

CHENXXX 第六章作业

完成情况

本次作业我依次完成了及格、良好和优秀的要求，并针对两个拓展任务做了论证。

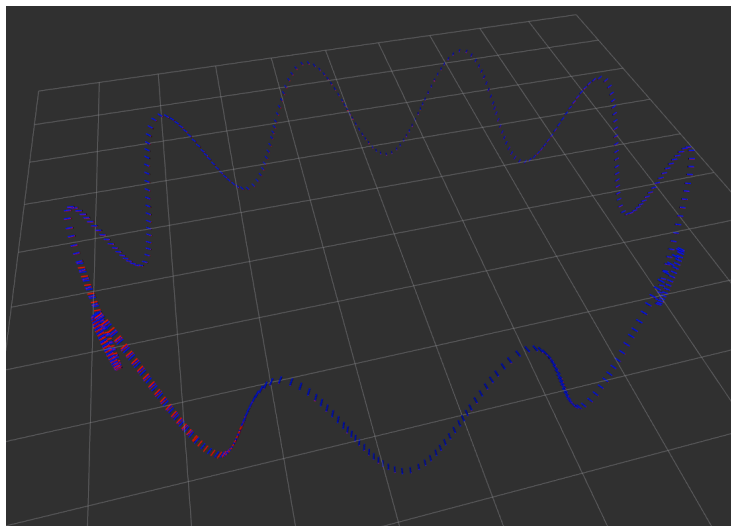
及格要求

思路

修改 `bool Activity::UpdatePose(void)` 函数的代码：

```
// 中值法
// get deltas:
Eigen::Vector3d angle_delta_mid;
GetAngularDelta(1, 0, angle_delta_mid);
// update orientation:
Eigen::Matrix3d R_prev_mid = pose_.block<3,3>(0,0);
Eigen::Matrix3d R_curr_mid = Eigen::Matrix3d::Identity();
UpdateOrientation(angle_delta_mid, R_curr_mid, R_prev_mid);
// get velocity delta:
Eigen::Vector3d velocity_delta_mid;
double delta_t_mid;
GetVelocityDelta(1, 0, R_curr_mid, R_prev_mid, delta_t_mid, velocity_delta_mid);
// update position:
UpdatePosition(delta_t_mid, velocity_delta_mid);
```

实验结果：



良好要求

思路

1.修改 `bool Activity::UpdatePose(void)` 函数的代码：

```
// 欧拉法
// get angluar delta
Eigen::Vector3d angle_delta_euler;
GetAngularDeltaEuler(1, 0, angle_delta_euler);
// update orientation
Eigen::Matrix3d R_prev_euler = pose_.block<3,3>(0,0);
Eigen::Matrix3d R_curr_euler = Eigen::Matrix3d::Identity();
UpdateOrientation(angle_delta_euler, R_curr_euler, R_prev_euler);
// get velocity delta
Eigen::Vector3d velocity_delta_euler;
double delta_t_euler;
GetVelocityDeltaEuler(1, 0, R_curr_euler, R_prev_euler, delta_t_euler, velocity_delta_euler);
// updae position
UpdatePosition(delta_t_euler, velocity_delta_euler);
```

2.增加 `GetAngularDeltaEuler()` 函数

```
const IMUData &imu_data_curr = imu_data_buff_.at(index_curr);
const IMUData &imu_data_prev = imu_data_buff_.at(index_prev);

double delta_t = imu_data_curr.time - imu_data_prev.time;

Eigen::Vector3d angular_vel_prev = GetUnbiasedAngularVel(imu_data_prev.angular_velocity);

angular_delta = delta_t * angular_vel_prev;
```

3.增加 `GetVelocityDeltaEuler()`函数

```
const IMUData &imu_data_curr = imu_data_buff_.at(index_curr);
const IMUData &imu_data_prev = imu_data_buff_.at(index_prev);

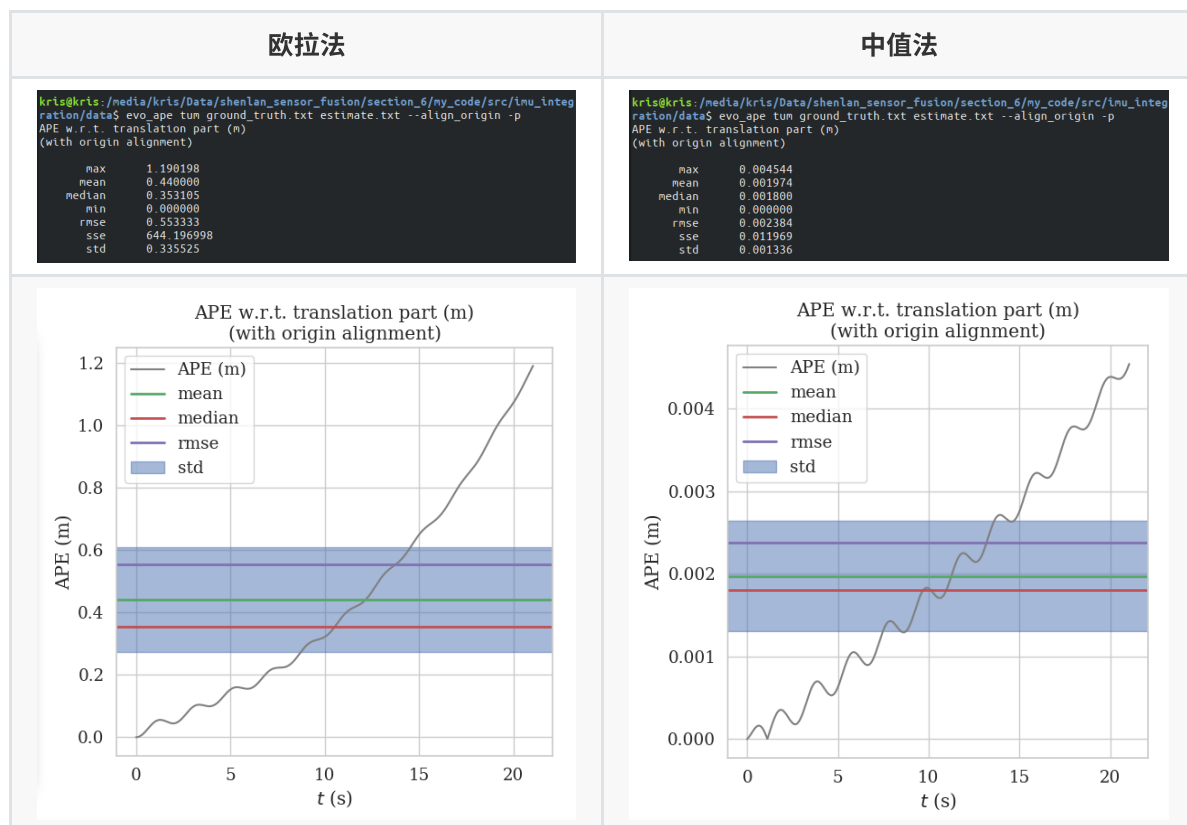
delta_t = imu_data_curr.time - imu_data_prev.time;

Eigen::Vector3d linear_acc_prev = GetUnbiasedLinearAcc(imu_data_prev.linear_acceleration, R_prev);

velocity_delta = delta_t * linear_acc_prev;
```

4.增加保存数据的相应代码

实验结果



从表格中可以看到，机体做机动性比较高的运动时，中值法精度要优于欧拉法。同时，我们也看到，两种方法随着时间推移，**APE值在不断升高，说明不断有误差累积**，进一步说明imu不适合长时间做积分

优秀要求

思路

1. 仿照 recorder_node_allan_variance_analysis.py 文件，重写了 my_generator.py 文件，增加了**无噪声**的陀螺仪，加速度计和ground truth输出
2. 编写 motion profile 文件，生成特定机动的仿真数据
3. 使用无噪声的传感器数据生成轨迹，并使用 evo 进行评价

motion profile文件的意思：

Command type	Comment
1	定义 相对于当前的 角度变化率(deg/s)和速度变化率(m/s^2), 第八列表示维持多少s
2	定义 相对于初始的 角度变化到多少deg, 速度变化到多少m/s; 如果到达指定角度或速度, 马上执行下一条命令; 如果到了规定时间还么到达制定角度或速度, 马上执行下一条命令
3	定义 相对于当前的 角度变化到多少deg, 速度变化到多少m/s; 执行时间规则同上
4	定于 相对于初始的 角度变化到多少deg, 相对于当前的 速度变化到多少m/s; 执行时间规则同上
5	定义 相对于当前的 角度变化到多少deg, 相对于初始的 速度变化到多少m/s; 执行时间规则同上

motion profile文件生成思路：

静止50s

```
ini lat (deg),ini lon (deg),ini alt (m),ini vx_body (m/s),ini vy_body (m/s),ini vz_body (m/s),ini yaw (deg),ini  
pitch (deg),ini roll (deg)  
31.224361,121.469170,0, 0,0,0, 0,0,0  
command type,yaw (deg),pitch (deg),roll (deg),vx_body (m/s),vy_body (m/s),vz_body (m/s),command  
duration (s),GPS visibility  
1,0,0,0,0,0,0,50,1
```

x轴匀速50s

```
ini lat (deg),ini lon (deg),ini alt (m),ini vx_body (m/s),ini vy_body (m/s),ini vz_body (m/s),ini yaw (deg),ini  
pitch (deg),ini roll (deg)  
31.224361,121.469170,0, 0.1,0,0, 0,0,0  
command type,yaw (deg),pitch (deg),roll (deg),vx_body (m/s),vy_body (m/s),vz_body (m/s),command  
duration (s),GPS visibility  
1,0,0,0,0,0,0,50,1
```

x轴0.1m/s²匀加速50s

```
ini lat (deg),ini lon (deg),ini alt (m),ini vx_body (m/s),ini vy_body (m/s),ini vz_body (m/s),ini yaw (deg),ini  
pitch (deg),ini roll (deg)  
31.224361,121.469170,0, 0,0,0, 0,0,0  
command type,yaw (deg),pitch (deg),roll (deg),vx_body (m/s),vy_body (m/s),vz_body (m/s),command  
duration (s),GPS visibility  
1,0,0,0,0.1,0,0,50,1
```

加减速

```
ini lat (deg),ini lon (deg),ini alt (m),ini vx_body (m/s),ini vy_body (m/s),ini vz_body (m/s),ini yaw (deg),ini  
pitch (deg),ini roll (deg)  
31.224361,121.469170,0, 0.1,0,0, 0,0,0  
command type,yaw (deg),pitch (deg),roll (deg),vx_body (m/s),vy_body (m/s),vz_body (m/s),command  
duration (s),GPS visibility  
1,0,0,0,0.1,0,0,10,1  
1,0,0,0,-0.2,0,0,10,1  
1,0,0,0,0.3,0,0,10,1  
1,0,0,0,-0.1,0,0,10,1  
1,0,0,0,0.05,0,0,10,1
```

连续转弯

command type,yaw (deg),pitch (deg),roll (deg),vx_body (m/s),vy_body (m/s),vz_body (m/s),command duration (s),GPS visibility

3,-90,,0,0,0,0,5,1

动作

欧拉法

中值法

静止 50s

匀速 50s

匀加速 50s

加减速

连续 转弯

分析

从上面实验结果可以看出，欧拉法与中值法在静止和匀速运动时精度相同；载体做加减速和连续转弯时，中值法精度要高于欧拉法。

由于静止和匀速运动时， k 时刻的速度相比于 $k-1$ 时刻是不变的，直接用 $k-1$ 时刻做积分与用 $k-1$ 和 k 时刻取平均后做积分的结果相同。当载体做加减速和转弯时，加速度和角速度是不断变化的，只用 $k-1$ 时刻的值做积分的累积误差肯定很大，所以中值法在在这种机动情况下的精度要优于欧拉法

拓展1

思路

RK4的公式推导请参考[hear](#)，代码参考 `open_vins` 中的RK4积分，具体实现请参考 `activity.cpp` 中 `RK4()` 函数

本程序复现的RK4是有问题的，精度整体不如中值法高，提交完作业再慢慢debug

实验结果

[illegible]

拓展2

思路

导航误差与器件误差关系验证思路：

1. 使用 `gnss-ins-sim` 生成有特定噪声，无运动的IMU数据
2. 向程序输入有特定噪声的IMU数据，输出估计值 \tilde{x}
3. 使用误差微分方程推导出导航误差 δx
4. 若 $\tilde{x} = x + \delta x$ ，则说明关系正确

实验结果

实验1

假设初始姿态为单位阵，当x轴加速度计有零偏 $b_{ax}=0.1m/s$ ，且没有其他器件误差和初始导航误差，仿真10s

根据x方向的速度误差，位置误差随时间变化：

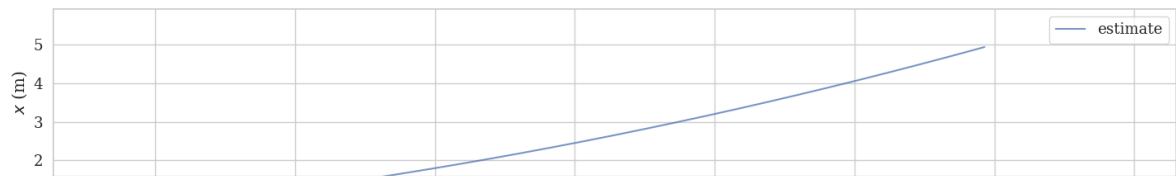
$$\begin{aligned}\delta v_x &= b_{ax} \Delta t \\ \delta p_x &= \frac{1}{2} b_{ax} \Delta t^2\end{aligned}$$

可得位置导航误差：

$$\delta p_x = 5m$$

程序输出结果：

$$\tilde{p}_x = 5m$$



载体实际是没有运动的，即 $p_x = 0m$ ：

$$\tilde{p}_x = p_x + \delta p_x$$

关系正确

实验2

假设初始姿态为单位阵，当x轴陀螺仪有零偏 $b_{gx} = 10deg/h$ ，且没有其他器件误差和初始导航误差，仿真360s

根据x方向的姿态误差随时间变换（角速度由于bias的影响，是时变的，从而导致速度误差，位置误差很难求，这里只考虑姿态）

$$\delta \theta = b_{gx} \Delta t$$

$\Delta t = 360s$ 求得：

$$\delta \theta_{roll} = 1^\circ$$

程序输出结果是：

$$\tilde{\theta}_{roll} = 1^\circ$$



载体实际是没有运动的，即 $\theta_{roll} = 0$:

$$\tilde{\theta}_{roll} = \theta_{roll} + \delta\theta_{roll}$$

关系正确

实验3

假设有初始姿态误差， \tilde{R} 是绕x轴转0.1度，真实载体没有动即 $R = I$, $p = 0$ ；没有其他器件噪声。可得失准角（弧度制）：

$$\delta\theta = \begin{bmatrix} 0.00087 \\ 0 \\ 0 \end{bmatrix}$$

根据x方向的速度误差，位置误差随时间变化：

$$\begin{aligned} \delta v &= [a - b_a]_{\times} \delta\theta \Delta t \\ \delta p &= \frac{1}{2} [a - b_a]_{\times} \delta\theta \Delta t^2 \end{aligned}$$

$\Delta t = 10s$ 求得：

$$\delta p = \begin{bmatrix} 0 \\ 0.426 \\ 0 \end{bmatrix}$$

程序估计：

$$\tilde{p} = \begin{bmatrix} 0 \\ 0.423 \\ 0.000018 \end{bmatrix}$$

满足 $\tilde{p} = p + \delta p$ ，关系正确。注意：推导姿态误差微分方程时使用了小角度近似，所以 $\delta\theta$ 的角度要很小时才能满足上述关系。同时，姿态误差微分方程随着失准角的增大会变得越来越不准