# IDEA

算法对比：WLS，EKF，FGO

赫尔默特方差+ AMHE本质上就是滑动窗口

目前的FGO问题：

计算成本显著增加；

定位精度仍有上限，残余误差依然较大；

无法完全解决异常值的“长尾”问题，算法虽然能抑制大部分显著的异常值，但无法完全将误差模型修正为标准的高斯分布，仍有一部分非高斯噪声残留；

Geman McClure 函数的核宽 (kernel width) 是“根据数据实验性选择的”。 在GNC的迭代过程中，控制参数 θ 的缩放因子被设为 1.4，但作者承认 **“如何明智地选择缩放因子仍然是一个悬而未决的问题；**

**依赖于批处理，影响实时性（滑动窗口解决即可，赫尔默特方差和AMHE）；**

**算法退化问题。**

MyThinking：

问题分解，将一个大的、非凸的复杂问题，分解为一系列在凸空间内的、更小、更易于处理的子问题。一个长时间的 FGO 问题可以被分解为多个时间段上的子图。每个子图可以被独立或并行优化，从而大幅降低单个优化问题的维度。可以先用一个计算代价小但精度较低的全局求解器（类似 Liang 论文中的 MAPF 引导）来确定每个子图之间的边界约束（即上一段的终点状态是下一段的起点状态），然后再调用 FGO-GNC 对每个子图进行精细优化。

Zhang et al. (2025) 在路径规划中，利用失败的搜索尝试来构建一个“估计信息集”（Estimated Informed Set），从而为后续的搜索提供一个更优的初始引导。FGO-GNC 是一个迭代优化过程，一个好的初始值至关重要。可以借鉴这个思路，在运行昂贵的 FGO-GNC 之前，先用一个快速但粗糙的定位算法（如 WLS）得到一个初始轨迹，或者直接利用上一时刻 FGO 的优化结果作为当前时刻的初始值。这种“智能初始化”能让优化器从一个更好的起点开始，从而减少迭代次数，并降低陷入局部最优的风险。

障碍函数的融入，将一个纯粹的**状态估计算法**，转变为一个能够进行**安全验证的系统。**

多普勒几何强度”因子 (Doppler Geometry Factor)：LEO多普勒测量对用户速度的解算能力极度依赖于其几何构型。一颗从头顶飞过的LEO卫星和一颗从地平线升起的卫星，它们提供的速度信息维度是完全不同的。我们可以设计一种新的因子，它不直接约束位置或速度，而是约束解的几何强度。在每个时刻，我们计算当前所有LEO卫星构成的“多普勒几何精度因子”（D-DOP）。然后，在FGO中加入一个 “几何约束因子”，该因子会惩罚那些导致D-DOP恶化的速度解。换句话说，优化器会被引导去寻找一个既满足测量残差，又在几何上最“可信”的速度状态。这为“松散”的速度状态节点之间提供了强大的隐式约束。（有点儿像我之前看的那个DGDOP）

信号机会的拓扑推断 (Topology Inference from Signal Opportunity)

核心思想: 在恶劣环境中，信号的出现和消失本身就蕴含了丰富的几何信息。

具体做法:

“首次捕获”因子 (First-Acquisition Factor): 当一颗刚刚从地平线升起的LEO或GNSS卫星首次被捕获时，我们可以推断出用户与这颗卫星之间大概率没有遮挡。这可以在FGO中形成一个\*\*“视线通畅”的几何约束\*\*，帮助限定用户的位置。

“信号中断”因子 (Signal-Loss Factor): 当一颗高仰角卫星的信号突然中断，很可能是进入了遮挡区（如隧道、高楼背后）。如果有粗略的地图模型，这个“信号消失”事件可以作为一个非常强的\*\*“位置约束因子”，将用户的位置限定在地图上的遮挡区域内。这是一种利用负信息（Negative Information）\*\*来增强FGO约束的全新思路。

**图的稀疏化 (Graph Sparsification)**: 当因子图变得过于庞大和稠密时，可以根据信息论的准则，**剪掉**那些信息量较低的边（因子）。例如，可以计算每个因子对整体信息矩阵的贡献度，周期性地移除贡献度最低的5%的因子，从而在不显著牺牲精度的前提下，有效控制计算量。**剪枝？**确实有说法，和滑动窗口优化可能还是要做区分一些？

噪声可以建设为随机游走模型，用多个因子来约束其随时间的变化，而不是简单地认为每一时刻的IMU测量都只与当前时刻的Bias有关。

**NOSTRA**论文虽然领域是贝叶斯优化（Bayesian Optimization），但其核心思想对于优化因子图（FGO）具有非常重要的启发意义。

FGO 本质上也是一个优化问题：寻找一组状态（轨迹）来最好地解释所有的测量数据。NOSTRA 论文要解决的核心问题与 FGO 遇到的困境惊人地相似：**如何在数据稀疏、稀少且充满噪声的情况下，高效地找到最优解？**

这篇论文为 FGO 的优化提供了两个极具价值的、可借鉴的核心思想：

**启发 1：信任域采样 (Trust Region Sampling) —— 从“全局蛮力搜索”到“重点区域精细挖掘”**

**NOSTRA 的做法：** 传统优化方法会在整个设计空间内盲目搜索。而 NOSTRA 则更加“聪明”，它不会浪费计算资源在那些“不太可能有好结果”的区域。它通过一个巧妙的机制，动态地在整个解空间中划定出一个\*\*“信任域” (Trust Region)**。这个信任域被认为是**最有可能包含帕累托最优解（Pareto frontier）的区域\*\*。然后，算法会集中火力在这个信任域内进行更密集的采样和优化，从而大大加快收敛速度。（其实就类似之前说的剪枝）

**对 FGO 的启发：** 当前的 FGO 求解器（如 Levenberg-Marquardt）在每次迭代时，本质上是在当前解附近的一个局部区域内寻找下一个更优的解。但是，这个“局部区域”的选择通常是基于简单的数学梯度（LAMBDA算法好像也是这玩意儿），缺乏更高层面的引导。

我们可以借鉴 NOSTRA 的思想，为 FGO 引入一个\*\*“状态空间信任域”\*\*：

**识别关键区域**: 在 FGO 优化前，可以先用一个计算成本极低的粗略算法（如 WLS）跑一遍，或者分析卫星几何构型（DOP值），识别出那些**不确定性可能非常大**的区域（例如，卫星信号严重遮挡的时段）。

**构建信任域**: 我们可以定义一个信任域，即**优先优化那些不确定性较小的、解比较可靠的区域**。

**分阶段优化策略**:

**阶段一（稳定阶段）**: 在信任域内（即信号良好、不确定性低的时段），FGO 可以采用更激进的优化策略，快速求解，并将这部分已经收敛的解作为整个图的“锚点”。

**阶段二（挑战阶段）**: 对于信任域之外的“坏数据”区域，FGO 可以采取更保守、更鲁棒的优化策略（例如，启用更强的鲁棒核函数，增加更多迭代次数），并利用第一阶段已经固定的“锚点”作为强大的先验信息来约束这部分的求解。

**解决了什么核心问题？** 这种方法将一个庞大而复杂的 FGO 问题，分解为了“先易后难”的两步走策略。它避免了让求解器从一开始就在最困难的数据上挣扎，而是先稳住基本盘，再集中优势兵力去攻克难点，从而**提高了整体求解的收敛性和鲁棒性**。

*FGO中的信任域：优先优化高质量数据区域，将其作为锚点来约束低质量区域的解*

**启发 2：先验知识注入 (Prior Knowledge Integration) —— 从“无知”的拟合到“有经验”的推理**

**NOSTRA 的做法：** 在数据稀少且有噪声时，单纯依赖数据进行模型拟合（例如，高斯过程的超参数估计）很容易出错。NOSTRA 强调，工程师通常对自己实验设备或仿真模型的噪声水平有一个**先验的认知**（Prior Knowledge）。NOSTRA 将这个先验知识（例如，噪声方差大概在什么范围）作为额外信息**注入**到高斯过程的训练中，从而在数据不足时，也能得到一个更合理、更稳定的代理模型。

**对 FGO 的启发：** FGO 的权重（信息矩阵）通常是基于一些固定的模型（如卫星仰角模型）来计算的，这本质上是一种“硬编码”的先验。我们可以让这个过程变得更灵活、更智能。

**可配置的先验因子**: 我们可以为 FGO 增加一系列\*\*“软性先验因子” (Soft Prior Factors)**。这些因子不是来自直接的测量，而是来自**领域知识\*\*：

**运动学先验**: “车辆的加速度通常不会超过5 m/s²”，“车辆的转弯半径通常大于5米”。这些都可以作为软约束因子加入图中。

**场景先验**: “在这条高速公路上，车辆大概率会以80-120 km/h的速度行驶”。这个速度范围可以作为一个先验因子，在 GNSS 信号质量差时提供强大的约束。

**LEO 卫星特性先验**: “Orbcomm 卫星的轨道误差通常在 XX 米量级，并且主要体现在 cross-track 方向上”。这个先验可以直接用来初始化 LEO 卫星轨道误差变量的协方差。

**贝叶斯化的参数估计**: FGO 中的很多参数，比如 IMU 的偏差、接收机钟差的漂移率，通常是被当作待优化的确定性变量。我们可以将它们**贝叶斯化**，即为它们设定一个先验分布。例如，我们可以根据传感器的手册，为 IMU 零偏的初始值设定一个高斯先验。在优化的过程中，算法将结合这个先验和数据，计算出一个后验分布，而不仅仅是一个点估计。

**解决了什么核心问题？** 通过注入先验知识，FGO 不再是一个单纯的“数学优化器”，而变成了一个\*\*“贝叶斯推理引擎”\*\*。它能够在测量数据本身不足以提供确定解时，利用先验知识做出最合理的推断，这对于处理 GNSS/LEO 信号频繁中断的挑战性场景至关重要，**极大地增强了算法在信息稀疏环境下的鲁棒性**。

**总结来说，NOSTRA 论文给 FGO 带来的最大启发是两种思维模式的转变：**

**从全局优化到重点突破**: 不要试图一次性解决所有问题，而是识别出问题的“薄弱环节”和“关键支撑点”，采用不同的策略来处理。

**从数据驱动到知识+数据双轮驱动**: 不要完全依赖有噪声的测量数据，而要将人类已知的、关于世界如何运作的先验知识，以数学化的方式融入到优化框架中。

# GraphGNSSLib-LEO

fgo\_leo

-> src -> GraphGNSSLib\_LEO\_V1.2/ -> docker(存镜像的地方)、img(一些基础图片)、support\_files(理论文档)、nlosexclution(存放.msg文件，在使用catkin\_make时根据这些文件自动生成C++头文件和Python脚本，并放到devel目录中，二次使用catkin时就是将这些文件打包成可执行程序；.msg文件定义了项目中所有节点之间通信时使用的自定义消息类型，主要还是用来规范消息标准，保证消息能正常传递；其余的python函数主要是坐标转换，以及北斗/GNSS组合定位、ROS监听/订阅节点、经纬度转KML文件等等)、rviz\_satellite(在轨迹下面叠上可视化的谷歌地图，包括对GPS节点的订阅处理等)、global\_fusion(核心代码所在)

-> result

-> devel -> 可执行文件

-> build

接下来主要讲解global\_fusion下的核心代码。

## Launch

储存各种模式的ROS启动节点。我们默认的roslaunch命令有时候rviz和程序是分开运行的，这是因为lauch里面把同时启动的代码给注释掉了。

## 2.ThirdParty

第三方库。主要包括LLH->ECEF，笛卡尔坐标系转换等文件。

## 3.Include

核心部分，大多数函数逻辑都在这儿写。

gnss\_tools.h：基础函数，提供坐标系转换，数据提取，数据预处理，误差建模，RTK/DD辅助函数。

pseudorange\_factor.h：基于 Google 的 Ceres Solver 库实现。定义了与伪距测量相关的因子，主要是标准伪距因子和双差伪距因子。

carrier\_phase\_factor.h：基于 Google 的 Ceres Solver 库实现。定义了与载波相位测量相关的因子，主要是标准载波相位因子和双差载波相位因子。

doppler\_factor.hpp：基于 Google 的 Ceres Solver 库实现。定义了与**多普勒频移 (Doppler)** 测量相关的因子。将不同时刻的状态节点连接起来形成时序约束。

psr\_doppler\_fusion.h：上层应用，完成伪距和多普勒因子的组合和优化的求解，调用 Ceres Solver 求解，并保存优化结果。

psr\_doppler\_car\_rtk.h：进阶应用，完成伪距、载波相位(均为双差)和多普勒因子的组合优化求解(RTK)。调用 Ceres Solver 求解得到浮点解后，调用LAMBDA函数搜索固定解。

psr\_doppler\_car\_rtk\_dyna：终极版本，额外使用了 Ceres Solver 的 **DynamicAutoDiffCostFunction** (动态自动微分代价函数)。

## 4.rviz

**ROS 可视化工具 RViz 的配置文件**。它们的功能是**保存一个预先设定好的可视化界面布局**，让开发者可以一键加载，方便地查看和调试 ROS 节点发布的数据。用户无需每次打开 RViz 都手动添加和配置各种显示项。

**rtk.rviz**

这个配置文件主要用于可视化和对比RTK(Real-Time Kinematic)定位算法的结果，完成开源卫星地图的显示，并订阅/gps/fix话题来确定地图的中心位置。

(1)PNT(WLS):显示一个Odometry消息，表示通过加权最小二乘法(WLS)计算出的定位结果。它订阅/rtk\_estimator/pntpos\_odometry话题，并用红色箭头表示。

(2)RTKLIB(integer):显示来自RTKLIB库的整周模糊度固定解(integer)。它订阅/rtk\_estimator/rtk\_integer\_odometry话题，并用绿色箭头表示。

(3)FGO\_RTK:显示本项目因子图优化(FGO)的RTK结果。它订阅/FGO话题，并用蓝色箭头表示。

(4)Path:显示一条蓝色的路径，用于可视化/rtk\_estimator/global\_path话题发布的全局轨迹。

其他显示项如GroundTruth(地面真值)和RTKLIB(float)(RTKLIB浮点解)被配置但默认未启用(Enabled:false)。

gnss\_positioning.rviz

这个配置文件用于更广泛地对比多种不同的GNSS定位算法，包括WLS、RTK和因子图优化(FGO)的结果。

核心功能:

多算法对比:设计用于同时可视化多达五种不同的定位轨迹和状态，并用不同颜色加以区分，是进行算法评估和性能比较的核心工具。

在线地图:同样加载了在线地图(https://cartodb-basemaps-a.global.ssl.fastly.net/light\_all/{z}/{x}/{y}.png)作为背景。

统一坐标系:所有显示项都在map这个固定坐标系下进行显示。

主要显示项(Displays):

GNSS-LEO-WLSENURTKLIB(Green):显示融合了LEO(低轨卫星)数据的WLS定位结果，用绿色箭头表示。订阅/psr\_spp\_gnssleo\_node/WLS\_spp\_GNSS\_LEO话题。

GNSS-only-WLSENUGoGPS(Yellow):显示仅使用传统GNSS的WLS定位结果，用黄色箭头表示。订阅/gnss\_preprocessor\_node/WLSENUGoGPS话题。

GNSS-only-RTKENURTKLIB(Red):显示仅使用传统GNSS的RTK结果，用红色箭头表示。订阅/gnss\_preprocessor\_node/ENUSolutionRTK话题。

GNSS-only-WLSENURTKLIB(Blue):另一个仅使用GNSS的WLS结果，用蓝色箭头表示。订阅/gnss\_preprocessor\_node/WLSENURTKLIB话题。

GNSS-only-FGOPath:显示仅使用GNSS的因子图优化输出的轨迹，为紫色路径。订阅/FGOGlobalPath话题。

GNSS-LEO-FGOPath(Cyan):显示融合了LEO数据的因子图优化输出的轨迹，为青色路径。订阅/GNSS\_LEO\_FGOGlobalPath话题。

**psr\_spp\_LEO.rviz**

这个配置文件专注于可视化融合了LEO(低轨卫星)的伪距单点定位(SPP)结果，并与因子图优化的路径进行对比。

核心功能:

LEO融合可视化:主要用于展示和调试融合LEO卫星数据的WLS(加权最小二乘法)定位算法的效果。

谷歌卫星地图:加载了谷歌卫星地图(http://mt3.google.com/vt/lyrs=s&x={x}&y={y}&z={z})作为高清背景。

主要显示项(Displays):

WLS\_spp\_LEO(Purple):显示融合LEO数据的WLS单点定位结果，用绿色箭头表示(配置文件中名称为紫色，但颜色设置为绿色)。它订阅/psr\_spp\_gnssleo\_node/WLS\_spp\_GNSS\_LEO话题。

FGO\_path:显示因子图优化的结果路径，为红色路径。它订阅/FGOGlobalPath话题。

GroundTruth:配置了地面真值显示项，用于对比，但默认未启用。

综上所述，这三个 .rviz 文件是该项目进行算法开发、调试和成果展示的重要工具。它们通过将不同来源和算法的定位结果在同一个 3D 空间中（通常带有卫星地图背景）进行可视化，使得开发者能够非常直观地：

**1.对比精度**: 将不同算法的轨迹与地面真值或其他高精度结果进行比较。

**2.调试问题**: 观察轨迹的漂移、跳变等异常现象，从而定位算法问题。

**3.展示效果**: 清晰地展示不同算法（如 WLS vs RTK vs FGO，或 GNSS-only vs GNSS+LEO）之间的性能差异。

## 5.dataset

主要存着需要用到的数据集。

## 6.RTKLIB

主要是从观测值中读取伪距，载波相位观测值等数据，以及LAMBDA等函数操作。

## 7.src

核心代码库。

## 8.launch

ROS启动文件。