

功能：

+ 主要功能是订阅激光里程计（低频里程计，来自 mapOptimization），对两帧激光里程计中间过程使用 IMU 信号做积分，得到高频的 IMU 里程计（与 IMU 同频）

功能要点：

+ IMU 信号积分使用 gtsam 的 IMU 预积分模块。里面共使用了两个队列，imuQueImu 和 imuQueOpt，以及两个预积分器 imuPreintegratorImu 和 imuPreintegratorOpt；

imuQueOpt 和 imuPreintegratorOpt 主要是根据历史信息计算 IMU 数据 bias 给真正的 IMU 里程计预积分器使用。imuQueImu 和 imuPreintegratorImu 是真正用来做 IMU 里程计的优化。

+ IMU 里程计主要是在 imageProjection 中被塞入 cloudInfo 数据结构，被当作每一帧雷达点云的初始估计位姿

+ 模块中有两个 handler，分别处理雷达里程计和 IMU 原始数据。雷达里程计的 handler 中，主要是将新到来的雷达里程计之前的 IMU 做积分，得出 bias；

IMU 的 handler 中主要是对当前里程计之后、下一时刻雷达里程计到来之前的时刻对 IMU 数据积分，并发布 IMU 里程计。

订阅：

1. IMU 原始数据
2. Lidar 里程计（来自 mapOptimization）

发布：

1. IMU 里程计（/lio_sam/imu/odometry）

流程：

1. 订阅雷达里程计

- * 1.1 如果系统没有初始化，则初始化系统，包括因子图、优化器、预积分器等
- * 1.2 每 100 帧雷达里程计之后重置优化器。清空因子图优化器，用优化出的结果作为先验
- * 1.3 将 imuQueOpt 队列中，所有早于当前雷达里程计的数据进行积分，获取最新的 IMU bias
- * 1.4 使用预积分器构造 ImuFactor，并加入因子图
- * 1.5 添加 Imu 的 BetweenFactor（偏差的相对差别）
- * 1.6 将雷达里程计平移对齐到 IMU（只做平移）
- * 1.7 构建雷达里程计因子，并加入因子图
- * 1.8 使用 IMU 预积分器的预测作为当前因子图的变量初始值
- * 1.9 将新的因子图和变量初始值加入优化器，并更新
- * 1.10 清空因子图和变量初始值缓存，为下一次加入因子准备
- * 1.11 从因子图中获取当前时刻优化后的各个变量
- * 1.12 重置预积分器
- * 1.13 检查优化结果，优化结果有问题时重置优化器
- * 1.14 将偏差优化器的结果传递到里程计优化器
- * 1.15 对里程计队列中剩余的数据进行积分

2. 订阅 IMU 原始数据

- * 2.1 加锁，对新来的 IMU 数据放入两个队列 (imuQueOpt,imuQueImu)
- * 2.2 对 IMU 数据直接使用 imuIntegratorImu_进行积分，并使用上一时刻的状态预测当前状态（积分结果）
- * 2.3 构建发布数据包，发布 IMU 里程计数据

备注：

1. 关于递增式因子图用法可以看 gtsam 仓库: [gtsam/examples/VisualSAM2Example.cpp](#)
2. 关于 IMU 预积分的用法可以看 gtsam 仓库:[gtsam/examples/ImuFactorExample.cpp](#)

Lidar 里程计？

// 整个框架认为雷达坐标系与底盘坐标系等同，因此这里直接发布的是单位矩阵