

## 第六章作业思路分享



主讲人 雍川



### ●1. 基于中值法的解算

b. 中值法: 
$$\phi = \frac{\omega_{k-1} + \omega_k}{2} (t_k - t_{k-1})$$

中值法的姿态更新用的是相邻两帧旋转角速度的均值，因此需要用到前后两相邻帧的信息。旋转角速度信息存储在消息队列中，所以前后两相邻帧的旋转角速度信息的索引号分别为0和1，在解算过程中，索引imu\_data\_buffer\_变量可以得到相邻的旋转角速度信息，用于更新位姿，然后删掉最早的 imu，如此循环更新位姿即可。

## 第六章作业思路分享

### ● 1. 基于中值法的解算

在完成作业时，大家需要用到以下函数：

```
bool Activity::GetAngularDelta(  
    const size_t index_curr, const size_t index_prev,  
    Eigen::Vector3d &angular_delta  
) {
```

通过相邻帧IMU数据  
计算角度变化量

```
void Activity::UpdateOrientation(  
    const Eigen::Vector3d &angular_delta,  
    Eigen::Matrix3d &R_curr, Eigen::Matrix3d &R_prev  
) {
```

利用角度变化量更新  
当前时刻的姿态

```
bool Activity::GetVelocityDelta(  
    const size_t index_curr, const size_t index_prev,  
    const Eigen::Matrix3d &R_curr, const Eigen::Matrix3d &R_prev,  
    double &delta_t, Eigen::Vector3d &velocity_delta  
) {
```

获取相邻帧的速度变化量

```
void Activity::UpdatePosition(const double &delta_t, const Eigen::Vector3d &velocity_delta)
```

更新速度和位置

### ● 1. 基于中值法的解算

```
// get deltas:
Eigen::Vector3d angular_delta;
GetAngularDelta(1, 0, angular_delta);
// update orientation:
Eigen::Matrix3d R_curr, R_prev;
UpdateOrientation(angular_delta, R_curr, R_prev);
// get velocity delta:
double delta_t;
Eigen::Vector3d velocity_delta;
GetVelocityDelta(1, 0, R_curr, R_prev, delta_t, velocity_delta);
// update position:
UpdatePosition(delta_t, velocity_delta);
// move forward:
imu_data_buff_.pop_front();
```

2. 基于欧拉法的解算，相对来说更简单，请大家自行实现。

- 2. 利用IMU仿真程序，生成不同运动状况(静止、匀速、加减速、快速转弯等)的仿真数据。大家需要仔细看一下仿真程序的readme文件。为了给基础薄弱的同学一个参考，我会分享我自己当时写的生成仿真数据的python文件，里面的一些配置参数和路径需要根据自己的实际情况修改。需要修改的参数如左图：

```
# motion def:
motion_file: /home/chongda517/catkin_practice/src
# IMU params:
imu: 1
# sample frequency of simulated GNSS/IMU data:
sample_frequency:
|   imu: 100.0
|   gps: 10.0
# topic name:
topic_name:
|   imu: /sim/sensor/imu
|   odom: /sim/sensor/odom
# output rosbag path:
output_path: /home/chongda517/data
# output name:
output_name: data5_acc.bag
```

## 第六章作业思路分享

b. motion\_file文件如下（参考上一期助教刘哲铭）：

比如绕飞8字，**command = 1**表示指令是角速度和加速度指令，每个指令的时间是36秒，**gps**可见，初始时刻只有**x**方向有速度。

A	B	C	D	E	F	G	H	I	J
ini lat (deg)	ini lon (deg)	ini alt (m)	ini vx_bo	ini vy_bo	ini vz_bo	ini yaw (deg)	ini pitch (deg)	ini roll (deg)	
32	120	0	5	0	0	0	0	0	
command	yaw (deg)	pitch (deg)	roll (deg)	vx_body	vy_body	vz_body	command	GPS visibility	
1	10	0	0	0	0	0	36	1	
1	-10	0	0	0	0	0	36	1	
1	10	0	0	0	0	0	36	1	
1	-10	0	0	0	0	0	36	1	



感谢各位聆听  
Thanks for Listening !

