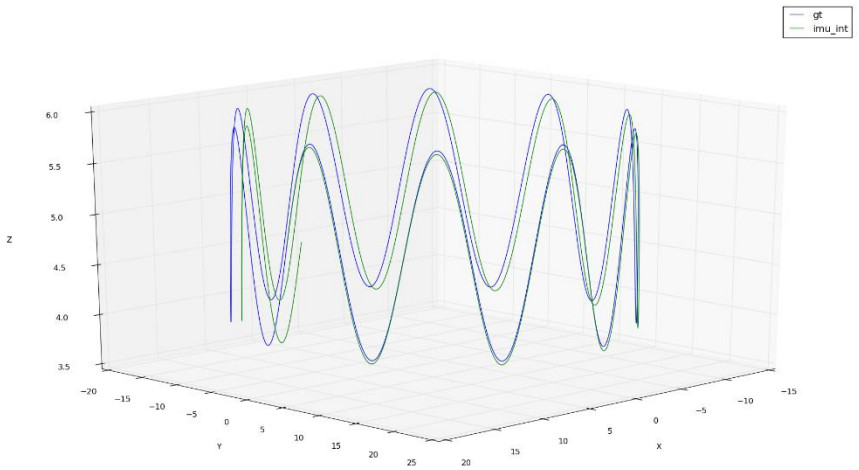
这里计算 dq 是使用 imu 陀螺仪在第 i 和第 i+1 时刻的中值
计算 acc_wy 使用的是加速度计在第 i 和第 i+1 时刻的中值
这里严格按照公式。

这里 for 循环重第 1 帧开始，我试过从第零帧开始，二者差别不大。

代码见附件，由于只改动了 imu.cpp 文件，因此附件中只放了 imu.cpp 文件，而没有放整个工程。

运行结果：





这里由于没有对仿真得到的轨迹和真实轨迹做出误差曲线，所以不能明显看出中值法与欧拉法之间的差异。但是从理论上来说中值法更为精确，但中值法的计算相比欧拉法要更复杂。

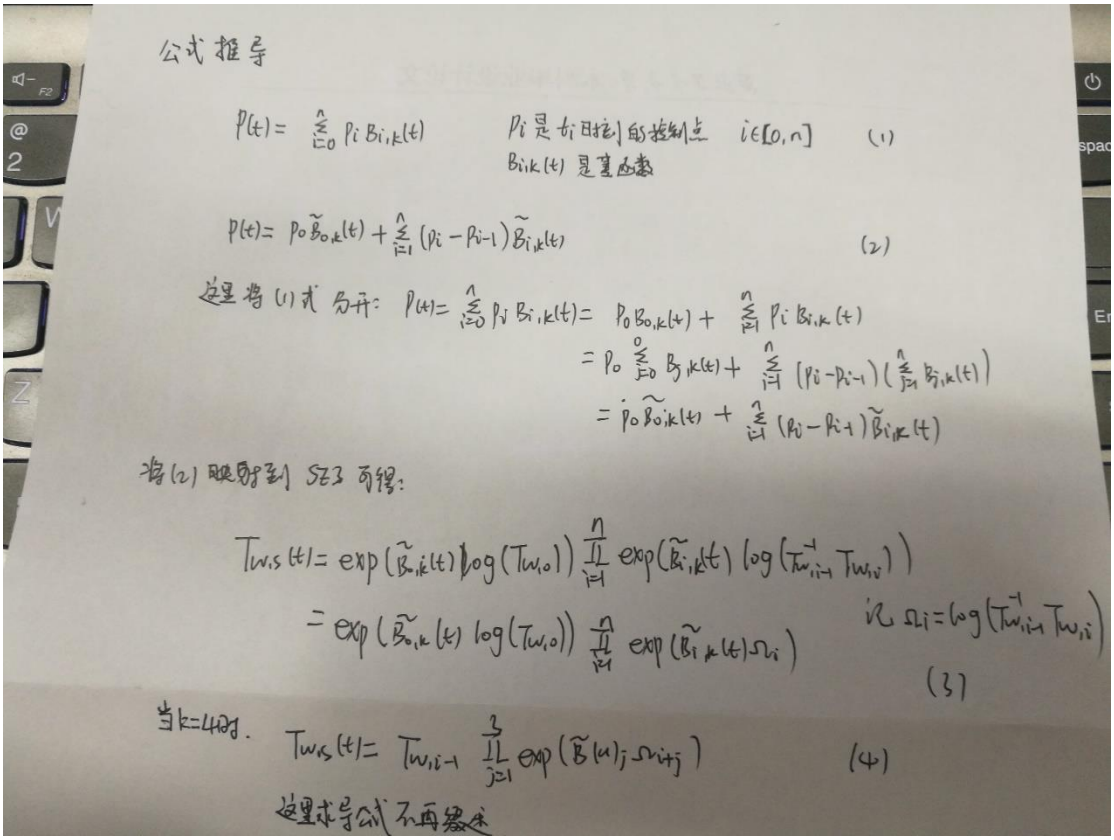
提升作业

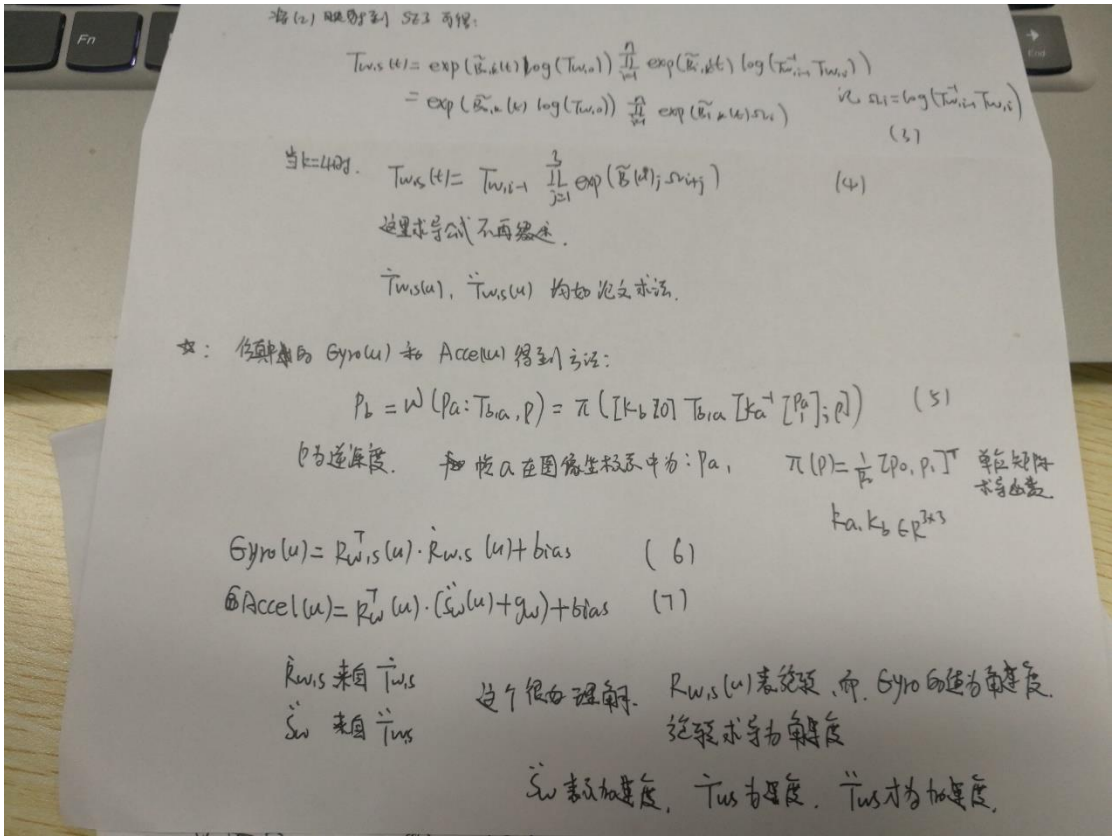
阅读从已有轨迹生成 imu 数据的论文，撰写总结推导

• 2013 年 BMVC, Steven Lovegrove ,Spline Fusion: A continuous-timerepresentation for

visual-inertial fusion withapplication to rolling shutter cameras.

这篇文章介绍了一种通用的用于视觉惯导 SLAM 标定的时间一致性框架。我们将演示如何使用与传感器的扭矩最小运动紧密匹配的样条参数化。与传统的离散时间解相比，连续时间公式对于解决高帧速率传感器和多个非同步器件的问题特别有用，我们通过精确确定相对姿态和多个非同步设备的内参，论证了该方法在多传感器视觉 SLAM 和校准中的适用性。我们还通过评估和统一处理视觉和视觉惯性 SLAM 系统中的全局和滚动快门相机，展示了该方法的优势。





$$T_{w,s}(t) = \exp(\tilde{\Omega}_{w,s}(t)\log(T_{w,0}))\prod_{i=1}^{n}\exp(\tilde{\Omega}_{w,s}(t)\log(T_{w,i-1}^{-1}T_{w,i}))$$

$$= \exp(\tilde{\Omega}_{w,n}(t)\log(T_{w,0}))\prod_{i=1}^{n}\exp(\tilde{\Omega}_{w,n}(t)s_i) \quad \text{记 } s_i = \log(T_{w,i-1}^{-1}T_{w,i}) \quad (3)$$

当 $k=4$ 时. $T_{w,s}(t) = T_{w,s-1}\prod_{j=1}^{3}\exp(\tilde{\Omega}(t)_j s_{i+j})$ (4)

这里求导公式不再赘述.

$\dot{T}_{w,s}(u)$, $\ddot{T}_{w,s}(u)$ 均如此交求法.

☆: 依靠离散的 Gyro(u) 和 Accel(u) 得到方法:

$$P_b = w(P_a; T_{ba}, p) = \pi([K_b\ 0]\,T_{ba}\,[K_a^{-1}[P_a^T]; p]) \quad (5)$$

p为逆深度. 相机a在图像坐标系中为: P_a, $\pi(P) = \frac{1}{z_0}[x_0, y_0]^T$ 单位矩阵投影函数.

$K_a, K_b \in R^{3\times3}$

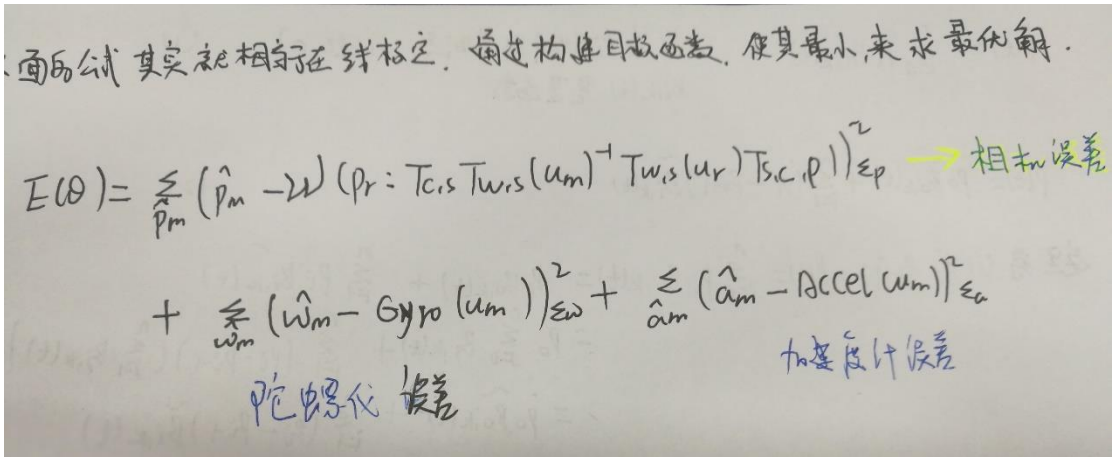
$$\text{Gyro}(u) = R_{w,s}^T(u)\cdot\dot{R}_{w,s}(u) + \text{bias} \quad (6)$$

$$\text{Accel}(u) = R_w^T(u)\cdot(\ddot{s}_w(u) + g_w) + \text{bias} \quad (7)$$

$R_{w,s}$ 和 $\dot{T}_{w,s}$

$\ddot{s}_w$ 和 $\ddot{T}_{w,s}$ 这个很好理解. $R_{w,s}(u)$ 表旋转, 而 Gyro 的值为角速度. 经过求导为角加速度

$\ddot{s}_w$ 表加速度, $\dot{T}_{w,s}$ 为速度, $\ddot{T}_{w,s}$ 为加速度.



面的公式其实就相对在线校正. 通过构造目标函数, 使其最小来求最优解.

$$E(\theta) = \sum_{p_m}(\hat{p}_m - w(p_r; T_{c,s}T_{w,s}(u_m)^{-1}T_{w,s}(u_r)T_{s,c}, p))^2_{\varepsilon_p} \rightarrow \text{相机误差}$$

$$+ \sum_{\hat{w}_m}(\hat{w}_m - \text{Gyro}(u_m))^2_{\varepsilon_w} + \sum_{\hat{a}_m}(\hat{a}_m - \text{Accel}(u_m))^2_{\varepsilon_a}$$

陀螺仪 误差　　加速度计误差

总结：本文基于摄像机轨迹的连续时间模型，对卷帘门序列的滑动窗视觉里程计进行了仿真验证。在这种模式下，我们能够仅从一台相机估算出准确的相机轨迹和场景结构，并证明忽略相机的卷帘会导致糟糕的结果。本文还展示了摄像机内部和 IMU 的视觉惯性校准，从真实数据到摄像机外部，给出了精确的对应关系。虽然这种校准是用卷帘照相机显示的，但是方法是灵活的，并且自然支持比以前的方法更广泛的传感器范围。通过在一个大窗口内对传感器数据进行联合融合，我们减少了卡尔曼滤波器内线性化可能引起的偏差。

助教辛苦了

祝好