GREAT
GNSS+ REsearch, Application and Teaching

GREAT-MSF多传感器融合软件培训

蔡灿烽 申志恒
武汉大学测绘学院

2025 年 3 月

主要内容

GREAT
GNSS+ REsearch，Application and Teaching

一、GREAT软件介绍

二、GREAT-MSF软件使用

三、GREAT-MSF代码讲解

四、答疑与交流

主要内容

GREAT
GNSS+ REsearch，Application and Teaching

一、GREAT软件介绍

二、GREAT-MSF软件使用

三、GREAT-MSF代码讲解

四、答疑与交流

GREAT软件平台介绍

GREAT
GNSS+ REsearch, Application and Teaching

GREAT软件

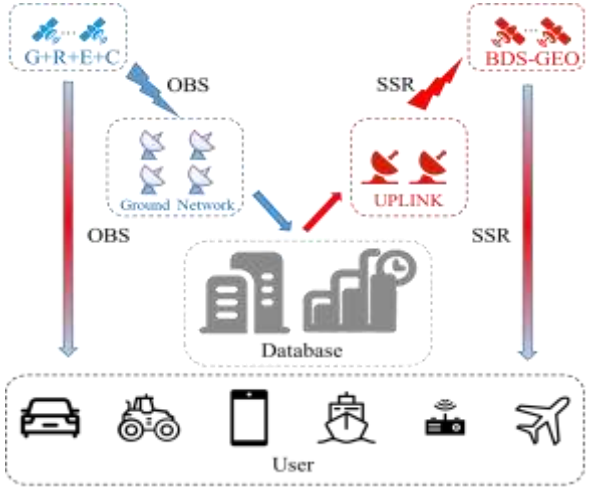
GREAT (GNSS+ REsearch, Application and Teaching) 软件由武汉大学测绘学院设计开发，是一个用于空间大地测量数据处理、精密定位和定轨以及多源融合导航的综合性软件平台

核心功能一:

导航卫星精密定轨、实时钟差、低轨导航增强以及空间大地测量数据融合处理

核心功能二:

基于 GNSS 的高精度定位，包括 PPP、PPP-AR、PPP-RTK和RTK



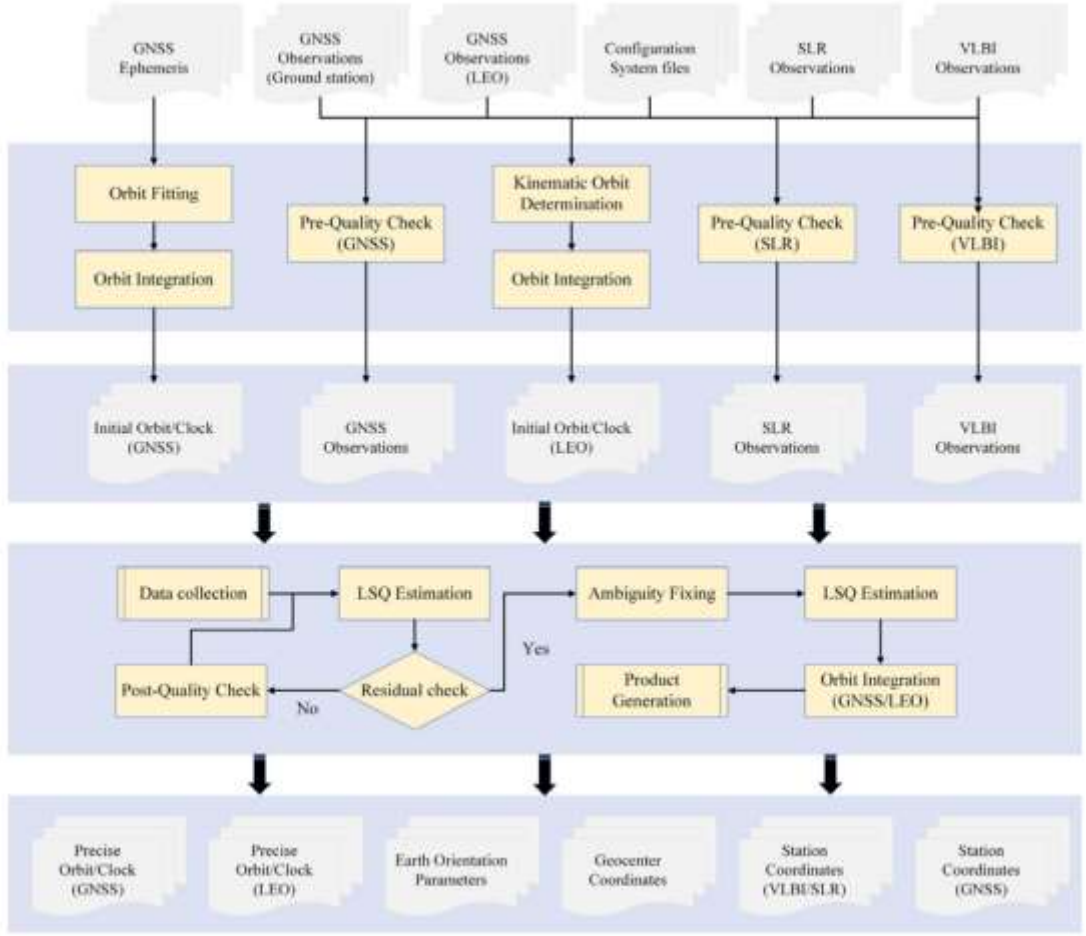
核心功能三:

基于因子图和滤波的GNSS、视觉、激光、惯性以及高精地图等多源信息紧融合算法

GREAT软件平台核心功能模块

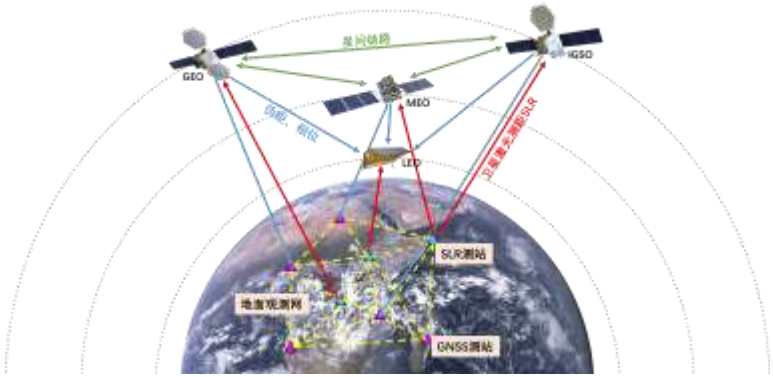
GREAT
GNSS+ REsearch, Application and Teaching

- 核心功能一：导航卫星精密定轨、实时钟差、低轨导航增强以及在观测值层面的多种空间大地测量技术（GNSS/SLR/VLBI）联合解算



空间大地测量数据处理示意图

多种空间大地测量数据融合



软件可实现高中低轨卫星联合处理，观测值层面融合GNSS、VLBI、SLR数据。

生成产品：

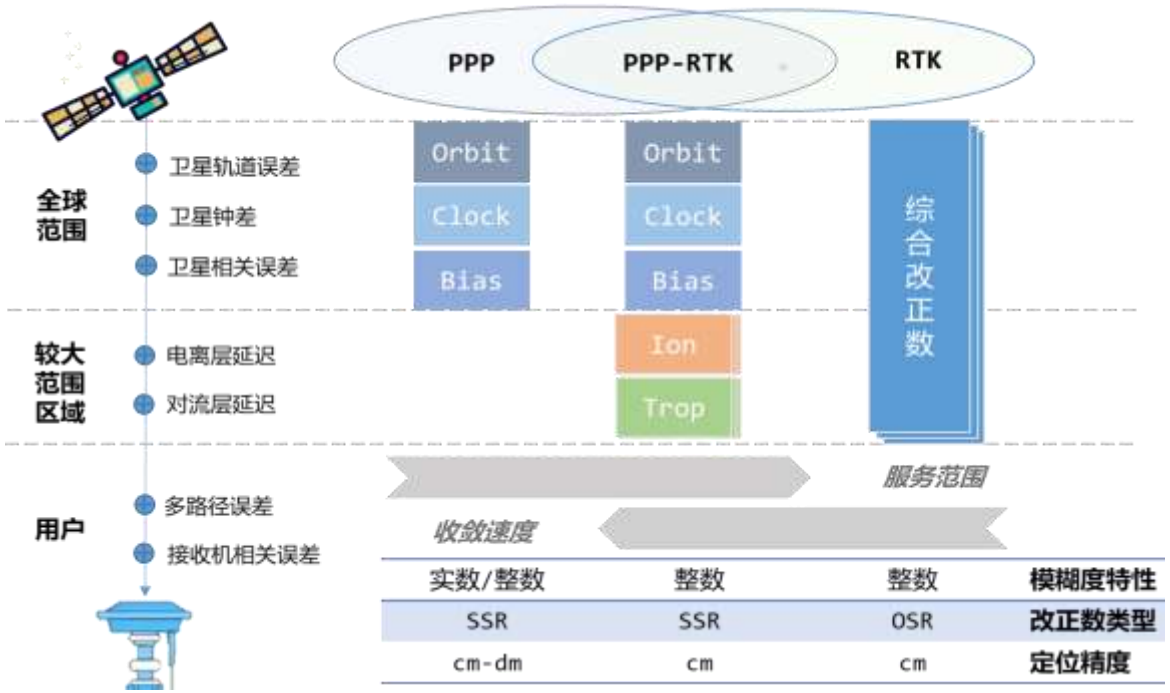
- GNSS轨道、钟差
- 低轨卫星轨道、钟差
- 地球自转参数
- 地心运动
- GNSS测站坐标
- SLR测站坐标
- VLBI测站坐标

GREAT软件平台核心功能模块

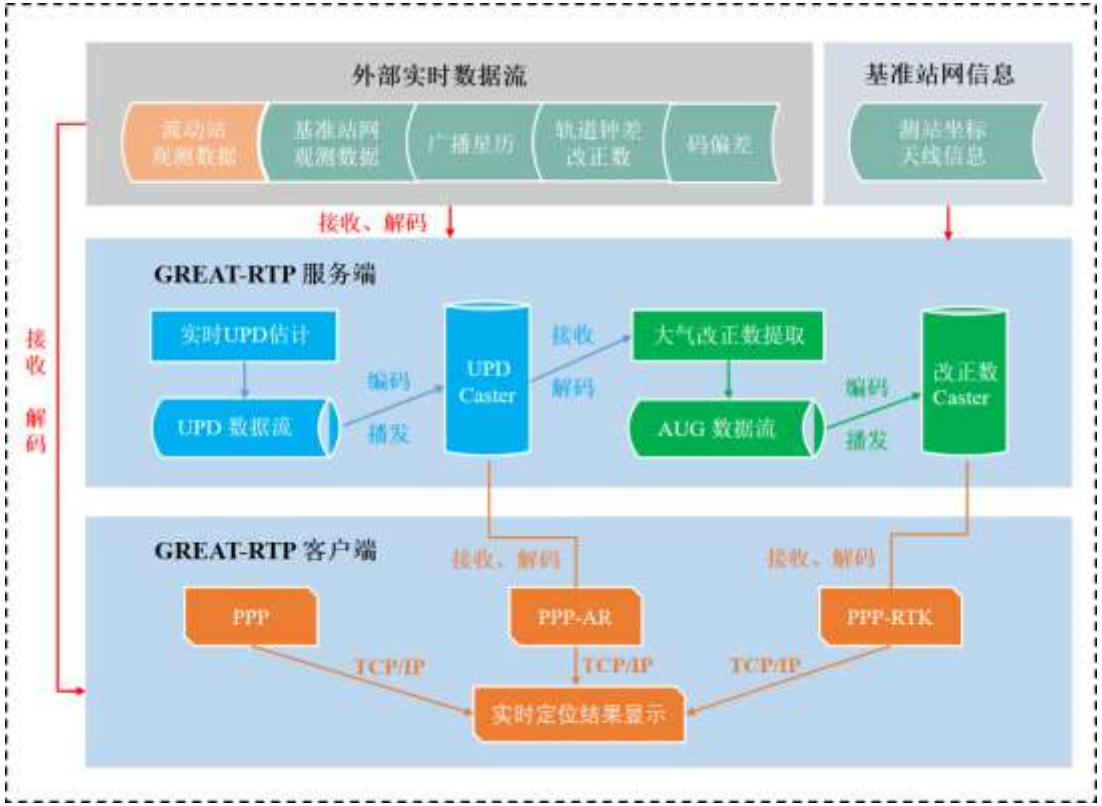
GREAT
GNSS+ REsearch, Application and Teaching

核心功能二：以PPP-RTK为代表的实时精密定位

✓生成并提供多频多系统实时精密轨道、实时钟差、实时UPD以及实时精密大气产品，支持星地一体化增强的快速精密定位（PPP/PPP-AR/PPP-RTK）



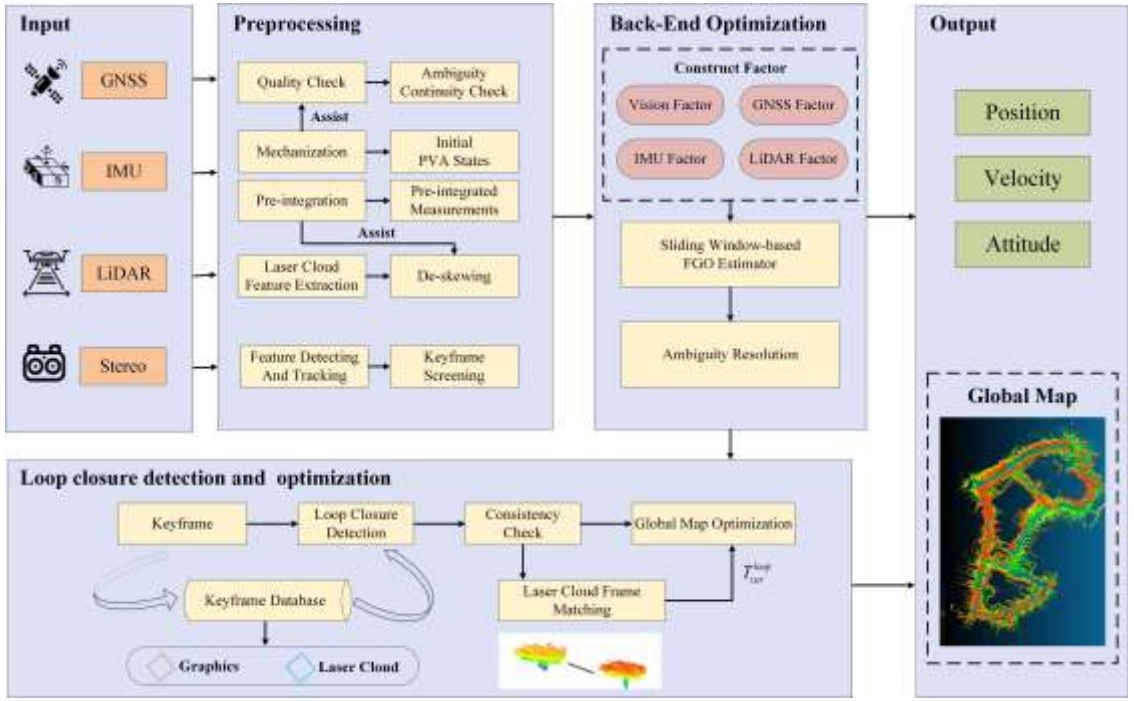
实数/整数	整数	整数	模糊度特性
SSR	SSR	OSR	改正数类型
cm-dm	cm	cm	定位精度

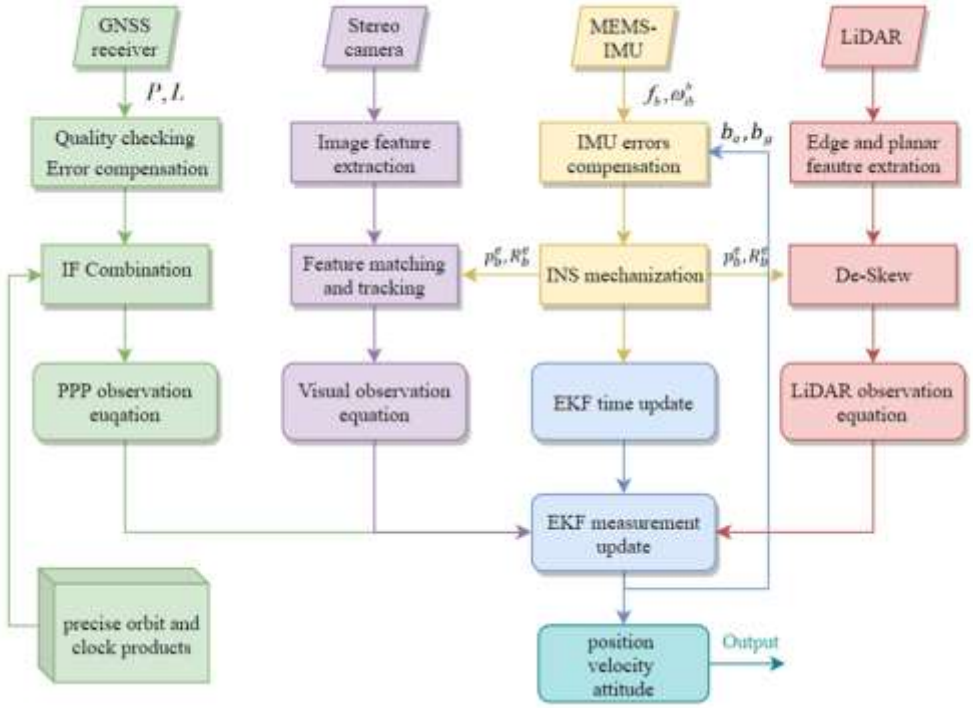


GREAT软件平台核心功能模块

GREAT
GNSS+ REsearch, Application and Teaching

- 核心功能三：基于因子图和滤波的GNSS、视觉、激光、惯性以及高精地图等多源信息融合算法





✓构建了基于原始观测信息的多传感器紧耦合高精度导航定位算法框架

✓设计了滑动窗口因子图估计器用于GNSS、惯性、视觉、激光雷达多源信息联合非线性优化

✓利用惯导机械编排传递系统状态，**辅助视觉特征跟踪与激光点云去畸变**

✓将全局地图统一到ECEF框架下，引入**关键帧地图**，用于实现高效、精确地激光点云配准

GREAT软件平台介绍

GREAT
GNSS+ REsearch, Application and Teaching

GREAT软件平台特点

- ✓ 采用主流编程语言C++进行核心库开发
- ✓ 使用标准化代码工具Git、CMake进行代码管理和分发
- ✓ 兼容GCC、Clang、MSVC等不同C++编译器
- ✓ 提供Window、Linux和MacOS等多种平台的应用程序

编程语言

核心计算库:C++
自动化脚本:Python&Shell

规范化代码管理

研发代码协同:本地Gitlab仓库
对外版本发布:GitHub仓库
https://github.com/GREAT-WHU

跨平台支持

基于CMAKE工具实现跨平台编译

GREAT智能导航实验室

GREAT
GNSS+ REsearch, Application and Teaching

团队开源项目GitHub主页：https://github.com/GREAT-WHU

精密定位导航软件	**GREAT-PVT**
多源融合导航软件	**GREAT-MSF**
频率间钟偏差软件	**GREAT-IFCB**
视觉众包建图软件	**RoadLib**
事件相机IMU标定软件	**RTEI-Calib**

相位小数偏差软件	**GREAT-UPD**
低轨导航增强软件	**GREAT-LAG**
多源传感器标定软件	**iKalibr**
视觉稠密建图软件	**DBA-Fusion**
多源传感器数据集	**GREAT-Dataset**

Things We Hope To Achieve:

更加稳定	**More Stable**
更加智能	**More Intelligent**
更加精密	**More Precise**
更加出彩	**More Outstanding**

Welcome To Follow!

主要内容

GREAT
GNSS+ REsearch，Application and Teaching

一、GREAT软件介绍

二、GREAT-MSF软件使用

三、GREAT-MSF代码讲解

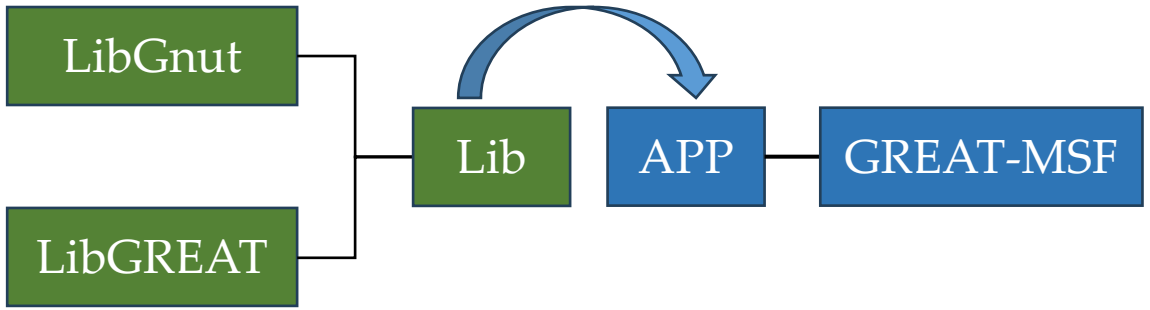
四、答疑与交流

GREAT-MSF 软件概述

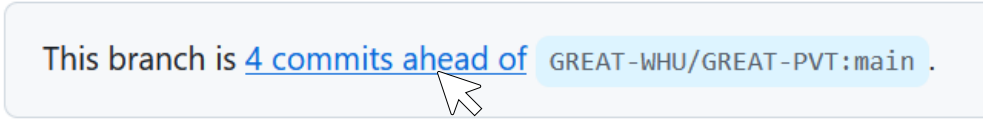
GREAT
GNSS+ REsearch, Application and Teaching

软件概述

- ✓ GREAT-MSF在GREAT-PVT的基础上开发，增加了多传感器融合导航解算功能
- ✓ GREAT-MSF由2个可移植程序库组成，分别是LibGREAT和LibGnut
- ✓ LibGREAT库除了原GREAT-PVT中的GNSS定位解决方案外，进一步集成了以下功能的实现：多源融合滤波估计中涉及的数据解码与存储、惯导解算以及融合算法。
- ✓ LibGnut库主要用于GNSS数据的解码和存储以及基本参数配置模块



GREAT-MSF在GREAT-PVT上做的增改详情可以在github仓库分支查看



GREAT-MSF 软件概述

GREAT
GNSS+ REsearch, Application and Teaching

软件功能

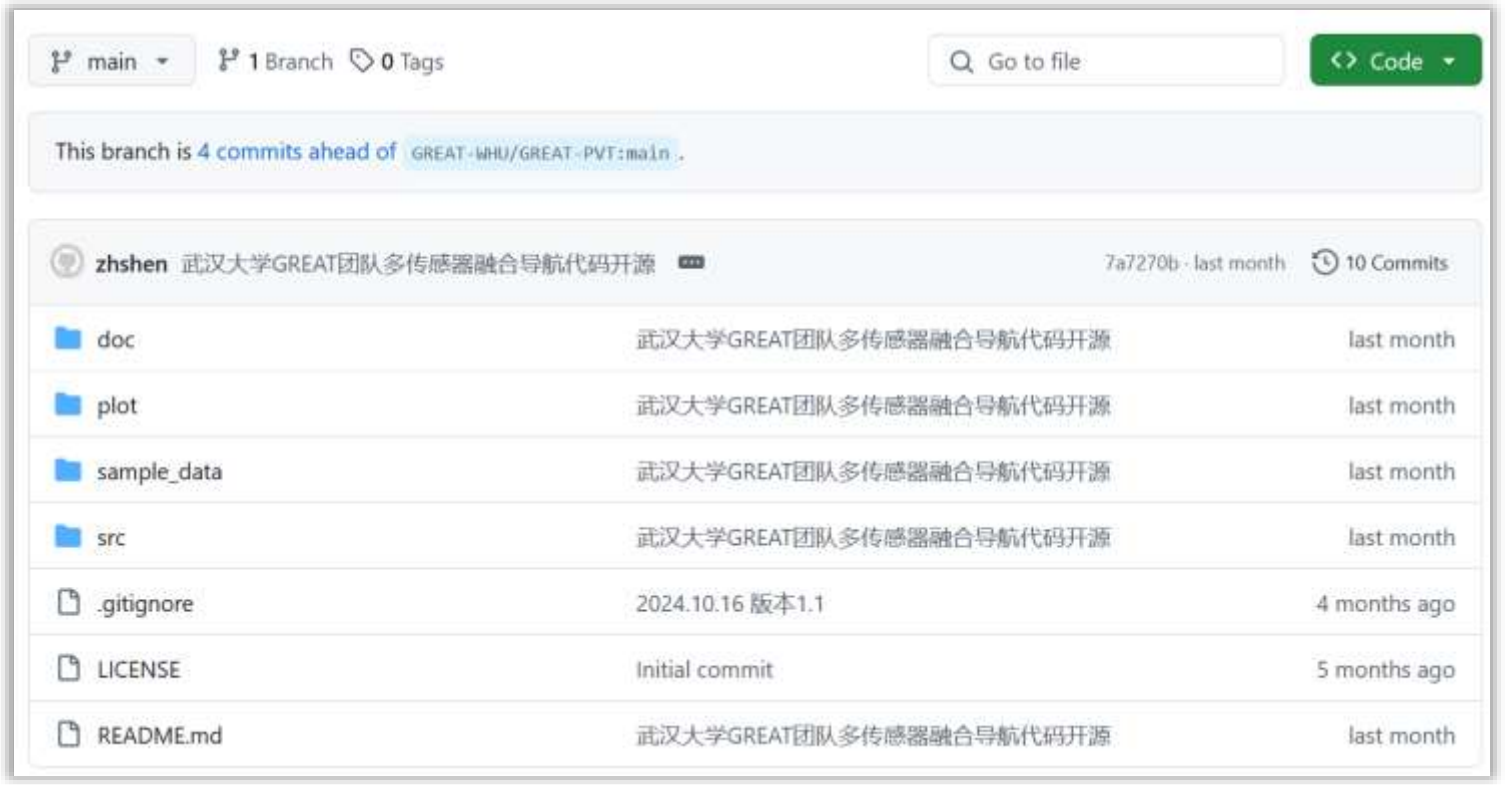
- ✓ 惯性导航机械编排与误差补偿校正
- ✓ **PPP/INS松耦合和紧耦合**，包括无电离层组合、非差非组合等PPP定位模型
- ✓ **RTK/INS松耦合和紧耦合**，支持载波相位**模糊度固定**
- ✓ 支持组合系统的动态快速初始化，包括位移矢量和速度矢量辅助对准
- ✓ 支持自定义的IMU数据格式和噪声模型
- ✓ 支持轨迹动态显示和谷歌地球查看
- ✓ 支持GPS、GLONASS、Galileo、BDS-2/3卫星导航系统

编译与安装

GREAT
GNSS+ REsearch, Application and Teaching

软件包目录结构

目录/文件	说明
./src	源代码
./sample_data	算例数据
./plot	绘图工具
./doc	文档文件
.gitignore	忽略的文件目录
LICENSE	许可证
README.md	软件介绍



代码发布地址为：https://github.com/GREAT-WHU/GREAT-MSF

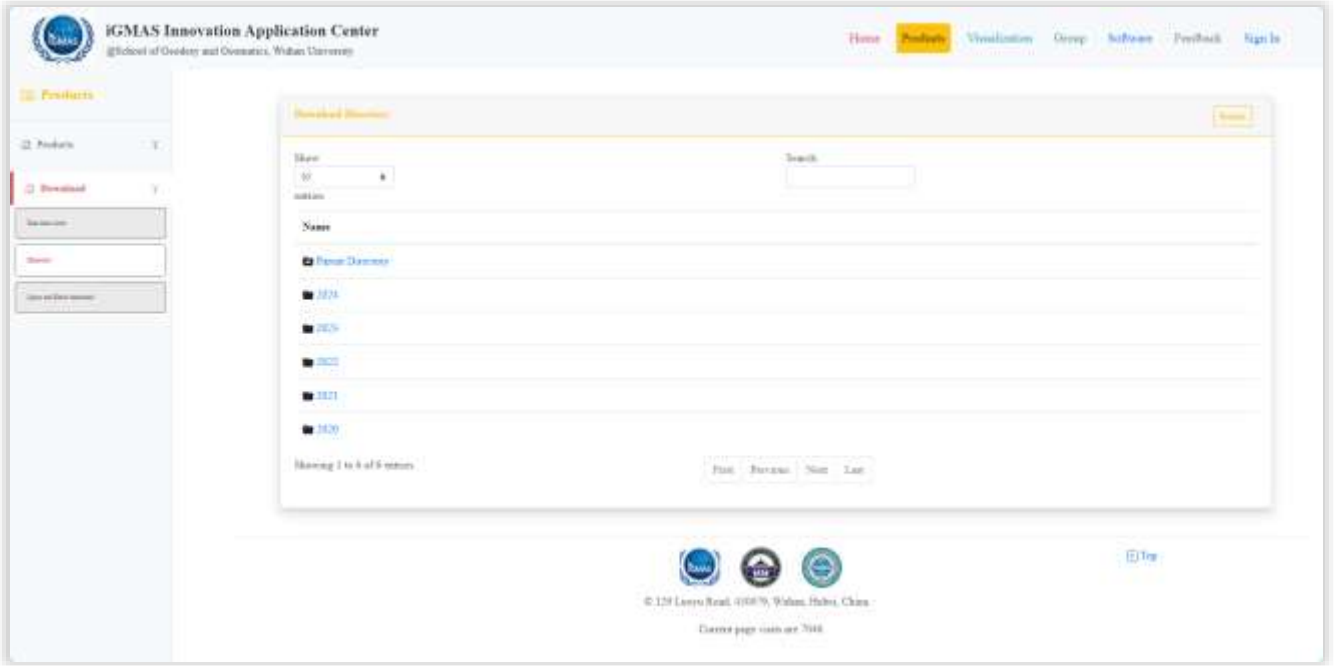
编译与安装

□数据准备

✓ 打开sample_data文件夹中任一算例

算例目录结构

目录	说明
./GNSS	GNSS数据文件夹
./IMU	IMU数据文件夹
./model	系统模型文件夹
./groudtruth	参考真值文件夹
./xml	xml配置文件夹
./result	结果输出文件夹



iGMAS创新应用中心

IFCB和UPD产品下载地址：
http://igmas.users.sgg.whu.edu.cn/products/download/directory/products/upd

编译与安装

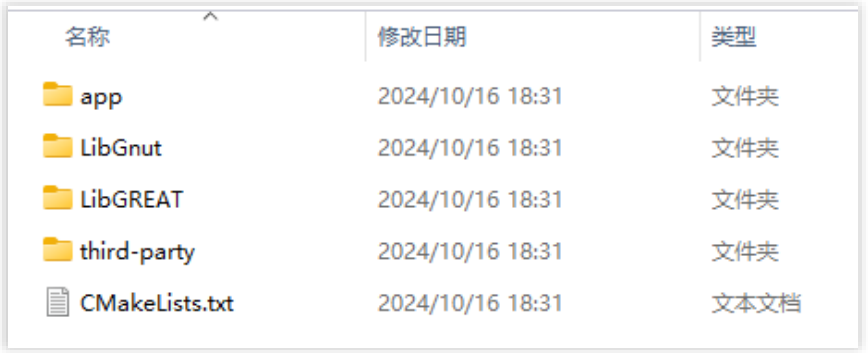
GREAT
GNSS+ REsearch, Application and Teaching

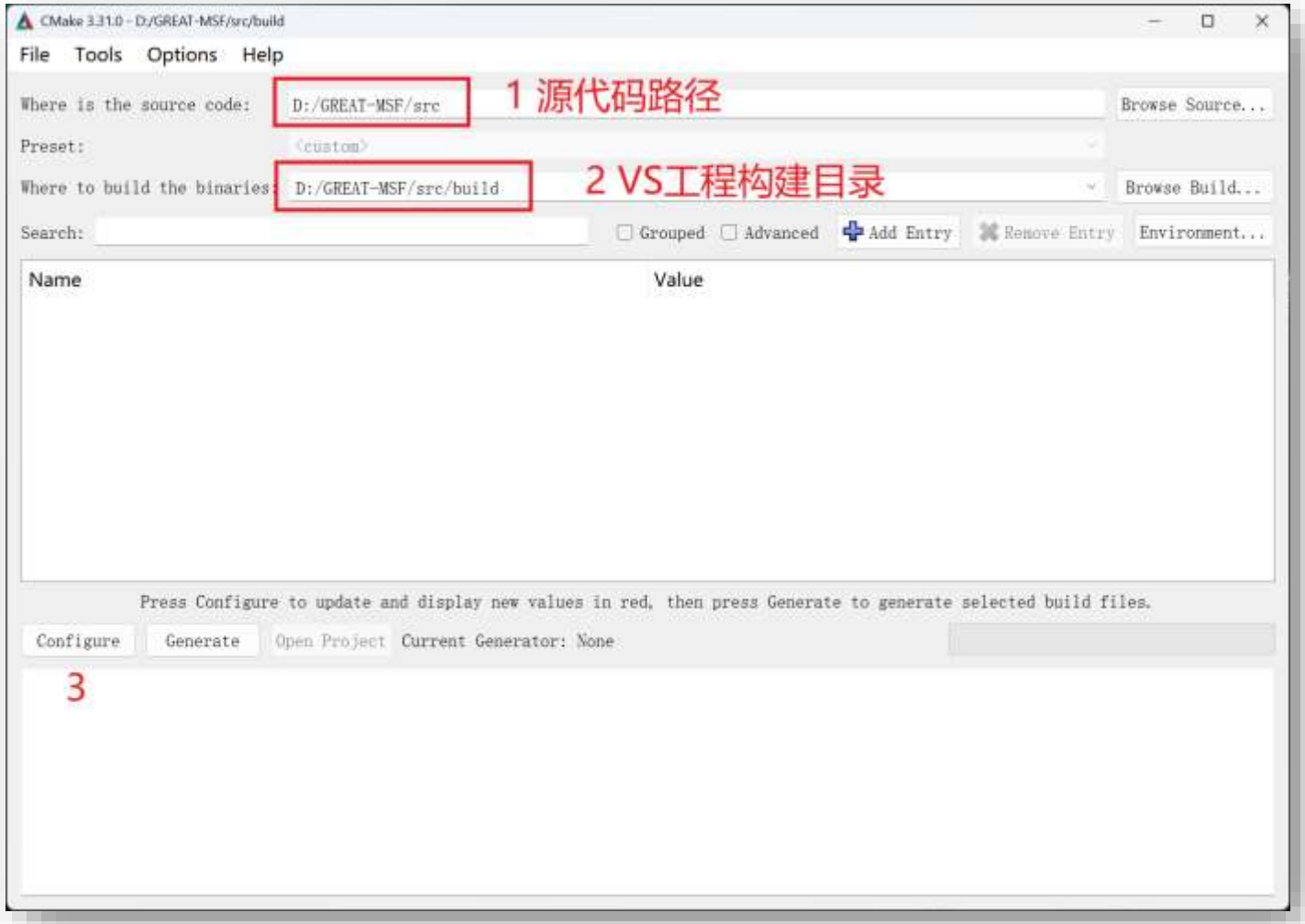
Windows下构建GREAT-MSF可执行程序

(1) 使用CMake GUI(Windows)

(2) 选择代码路径和工程构建路径

(3) 执行“Configure”

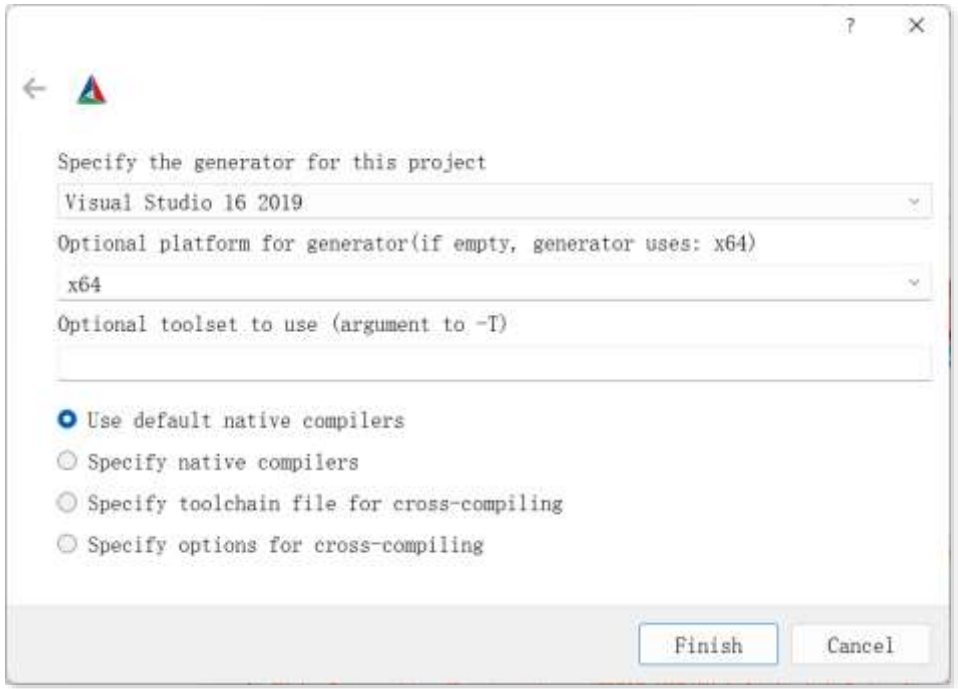




编译与安装

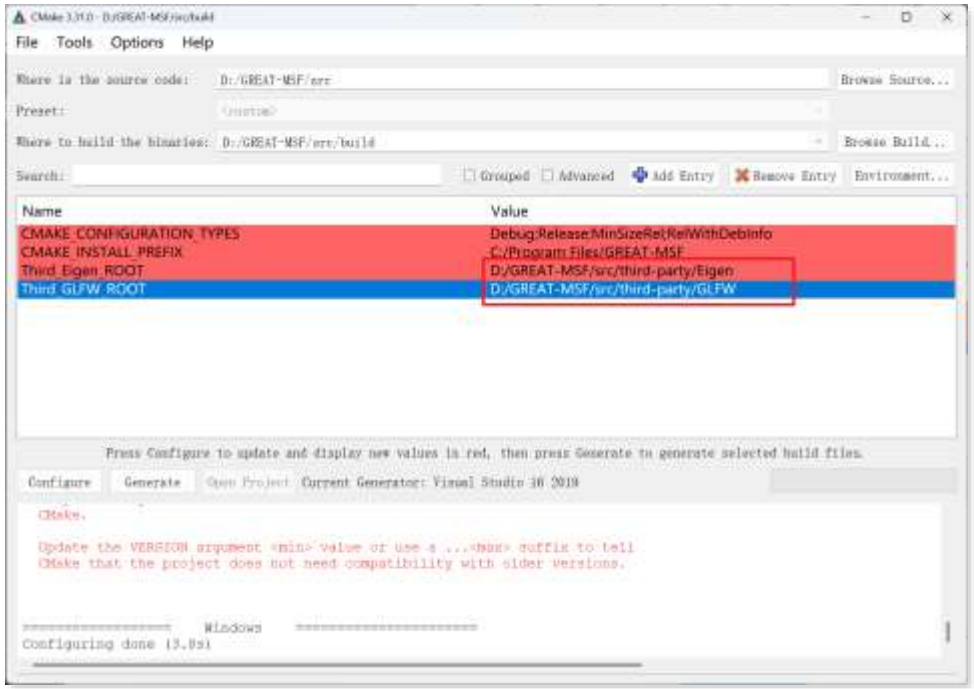
Windows下构建GREAT-MSF可执行程序

(4) 选择IDE类型



- ✓ 推荐使用VS2019及以上版本
- ✓ 推荐使用x64模式

(5) 配置三方库路径，执行“Generate”

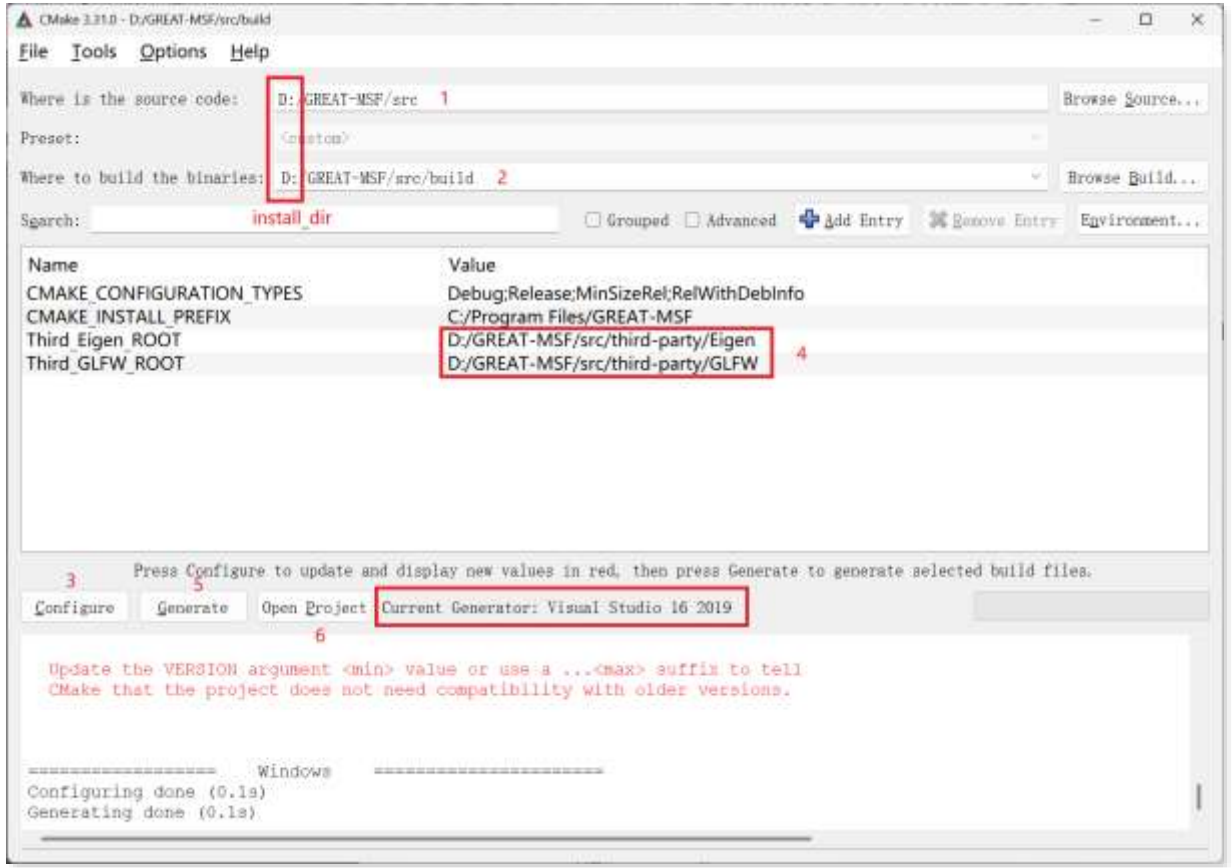


- ✓ 注意三方库路径到第一级目录即可

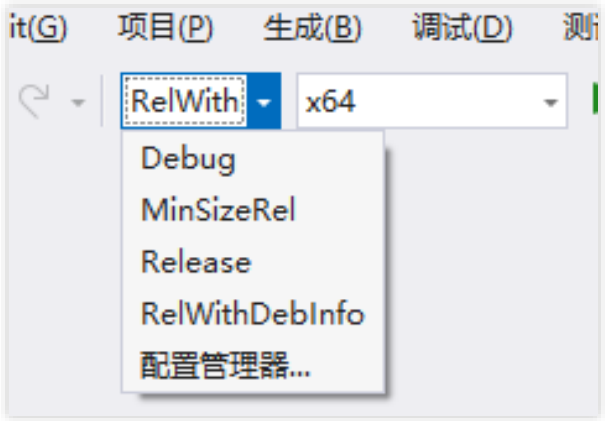
编译与安装

Windows下构建GREAT-MSF可执行程序

(6) 执行“Open Project”，然后在VS中编译源代码



VS中编译选项选择 RelWithDebInfo：



✓ **RelWithDebInfo**：Debug和Release的折中，有部分调试信息，速度也较快

PPP/INS算例数据处理

GREAT
GNSS+ REsearch, Application and Teaching

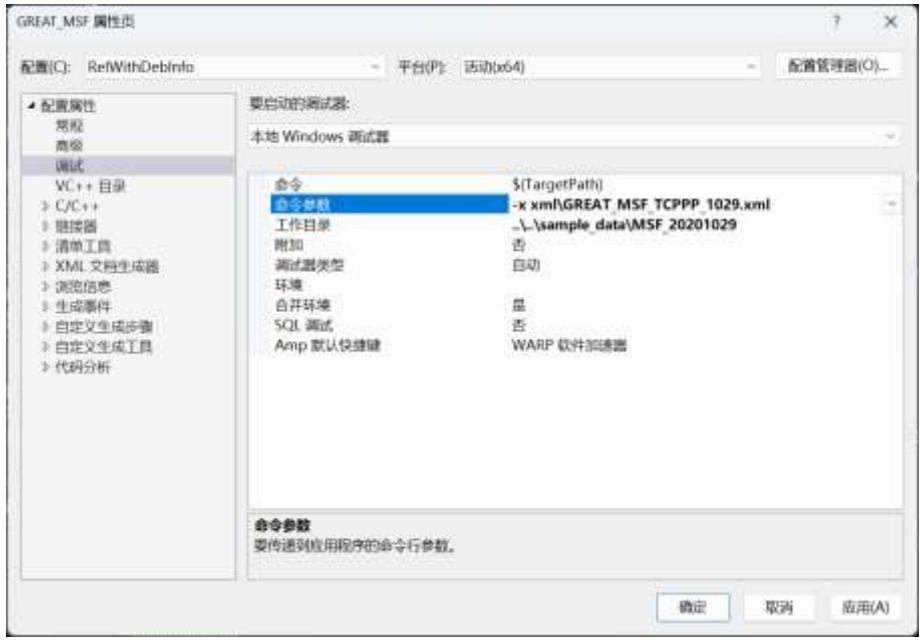
运行算例（VS）

(1) 打开VS，将GREAT_MSF设为启动项目

(2) 右键GREAT_MSF，选择属性，将算例所在目录设为工作目录，并设置命令参数

(3) 运行即可

第一步



第二步

第三步

PPP/INS算例数据处理

GREAT
GNSS+ REsearch, Application and Teaching

修改XML配置文件

<gen>总体控制信息节点：起止时间、时间间隔、系统、测站

```
<config>
<gen>
    <beg> "2020-10-29 05:59:00" </beg>
    <end> "2020-10-29 06:24:00" </end>
    <sys> GPS GAL BDS </sys>
    <rec>  SEPT </rec>
    <int>  1   </int>
</gen>
```

数据频率

<outputs>节点：输出文件

```
<outputs append="false" verb="1">
    <log type="BASIC" level="INFO"  />
    <flt> .\result\$(rec)-MSF.flt </flt>
    <ins> .\result\$(rec)-MSF.ins </ins>
    <kml> .\result\$(rec)-MSF.kml </kml>
</outputs>
```

<inputs>输入文件节点

```
<inputs>
  <rinexo>  .\GNSS\Rover\SEPT3030.20O      </rinexo>
  <rinexn>  .\GNSS\Product\BRDM00DLR_S_20203030000_01D_MN.rnx      </rinexn>
  <rinexc>  .\GNSS\Product\GBM0MGXRAP_20203030000_01D_30S_CLK.CLK  </rinexc>
  <sp3>     .\GNSS\Product\GBM0MGXRAP_20203030000_01D_05M_ORB.SP3  </sp3>
  <atx> .\model\igs14_2185.atx    </atx>
  <blq> .\model\oceanload         </blq>
  <DE>  .\model\jpleph_de405      </DE>
  <EOP> .\model\poleut1_2109_2201 </EOP>
  <imu> .\IMU\ADIS.txt            </imu>
</inputs>
```

系统模型文件

注意：

- 测站名与文件名需要吻合

  <rinexo>: hx01.20o → <rec>: HX01

PPP/INS算例数据处理

GREAT
GNSS+ REsearch, Application and Teaching

修改XML配置文件

<gps>节点：卫星设置

```
<gps sigma_C="2" sigma_L="0.02" >
 <freq>  1  2 </freq>
 <band> 1  2  </band>
</gps>
```

频带序号
观测值频带

<ambiguity>节点：模糊度固定参数设置

```
<ambiguity>
 <fix_mode> NO </fix_mode>
 <part_fix>  YES  </part_fix>
 <part_fix_num> 4 </part_fix_num>
 <ratio>  3  </ratio>
 <add_leo>  NO   </add_leo>
 <set_refsat> YES </set_refsat>
 <all_baselines>  NO  </all_baselines>
 <min_common_time>  1  </min_common_time>
 <baseline_length_limit>  3500  </baseline_length_limit>
 <widelane_interval> 30 </widelane_interval>
 <extra_widelane_decision  maxdev = "0.275" maxsig = "0.10" alpha = "1000" />
 <widelane_decision     maxdev = "0.275" maxsig = "0.10" alpha = "1000" />
 <narrowlane_decision   maxdev = "0.375" maxsig = "0.10" alpha = "1000" />
</ambiguity>
```

设置为NO时计算浮点解，
固定解设置为SEARCH

动态模式设置为true

<process>节点：GNSS解算处理过程设置

<filter>节点：滤波设置

```
<process>
 <phase>          true        </phase>
 <tropo>          true        </tropo>
 <iono>           false       </iono>
 <doppler>          false           </doppler>
 <tropo_model>            saastamoinen  </tropo_model>
 <sig_init_crd>           30            </sig_init_crd>
 <sig_init_vel>           10             </sig_init_vel>
 <sig_init_ztd>           10             </sig_init_ztd>
 <sig_init_amb>           30            </sig_init_amb>
 <sig_init_gal>           10            </sig_init_gal>
 <sig_init_glo>           10            </sig_init_glo>
 <sig_init_bds>           10            </sig_init_bds>
 <sig_init_vion>          100           </sig_init_vion>
 <minimum_elev>           7             </minimum_elev>
 <obs_combination>      IONO_FREE     </obs_combination>
 <max_res_norm>       5         </max_res_norm>
 <pos_kin>          true          </pos_kin>
 <sd_sat>           false         </sd_sat>
 <min_sat>          2       </min_sat>
 <obs_weight>         SINEL       </obs_weight>
 <basepos>            CFILE       </basepos>
 <bds_code_bias_corr>     true        </bds_code_bias_corr>
 <realtime>             false       </realtime>
 <slip_model>         default     </slip_model>
 <frequency>              2           </frequency>
</process>
```

频数

解算动态解

```
<filter
    method_flt="kalman"
    noise_crd="100"
    noise_vel="1"
    noise_clk = "1000"
    noise_dclk="100"
    noise_vion="100"
    rndwk_ztd="6"
    rndwk_amb="5"
    rndwk_glo = "20"
    rndwk_gal = "20"
    rndwk_bds = "20"
    rndwk_gps = "20"
/>
```

PPP/INS算例数据处理

GREAT
GNSS+ REsearch, Application and Teaching

修改XML配置文件

<ins>节点：设置惯导解算的数据格式、对准方式和初始状态

```
<ins>
 <ProcTime Start="0" End="100000000000"/>

 <DataFormat>
     <AxisOrder Type="garfu"  Format="garfu/gaflu/gafrd" />
     <GyroUnit Type="DPS" Format="DPS/RPS/RAD/DPH/RPH" />
     <AcceUnit Type="MPS2" Format="MPS/MPS2" />
     <Frequency Value ="100" Description="Hz" />
 </DataFormat>

 <Alignment Type="POS"  Format="OFF/STATIC/POS/VEL">
     <PositionVector Value="3" Description="m" />
     <VelocityVector Value="1" Description="m/s" />
     <CoarseAlignTime Value="300" Description="s" />
 </Alignment>
```

- ✓ 陀螺单位DPS和RPS分别表示度每秒与弧度每秒，H表示小时
- ✓ PositionVector是动态对准的基线长度阈值，两个历元间的水平位移大于所设定的值才被认为是载体是动态的，可以尝试对准；VelocityVector是允许速度对准的最小阈值；CoarseAlignTime是静态粗对准时长。
- ✓ 如果对准设置为OFF，则需要在节点InitialStates中输入惯导的初始状态。

```
<InitialStates>
    <Position Type="OFF" Format="OFF/Cartesian/Geodetic"  X="-2264975.053"  Y="5011133.582"  Z="3220185.787" Description="m"  />
    <Velocity Type="OFF" Format="OFF/Cartesian"  X="0.096" Y="4.552"  Z="-6.327"  Description="m/s" />
    <Attitude Type="OFF" Format="OFF/ON" Pitch="3.0932157894" Roll="0.1179490917" Yaw="165.921753953" Description="deg" />
    <GyroBias Type="OFF" Format="OFF/ON" X="401" Y="32"  Z="-85" Description="deg/h" />
    <AcceBias Type="OFF" Format="OFF/ON" X="0" Y="0"  Z="0" Description="mg" />
</InitialStates>
```

PPP/INS算例数据处理

GREAT
GNSS+ REsearch, Application and Teaching

修改XML配置文件

<integration>节点：配置组合算法参数，包括组合类型、杆臂、IMU误差模型

```
<integration>
    <GNSS Type="TCI" Format="OFF/LCI/TCI">
        <AntennaLever Type="RFU" Format="RFU" X="-0.05" Y="-0.41" Z="0.15" Description="m" />
        <DelayTime Value="0.01" Description="s"/>
        <LCISetting MinSat="5" MaxPDOP="3" />
    </GNSS>

    <Estimator >
        <IMUErrorModel Type="ADIS 16470"   Format="Customize/ADIS 16470/StarNeto"   />
        <Attitude InitialSTD="1,1,10" ProcNoiseSD="1,3,3" Description="deg" />
        <Velocity InitialSTD="1,1,1" ProcNoiseSD="1,1,1" Description="m/s" />
        <Position InitialSTD="10,10,10" ProcNoiseSD="0,0,0" Description="m" />
        <GyroBias InitialSTD="100,100,100" ProcNoiseSD="0,0,0" Description="deg/h" />
        <AcceBias InitialSTD="10,10,10" ProcNoiseSD="0,0,0" Description="mg" />
    </Estimator>
</integration>
```

- ✓ 使用TCI或LCI设置组合算法类型
- ✓ 坐标系以IMU为中心，指向天线
- ✓ DelayTime：IMU数据与GNSS数据之间允许的最大时间差。即在小于0.01s时认为IMU数据与GNSS数据实现同步。
- ✓ 将误差模型设置为Customize，下方对初始方差、过程噪声的设置才会生效。

RTK/INS算例数据处理

GREAT
GNSS+ REsearch, Application and Teaching

数据准备与XML配置修改

算例目录结构

目录	说明
./GNSS	GNSS数据文件夹
./IMU	IMU数据文件夹
./model	系统模型文件夹
./groudtruth	参考真值文件夹
./xml	xml配置文件夹
./result	结果输出文件夹

<gen>节点：起止时间、系统、**基准站与流动站**、时间间隔

```
<gen>
    <beg> "2020-10-29 05:59:00" </beg>
    <end> "2020-10-29 06:24:00" </end>
    <sys> GPS GAL BDS </sys>
    <rec>  SEPT R293</rec>
    <base> R293 </base>
    <rover> SEPT</rover>
    <int>  1   </int>
</gen>
```

<receiver>节点：基站坐标

```
<receiver>
    <rec id="R293" X="-2267776.01115" Y="5009357.43174" Z="3220982.10379"/>
</receiver>
```

注意：

- 处理过程节点<**process**>中的<**basepos**>节点“**CFILE**”表示从配置文件读取基站坐标，“**spp**”表示由标准单点定位算得基站坐标
- <**process**>中的一些其余参数可能需要根据实际情况做一些调整，例如中短基线就可不估计电离层误差，将<**iono**>设置为false

RTK/INS算例数据处理

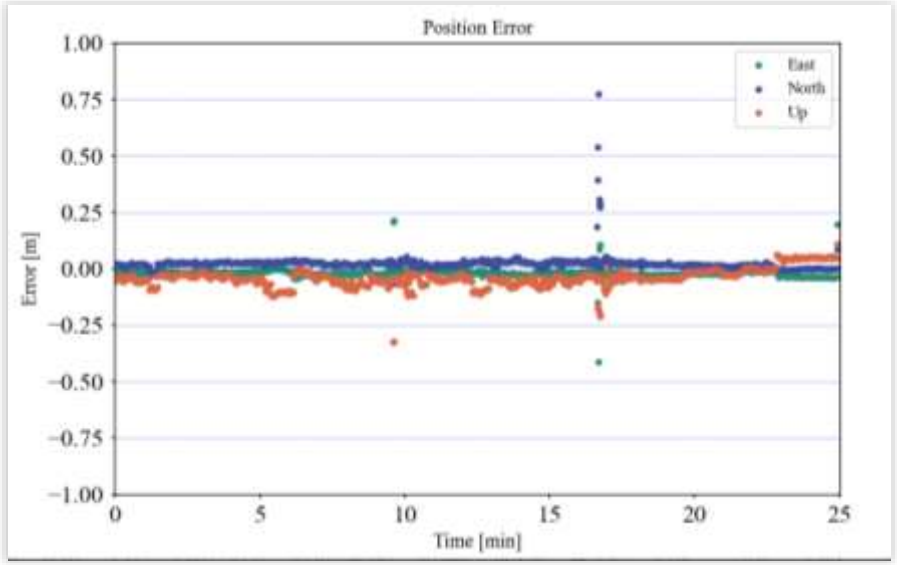
GREAT
GNSS+ REsearch, Application and Teaching

结果文件

- 程序运行完毕后在result文件夹中生成<site>-MSF.ins文件，同时生成log日志文件、卫星定位结果flt文件、供谷歌地球使用的kml轨迹文件

# Seconds of Week	X-ECEF	Y-ECEF	Z-ECEF	VX	VY	VZ	Pitch	Roll	Yaw	GyroBiasX	GyroBiasY	GyroBiasZ	AcceBiasX	AcceBiasY	AcceBiasZ	MeasType	Nsat	PDOP	AmbStatus	
# (s)	(m)	(m)	(m)	(m/s)	(m/s)	(m/s)	(deg)	(deg)	(deg)	(deg/h)	(deg/h)	(deg/h)	(mg)	(mg)	(mg)		#	#		
367144.000000	-2280982.459	5008159.677	3213507.912	-9.507	-4.109	-0.003	0.3023	0.6027	-90.1376	0.0281	-0.0521	-0.0098	-0.0584	-0.0378	0.0326	GNSS	23	1.20	Fixed	3.91
367145.000000	-2280992.046	5008155.453	3213507.896	-9.599	-4.239	0.039	-0.8756	0.7578	-89.7128	13.1876	-3.3472	0.0443	-1.3753	-5.0783	8.2498	GNSS	23	1.20	Fixed	3.86
367146.000000	-2281001.625	5008151.203	3213507.962	-9.597	-4.230	0.049	-0.8622	0.5917	-89.8021	-111.3782	60.8613	-2.4079	-1.2225	-5.1614	9.3489	GNSS	23	1.20	Fixed	3.90
367147.000000	-2281011.216	5008146.972	3213507.994	-9.584	-4.237	-0.002	-0.8788	0.5849	-90.1949	-339.7108	200.5666	-1.1697	-1.1876	-5.1560	9.3603	GNSS	23	1.20	Fixed	3.87
367148.000000	-2281020.802	5008142.716	3213508.045	-9.593	-4.265	0.109	-0.5798	0.6957	-89.7691	-494.4870	225.2750	-6.8426	-1.5014	-5.6708	9.7857	GNSS	23	1.20	Fixed	3.89

- 结果绘制



示例数据RTK/INS结果

载体轨迹图

主要内容

GREAT
GNSS+ REsearch，Application and Teaching

一、GREAT软件介绍

二、GREAT-MSF软件使用

三、GREAT-MSF代码讲解

四、答疑与交流

松组合基本原理

GNSS /INS 位置 速度 松组合

空间关系

$$\tilde{\boldsymbol{r}}^e_{ant} - \delta \boldsymbol{r}^e_{ant} = \tilde{\boldsymbol{r}}^e_{imu} - \delta \boldsymbol{r}^e_{imu} + (1+\boldsymbol{\phi}\times)\tilde{\boldsymbol{C}}^e_b \cdot \boldsymbol{l}^b_{ant}$$

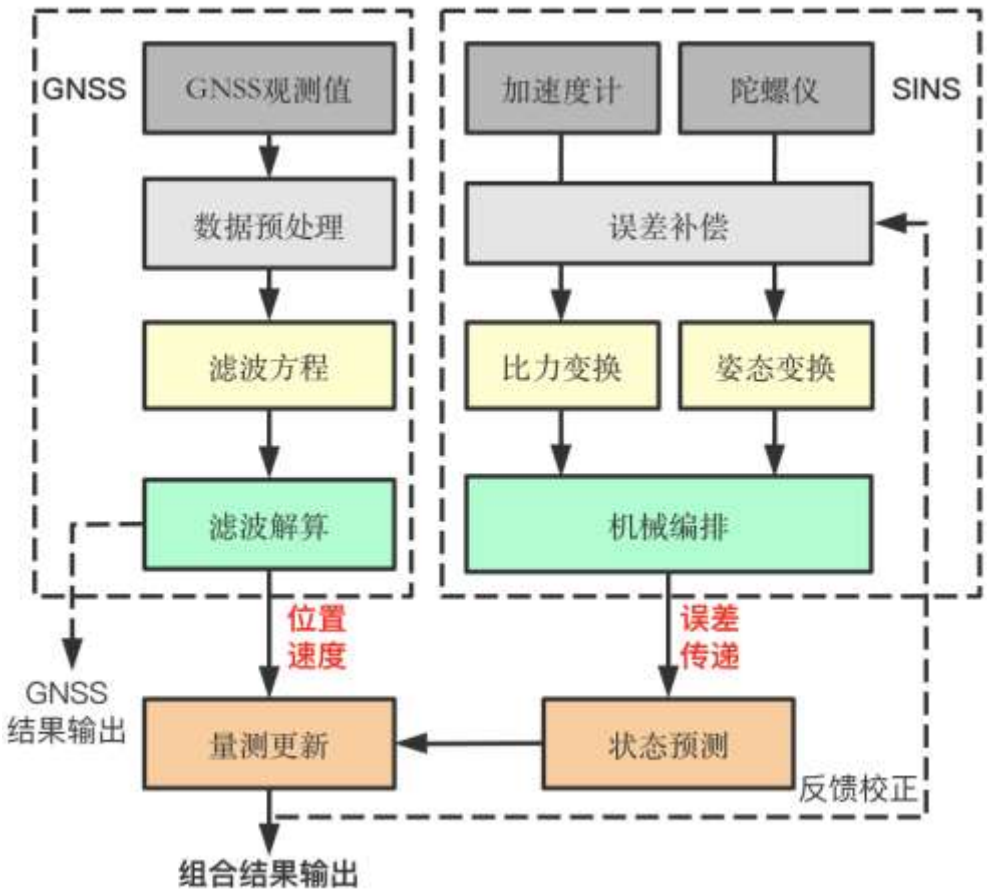
$$\Rightarrow \delta \tilde{\boldsymbol{r}}^e_{ant} = \delta \boldsymbol{r}^e_{imu} + \left(\tilde{\boldsymbol{C}}^e_b \cdot \boldsymbol{l}^b_{ant}\right)\times \boldsymbol{\phi}$$

速度杆臂补偿项 $\tilde{\boldsymbol{v}}^e_{trans} = \left[\tilde{\boldsymbol{C}}^e_b \cdot (\tilde{\boldsymbol{\omega}}^b_{ib}\times) - (\tilde{\boldsymbol{\omega}}^e_{ie}\times)\cdot\tilde{\boldsymbol{C}}^e_b\right]\cdot l^b_{gnss}$

量测模型

$$\tilde{\boldsymbol{p}}^e_b - \tilde{\boldsymbol{p}}^e_{ant} + \tilde{\boldsymbol{C}}^e_b \cdot \boldsymbol{p}^b_{ant} + \boldsymbol{\varepsilon}_p = \left(\tilde{\boldsymbol{C}}^e_b \cdot \boldsymbol{l}^b_{ant}\right)\times \boldsymbol{\phi}^e_b + \delta \boldsymbol{p}^e_b$$

$$\tilde{\boldsymbol{v}}^e_b - \tilde{\boldsymbol{v}}^e_{ant} + \tilde{\boldsymbol{v}}^e_{trans} + \boldsymbol{\varepsilon}_v = \tilde{\boldsymbol{v}}^e_{trans}\times \delta\boldsymbol{\phi}^e_b + \delta \boldsymbol{v}^e_b - \boldsymbol{C}^e_b \cdot (\boldsymbol{l}^b_{ant}\times)\cdot \delta\boldsymbol{\omega}^b_{ib}$$



✓ GNSS量测本质是 **位置/速度** 量测，因此对于其他方式获取的位置速度信息均可以通过此模型建立量测关系

PPP/INS定位基本原理

GREAT
GNSS+ REsearch, Application and Teaching

GNSS PPP/INS 伪距相位观测值紧组合(TCI)

✓ 合并PPP/INS 状态信息，**原始伪距相位**量测

状态合并

$$\boldsymbol{X}_{GNSS} = (\delta \boldsymbol{r}^e, \delta t, \delta Trop, \delta \boldsymbol{N}_i)^T \quad \sim \quad Q_{GNSS}$$

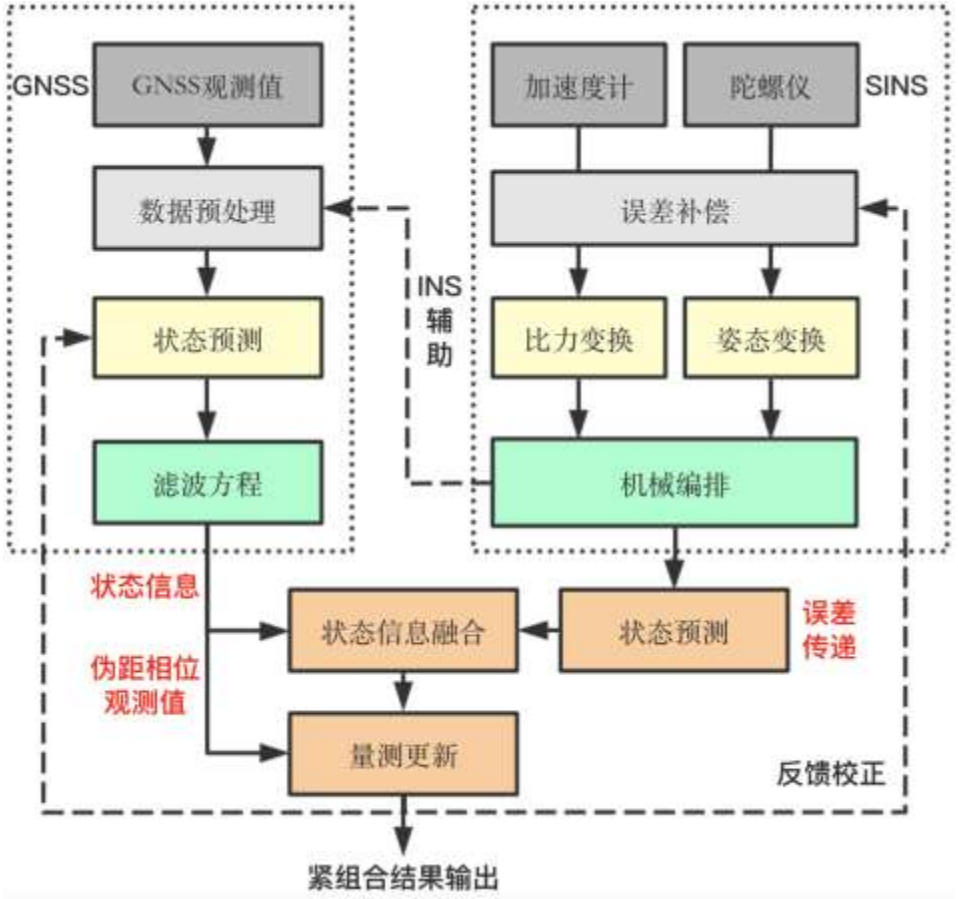
$$\boldsymbol{X}_{INS} = \left(\boldsymbol{\phi}^e, \delta \boldsymbol{v}^e, \delta \boldsymbol{r}^e, \delta \boldsymbol{b}_g, \delta \boldsymbol{b}_a\right)^T \sim \quad Q_{INS}$$

$$\boldsymbol{X} = \left(\boldsymbol{\phi}^e, \delta \boldsymbol{v}^e, \delta \boldsymbol{r}^e, \delta \boldsymbol{b}_g, \delta \boldsymbol{b}_a, \delta t, \delta Trop, \delta \boldsymbol{N}_i\right)^T \sim Q_{TCI}$$

量测模型

$$P - P_0 = \boldsymbol{n} \cdot \delta \boldsymbol{r}^e_{INS} + \boldsymbol{n}\left(\tilde{\boldsymbol{l}}^e \times\right)\boldsymbol{\phi}^e + \delta t + mf \cdot \delta Trop$$

$$L - L_0 = \boldsymbol{n} \cdot \delta \boldsymbol{r}^e_{INS} + \boldsymbol{n}\left(\tilde{\boldsymbol{l}}^e \times\right)\boldsymbol{\phi}^e + \delta t + mf \cdot \delta Trop + \delta N$$



✓ 紧组合利用了原始伪距相位观测值，从 GNSS/INS 整个滤波器的角度来看是一种**全局最优估计**，理论上具有更高的精度。

RTK/INS定位基本原理

GREAT
GNSS+ REsearch, Application and Teaching

GNSS RTK/INS 相位观测值紧组合(TCI)

状态合并

$$\boldsymbol{X}_{GNSS} = (\delta \boldsymbol{r}^e, \delta \boldsymbol{N}_i)^T \quad \sim \quad Q_{GNSS}$$

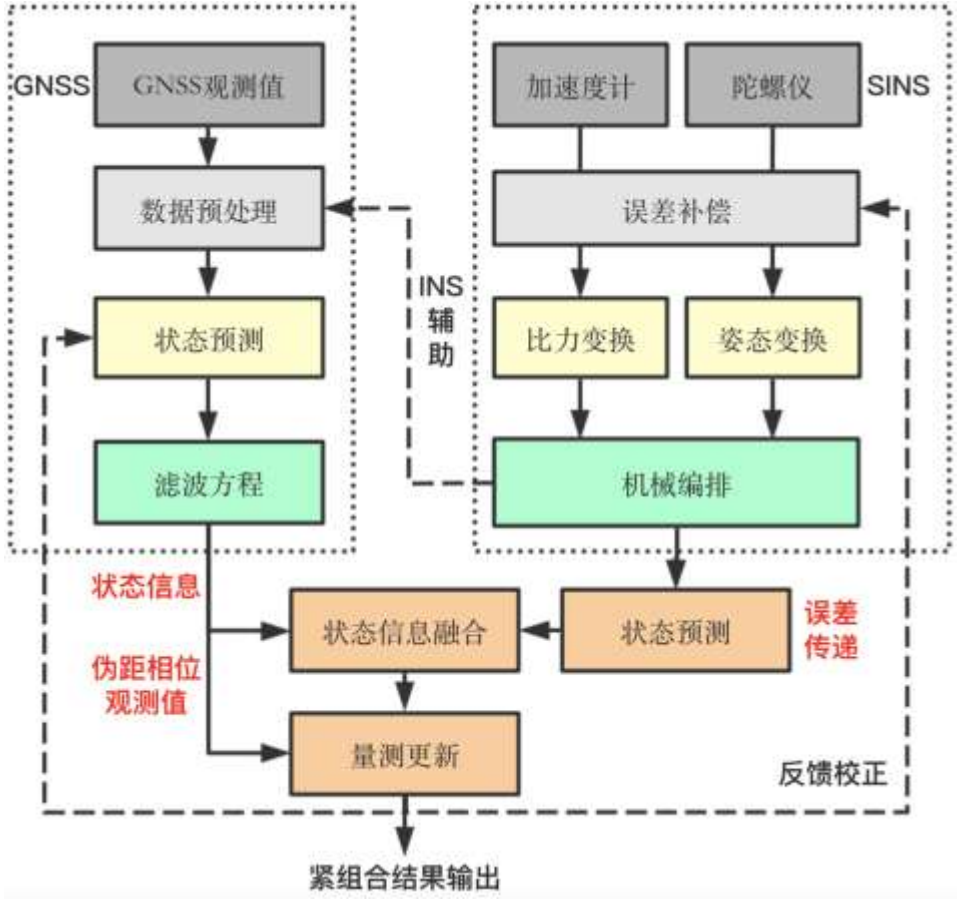
$$\boldsymbol{X}_{INS} = \left(\boldsymbol{\phi}^{\boldsymbol{n}}, \delta \boldsymbol{v}^n, \delta \boldsymbol{r}^n, \delta \boldsymbol{b}_g, \delta \boldsymbol{b}_a\right)^T \sim \quad Q_{INS}$$

$$\boldsymbol{X} = \left(\boldsymbol{\phi}^{\boldsymbol{n}}, \delta \boldsymbol{v}^n, \delta \boldsymbol{r}^n, \delta \boldsymbol{b}_g, \delta \boldsymbol{b}_a, \delta \boldsymbol{N}_i\right)^T \sim Q_{TCI}$$

量测模型

$$\nabla \Delta P = \boldsymbol{nA} \cdot \delta \boldsymbol{r}_{INS}^n + \boldsymbol{n}\left(\tilde{\boldsymbol{l}}^n \times\right)\boldsymbol{\phi}^n$$

$$\nabla \Delta L = \boldsymbol{nA} \cdot \delta \boldsymbol{r}_{INS}^n + \boldsymbol{n}\left(\tilde{\boldsymbol{l}}^n \times\right)\boldsymbol{\phi}^n + \nabla \Delta N$$



GREAT-MSF代码讲解

GREAT
GNSS+ REsearch, Application and Teaching

源码构成

GREAT-MSF代码讲解

GREAT
GNSS+ REsearch, Application and Teaching

□代码详解—main函数

GREAT_MSF.cpp

```
22  // MAIN
23  int main(int argc, char** argv)
24  {
25      // Only to cout the Reminder here
26      signal(SIGINT, catch_signal);
27
28      // Construct the gset class and init some values in the class
29      t_gcfg_ign gset;
30      gset.app("GREAT-MSF", "1.0.0", "$Rev: 2448 $", "(@whu,edu,cn)", __DATE__, __TIME__);
31      // Get the arguments from the command line
32      gset.arg(argc, argv, true, false);
33
34      // Creat and set the log file : ins.log
35      auto log_type = dynamic_cast<t_gsetout*>(&gset)->log_type();
36      auto log_level = dynamic_cast<t_gsetout*>(&gset)->log_level();
37      auto log_name = dynamic_cast<t_gsetout*>(&gset)->log_name();
38      auto log_pattern = dynamic_cast<t_gsetout*>(&gset)->log_pattern();
39      spdlog::set_level(log_level);
40      spdlog::set_pattern(log_pattern);
41      spdlog::flush_on(spdlog::level::err);
42      t_grtlog great_log = t_grtlog(log_type, log_level, log_name);
43      auto my_logger = great_log.spdlog();
44
45      bool isBase = false;
46      if (dynamic_cast<t_gsetgen*>(&gset)->list_base().size()) isBase = true;
47
48      // Prepare site list from gset
49      set<std::string> sites = dynamic_cast<t_gsetgen*>(&gset)->recs();
50      // Prepare input files list form gset
51      multimap<IFMT, std::string> inp = gset.inputs_all();
52      // Get sample intval from gset. if not, init with the default value
53      int sample = int(dynamic_cast<t_gsetgen*>(&gset)->sampling());
```

用来判断PPP/INS与RTK/INS

GREAT-MSF的入口main函数

① 获取配置文件参数

```
// Get the arguments from the command line
gset.arg(argc, argv, true, false);
```

② 逐文件数据读取

```
// DATA READING
multimap<IFMT, std::string>::const_iterator itINP = inp.begin();
for (size_t i = 0; i < inp.size() && itINP != inp.end(); ++i, ++itINP)
{
```

③ 进入组合解算

组合算法主函数

```
vgmsf[idx]->processBatchFB(beg, end, true);
```

④ 析构对象，释放内存

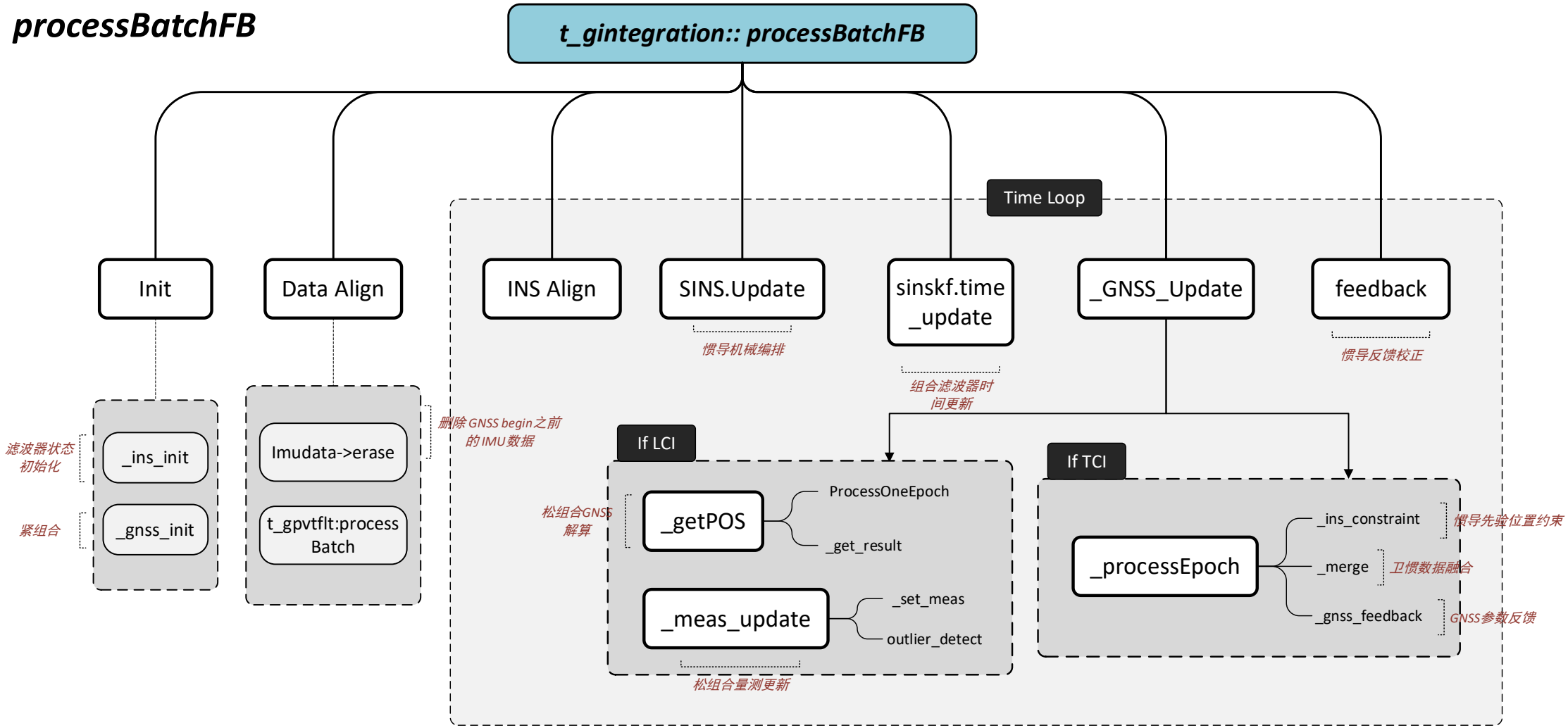
```
if (gobs) delete gobs;
if (gpcv) delete gpcv;
```

GREAT-MSF代码讲解

GREAT
GNSS+ REsearch, Application and Teaching

代码详解—processBatchFB函数

✓ 在 ***GREAT_MSF.cpp*** 的 *main* 函数中读取完 *xml* 文件与数据后进入组合算法主函数：***t_gintegration::processBatchFB***



GREAT-MSF代码讲解

GREAT
GNSS+ REsearch, Application and Teaching

代码详解—初始化

滤波方向

```
int great::t_gintegration::processBatchFB(const t_gtime& beg, const t_gtime& end, bool beg_end)
{
#ifdef BMUTEX
    boost::mutex::scoped_lock lock(_mutex);
#endif

    if (_grec == nullptr) { ... }

    t_gpvtflt::InitProc(beg, end);

    if (!this->_init()) {
        if (_spdlog) SPDLOG_LOGGER_ERROR(_spdlog, string("gintegration:  ") +  "init failed");
        return -1;
    }
```

起始时间　　结束时间

初始化函数入口

```
bool great::t_gsinskf::_ins_init()
{
    _imudata->interpolate(_resample_intv);
    kftk = sins.t; MeasVel = MeasPos = Eigen::Vector3d::Zero(), Flag = NO_MEAS;
    this->lever = dynamic_cast<t_gsetign*>(_setkf)->lever();
```

- ✓ 插值：规避imu输出中可能存在的少数历元数据丢失
- ✓ TimeLoop中的_merge_init使用GNSS和对准的结果来初始化惯导系统的状态

① 进入初始化过程 **this->_init()**

```
int great::t_gintegration::_init()
{
    if (!_ins_init() || !_gnss_init())
    {
        if (_spdlog) SPDLOG_LOGGER_ERROR(_spdlog, string("gintegrat
        return -1;
    }
```

② 在**_ins_init**中初始化滤波状态，包括插值、初始化误差协方差阵Pk和过程噪声协方差阵Qt

③ 如果组合模式是紧组合，会在**_gnss_init**函数中对一些必要参数变量进行维护，如对Qx进行扩维

```
bool great::t_gintegration::_gnss_init()
{
    if ((_ign_type == IGN_TYPE::TCI || _ign_type == IG
    {
        try
        {
            SymmetricMatrix Qx_extended;
```

GREAT-MSF代码讲解

GREAT
GNSS+ REsearch, Application and Teaching

代码详解—时间对齐

```
// time align
if (_ins_beg < _gnss_beg) {
    _ins_crt = _imudata->erase_bef(_gnss_beg);
}
else
{
    t_gpvtflt::processBatch(_gnss_crt, _ins_crt, false);
    if (_gnss_crt < _ins_crt) {
        int nEpo = round(_ins_crt.diff(_gnss_crt) / (_sampling) + 0.5);
        if (_sampling > 1) _gnss_crt.add_secs(int(_sampling * nEpo));  //  < 1Hz data
        else               _gnss_crt.add_dsec(_sampling * nEpo);       //  >=1Hz data
    }
}
```

t_gintegration:: processBatchFB

```
int great::t_gpvtflt::processBatch(const t_gtime &beg_r, const t_gtime &end_r, bool prtOut)
{
    _gmutex.lock();
```

t_gpvtflt::processBatch()，GREAT-PVT的算法主函数

将GNSS与INS数据做时间对齐处理

① 如果惯导数据开始时间较早，则调用函数 **erase_bef** 裁切掉早于GNSS开始时间的部分

```
t_gtime great::t_gimudata::erase_bef(t_gtime t, bool _beg_end)
{
    double t_ins;
    double t_gnss = t.sow() + t.dsec();
    t_ins = _imu_forward.front().t;
    while (t_ins < t_gnss)
```

② 如果卫星数据开始时间较早，则会进行纯GNSS定位，向flt文件写入定位结果，直到时间同步；通过输入GNSS当前历元作为起始时刻，INS当前历元作为结束时刻，以处理仅存在GNSS数据的片段。

GREAT-MSF代码讲解

GREAT
GNSS+ REsearch, Application and Teaching

代码详解—对准

```
bool great::t_gsinskf::cascaded_align(const Eigen::Vector3d& pos, const Eigen::Vector3d& vel)
{
    sins.imu.Update(_wm, _vm, _shm);

    Eigen::Vector3d blh = Cart2Geod(pos, false);
    Eigen::Vector3d vn = t_gbase::Cen(blh).transpose()*vel;
    sins.set_posvel(blh, vn);

    ALIGN_TYPE align_type = dynamic_cast<t_gsetins*>(_setkf)->align_type();
    bool ok = false;
    if (align_type == STC_AGN)
    {
        ok = align_coarse(_wm, _vm, _shm);
    }
    else if (align_type == VEL_AGN)
    {
        if (SQRT(SQR(vn(0)) + SQR(vn(1))) > 2)ok = align_vva(vn, _shm);
    }
    else if (align_type == POS_AGN)
    {
        ok = align_pva(pos, _shm);
    }
    if (ok)
        cerr << "\r" << _ins_crt.str_ymdhms("Alignment finished successfully: ") << endl;

    return ok;
```

静态粗对准

速度对准

位置对准

t_gsinskf::cascaded_align

```
bool great::t_gsinskf::align_pva(const Eigen::Vector3d& pos, const t_scheme & scm)
{
    if (_first_align)
    {
        _first_pos = _pre_pos = pos;
        _first_align = false;
        return false;
    }
```

初始化

t_gsinskf::align_pva

① 在TimeLoop中判断，如果还没有完成对准，则在验证时间有效性后获取GNSS的位置/速度解，进行动态对准。

② 在函数**align_pva**中，检查基线水平长度是否大于阈值，以检查载体是否处于运动状态；小幅变化无法进行有效对准。

```
// double yaw;
double dyaw = 10*t_gglv::deg;
double pos_dist = dynamic_cast<t_gsetins*>(_setkf)->pos_dist();
if (dist > pos_dist)
```

③ 判断当前航向角是否与上一航向角的差值足够小，若是则对准成功。

GREAT-MSF代码讲解

GREAT
GNSS+ REsearch, Application and Teaching

代码详解—机械编排，滤波器时间更新

```
void t_gsins::Update(const vector<Vector3d>& wm, const vector<Vector3d>& vm, const t_scheme& scm)
{
    nts = scm.ts;
    t = scm.t;
    nts = abs(nts);
    double nts_2 = nts / 2.0;

    imu.Update(wm, vm, scm);

    imu.phim = Kg.asDiagonal()*imu.phim - eb*nts; imu.dvbm = Ka.asDiagonal() *imu.dvbm - db*nts;
    Vector3d vn1_2 = vn + an*nts_2, pos1_2 = pos + eth.v2dp(vn1_2, nts_2);
    eth.Update(pos1_2, vn1_2);
    wib = imu.phim / nts; fb = imu.dvbm / nts;
    web = wib - Cbn*eth.wnie;
    wnb = wib - t_gquat::conj(qnb * t_gbase::rv2q(imu.phim / 2)) * eth.wnin;
    fn = qnb*fb;
    an = t_gbase::rv2q(-eth.wnin*nts_2)*fn + eth.gcc;
    Vector3d vn1 = vn + an*nts;
    pos = pos + eth.v2dp(vn + vn1, nts_2);    vn = vn1;
    qnb = t_gbase::rv2q(-eth.wnin*nts)*qnb*t_gbase::rv2q(imu.phim);
    Cnb = t_gbase::q2mat(qnb); att = t_gbase::m2att(Cnb); Cbn = Cnb.transpose(); vb = Cbn*vn;
    eth.Update(pos, vn); pos_ecef = Geod2Cart(pos, false);
    Ceb = eth.Cen*Cnb; Cbe = Ceb.transpose(); qeb = t_gbase::m2qua(Ceb); ve = eth.Cen*vn; ae = eth.Cen*an;
    pure_ins_time += nts;
}
```

t_gsins::Update

```
void great::t_gsinskf::time_update(double kfts, double inflation)
{
    set_Ft();
    Eigen::MatrixXd Fk = Eigen::MatrixXd::Identity(nq, nq) + (Ft * kfts);
    Phik = Fk * Phik;
    Xk = Fk * Xk;
    Eigen::MatrixXd Qk = (Qt * kfts * inflation).array().matrix().asDiagonal();
    Pk = Fk * Pk * (Fk.transpose());  Pk += Qk;
}
```

t_gsinskf::time_update

惯导机械编排函数

① 获取步长、时间等必要参数

② 对IMU输出进行圆锥误差和多项式误差补偿

imu.Update(wm, vm, scm);

③ 在导航系下进行常规的惯导算法

④ 更新四元数、方向余弦矩阵、速度等。

滤波器时间更新

① 计算系统状态转移矩阵 **Fk**

② 通过状态转移矩阵预测参数 **Xk**

③ 更新过程噪声矩阵 **Qk** 和误差协方差阵 **Pk**

GREAT-MSF代码讲解

代码详解—松组合量测更新

在TimeLoop中会检测是否存在有效量测

```
case GNSS_MEAS:   irc = _GNSS_Update();   break;
```

t_gintegration:: processBatchFB

```
t_gposdata::data_pos posdata;

if (_ign_type == IGN_TYPE::LCI)
{
    _global_variance = Pk;
    Flag = _getPOS(posdata);
    if (Flag != NO_MEAS) {
        _meas_update();
    }
}
```

t_gintegration::_GNSS_Update()

- ✓ **ProcessOneEpoch**函数调用时只需输入时间，函数内部会判断是PPP模式还是RTK模式
- ✓ 返回值： -1失败 0浮点 1固定

① 调用**_getPOS** 函数返回GNSS定位结果

```
int irc = ProcessOneEpoch(runEpoch);
if (irc < 0) {
    return MEAS_TYPE::NO_MEAS;
}
_get_result(runEpoch, pos);
```

对单个历元进行解算，参考GREAT_PVT

获取解算结果

② 在**_meas_update**中执行测量更新

```
if (Flag == NO_MEAS)  return -1;
Eigen::VectorXd Pxz = Eigen::VectorXd::Zero(Pk.rows()),
    Kk = Eigen::VectorXd::Zero(Pk.rows()), Hi = Xk = Eigen::VectorXd::Zero(Pk.rows());
this->_set_meas();

for (int i = 0; i < nr; i++)
{
    Hi = Hk.block(i, 0, 1, Pk.rows()).transpose();
    Pxz = Pk * Hi;
    double Pz0 = (Hi.transpose() * Pxz), r = Zk(i) - (Hi.transpose() * Xk);
    double Pzz = Pz0 + Rk(i, i);
    Kk = Pxz * (1.0 / Pzz);
    Xk += Kk * r;
    Pk = Pk - Kk * Pxz.transpose();
}

if (outlier_detect()!=0)
{
    Flag = NO_MEAS; return -1;
}
```

设置Zk和Rk

进行粗差探测，查看标准化残差是否超过设定阈值

GREAT-MSF代码讲解

GREAT
GNSS+ REsearch, Application and Teaching

代码详解——紧组合量测更新

在_processEpoch函数中实现GNSS/INS紧耦合解算

```
else if (_ign_type == IGN_TYPE::TCI)
{
    _timeUpdate(_gnss_crt);
    Flag = MEAS_TYPE(t_gintegration::_processEpoch(_gnss_crt));
```

t_gintegration::_GNSS_Update()

```
int great::t_gintegration::_processEpoch(const t_gtime& runEpoch)
{
    if (_grec == nullptr) { ... }

    _epoch = runEpoch;
    _amb_state = false;

    if (!_crd_xml_valid())  _sig_init_crd = 100.0;
```

t_gintegration::_processEpoch

```
_filter->add_data(_param, dx, _Qx, _sig_unit, Qsav);
_filter->add_data(A, P, l);
_filter->add_data(vtpv, nobs_total, npar_number);

Xk = Columns2VectorXd(dx);
_amb_resolution();
```

Xk：浮点解

在此函数估计固定解，结果存于_filter->dx中

t_gintegration::_processEpoch

① 接收机配置检查与初始化

② 数据预处理：移除不可信的卫星观测等

```
_remove_sat(outlier);
```

③ GNSS相关状态维护和方差预测

```
_predict(runEpoch);
```

④ 构建观测方程

```
iobs = _cmp_equ(equ);
equ.chageNewMat(A, P, l, nPar);
```

```
if (_merge_pose(A) < 0)
```

⑤ 滤波更新与后验残差计算

```
_filter->update(A, P, l, dx, _Qx);
```

⑥ 异常值检测：返回步骤②或输出结果

```
while (_outlierDetect(v_norm, Qsav, outlier) != 0);
```

GREAT-MSF代码讲解

GREAT
GNSS+ REsearch, Application and Teaching

代码详解—惯导反馈校正

```
void great::t_gintegration::_feedback(MEAS_TYPE type, bool successed)
{
    if (type == MEAS_TYPE::GNSS_MEAS && _ign_type == TCI)
    {
        t_gsins sins_out = sins;
        Eigen::VectorXd Xk_out = Xk;                    紧组合
        _global_variance = BaseMatrix2Eigen(_Qx);
        if (_amb_state) Xk = Columns2VectorXd(_filter->dx()).block(0, 0, nq, 1);
        this->feedback();
        if (_amb_state)this->_write();
        sins = sins_out; Xk = Xk_out;
    }
    else
    {
        if (successed) _global_variance = Pk;           松组合
        else  _global_variance = _gv_sav;
    }
```

t_gintegration::_feedback

① 根据测量类型进行惯导反馈校正：GREAT-MSF 的滤波选择维护浮点解，在紧组合情况下，模糊度成功固定则会在文件中输出固定解的结果，因此在本函数中存在sins和Xk的备份与回退

② 使用回退后的浮点解进行反馈矫正

```
Xk.conservativeResize(nq);
t_gsinskf::feedback();
Xk = Eigen::VectorXd::Zero(global_size);
_gv_sav = _global_variance;
```

③ PPP/INS模式则需要进入**_gnss_feedback**中反馈如对流层等参数

```
// GNSS feedback
if (_ign_type == IGN_TYPE::TCI)
{
    ColumnVector dx; dx.ReSize(global_size);
    dx = VectorXd2Columns(Xk);
    _gnss_feedback(dx);
}
```

GREAT-MSF代码讲解

GREAT
GNSS+ REsearch, Application and Teaching

代码详解—主要数据容器

this	0x000001fa41a5c040 {_ins_beg={_mjd_...	great::t_gintegration *
great::t_gsinskf	{kftk=367144.00000000000 nq=15 nr=6...	great::t_gsinskf
great::t_gpvtflt	{_dclkStoModel=0x000001fa3bc43220 ...	great::t_gpvtflt
gnut::t_gspp	{_grec=shared_ptr {_maprec={ size=2 } ...	gnut::t_gspp
__vfptr	0x00007ff8799ad858 {LibGREATrd.dll!v...	void * *
_ins_beg	{_mjd_conv=59151 _sod_conv=21540 _...	gnut::t_gtime
_gnss_beg	{_mjd_conv=59151 _sod_conv=21540 _...	gnut::t_gtime
_ins_end	{_mjd_conv=59151 _sod_conv=23040 _...	gnut::t_gtime
_gnss_end	{_mjd_conv=59151 _sod_conv=23040 _...	gnut::t_gtime
_gnss_crt	{_mjd_conv=59151 _sod_conv=21544 _...	gnut::t_gtime
_initial_merge	true	bool
_publisher	{viewer=0x000001fa3f7084c0 {window...	great::t_gpublish
_imu_state	{id=0 time=0.0000000000000000 orien...	great::IMUState

- ✓ 类t_gintegration虚拟继承类t_gsinskf（*捷联惯导滤波*）和类t_gpvtflt（*GNSS滤波*）

great::t_gsinskf	{_Cov_...	great::t_gsinskf
_Cov_MeasYaw	0.0000...	double
__vfptr	0x000...	void * *
kftk	36714...	double
nq	15	int
nr	6	int
measflag	0	int
Ft	{...}	Eigen::Matrix<dou..
Pk	{...}	Eigen::Matrix<dou..
Hk	{...}	Eigen::Matrix<dou..
Rk	{...}	Eigen::Matrix<dou..
Phik	{...}	Eigen::Matrix<dou..
Xk	{...}	Eigen::Matrix<dou..
Zk	{...}	Eigen::Matrix<dou..
Qt	{...}	Eigen::Matrix<dou..
Flag	NO_M...	great::MEAS_TYPE
_ign_type	TCI (2)	great::IGN_TYPE
lever	{...}	Eigen::Matrix<dou..
odo_lever	{...}	Eigen::Matrix<dou..
uwb_lever	{...}	Eigen::Matrix<dou..
MeasVel	{...}	Eigen::Matrix<dou..
MeasPos	{...}	Eigen::Matrix<dou..
MeasAtt	{...}	Eigen::Matrix<dou..
_Cov_MeasVn	{...}	Eigen::Matrix<dou..
_Cov_MeasPos	{...}	Eigen::Matrix<dou..
_Cov_MeasAtt	{...}	Eigen::Matrix<dou..

t_gsinskf 类内对象

sins	{nts=0....	great::t_gsins
nts	0.0100...	double
t	36714...	double
imu	{phim...	great::t_gimu
eth	{a=63...	great::t_gearth
att	{...}	Eigen::Matrix<dou.
vn	{...}	Eigen::Matrix<dou.
pos	{...}	Eigen::Matrix<dou.
an	{...}	Eigen::Matrix<dou.
fb	{...}	Eigen::Matrix<dou.
fn	{...}	Eigen::Matrix<dou.
wib	{...}	Eigen::Matrix<dou.
web	{...}	Eigen::Matrix<dou.
wnb	{...}	Eigen::Matrix<dou.
vb	{...}	Eigen::Matrix<dou.
eb	{...}	Eigen::Matrix<dou.
db	{...}	Eigen::Matrix<dou.
Kg	{...}	Eigen::Matrix<dou.
Ka	{...}	Eigen::Matrix<dou.
_tauG	{...}	Eigen::Matrix<dou.
_tauA	{...}	Eigen::Matrix<dou.

sins 类内对象

- ✓ GNSS参数的添加与删除、初值、初始方差的设置位于**_syncSys()**、**_syncIono()**、**_syncAmb()**等函数中，状态预测与过程噪声设置在**_predictCrd()**、**_predictClk()**、**_predictBias()**等函数中

GREAT-MSF代码讲解

GREAT
GNSS+ REsearch, Application and Teaching

代码详解—输出与调试函数

```
sins.prt_sins(os);
os << fixed << setprecision(0)
    << " " << setw(10) << meas            // meas
    << " " << setw(5) << nsat             // nsat
    << fixed << setprecision(2)
    << " " << setw(7) << _dop.pdop()      // pdop
    << fixed << setprecision(2)
    << " " << setw(8) << amb
    << setw(10) << (_amb_state ? _ambfix->get_ratio() : 0.0);
os << endl;
_fins->write(os.str().c_str(), os.str().size());
os.str("");
```

t_gintegration::_write

```
// output
os << fixed << setprecision(6) << setw(18) << t;
os << fixed << setprecision(3) <<
    setw(18) << pos_out(0) <<
    setw(18) << pos_out(1) <<
    setw(18) << pos_out(2);
os << fixed << setprecision(3) <<
    setw(10) << vel_out(0) <<
    setw(10) << vel_out(1) <<
    setw(10) << vel_out(2);
os << fixed << setprecision(4) <<
    setw(10) << att_out(0) <<
    setw(10) << att_out(1) <<
    setw(10) << att_out(2);
os << fixed << setprecision(4) <<
    setw(12) << eb_out(0) <<
    setw(12) << eb_out(1) <<
    setw(12) << eb_out(2);
os << fixed << setprecision(4) <<
    setw(12) << db_out(0) <<
    setw(12) << db_out(1) <<
    setw(12) << db_out(2);
```

t_gsins::prt_sins

```
void great::t_gsins::debug_ins_info()
{
    cout << "ins info : \n"
         << "      time: " << setw(16) << t << "\n"
         << "      att: " << setw(10) << att.transpose() / t_gglv::deg << "\n"
         << "      vel: " << setw(10) << ve.transpose() << "\n"
         << "      pos: " << setw(10) << pos_ecef.transpose() << "\n"
         << "      an: " << setw(10) << an.transpose() << "\n"
         << "      ae: " << setw(10) << ae.transpose() << "\n"
         << "      web: " << setw(10) << web.transpose() / t_gglv::dps << "\n"
         << "      wnb: " << setw(10) << wnb.transpose() / t_gglv::dps << "\n"
         << endl;
}
```

t_gsins::debug_ins_info

- ✓ 轨迹文件输出函数：t_ gsinskf:: _prt_ins_kml
- ✓ 经过**prt_sins**函数和**_write**函数，最终会在*.ins中输出位置、速度、姿态、陀螺与加表零偏、结果类型、卫星数、PDOP、模糊度状态和Ratio值

# Seconds of Week	X-ECEF	Y-ECEF	Z-ECEF	VX	VY	VZ	Pitch	Roll	Yaw	GyroBiasX	GyroBiasY	GyroBiasZ	AcceBiasX	AcceBiasY	AcceBiasZ	MeasType	Nsat	PDOP	AmbStatus	
#	(s)	(m)	(m)	(m)	(m/s)	(m/s)	(m/s)	(deg)	(deg)	(deg)	(deg/h)	(deg/h)	(deg/h)	(mg)	(mg)	(mg)		#	#	
	367144.000000	-2280982.459	5008159.677	3213507.912	-9.507	-4.109	-0.003	0.3023	0.6027	-90.1376	0.0281	-0.0521	-0.0098	-0.0584	-0.0378	0.0326	GNSS	23	1.20	Fixed	3.91
	367145.000000	-2280992.046	5008155.453	3213507.896	-9.599	-4.239	0.039	-0.8756	0.7578	-89.7128	13.1876	-3.3472	0.0443	-1.3753	-5.0783	8.2498	GNSS	23	1.20	Fixed	3.86
	367146.000000	-2281001.625	5008151.203	3213507.962	-9.597	-4.230	0.049	-0.8622	0.5917	-89.8021	-111.3782	60.8613	-2.4079	-1.2225	-5.1614	9.3489	GNSS	23	1.20	Fixed	3.90
	367147.000000	-2281011.216	5008146.972	3213507.994	-9.584	-4.237	-0.002	-0.8788	0.5849	-90.1949	-339.7108	200.5666	-1.1697	-1.1876	-5.1560	9.3603	GNSS	23	1.20	Fixed	3.87
	367148.000000	-2281020.802	5008142.716	3213508.045	-9.593	-4.265	0.109	-0.5798	0.6957	-89.7691	-494.4870	225.2750	-6.8426	-1.5014	-5.6708	9.7857	GNSS	23	1.20	Fixed	3.89

主要内容

GREAT
GNSS+ REsearch，Application and Teaching

一、GREAT软件介绍

二、GREAT-MSF软件使用

三、GREAT-MSF代码讲解

四、答疑与交流

答疑与交流

2025年3月2日

Things We Hope To Achieve:

更加稳定	More Stable
更加智能	More Intelligent
更加精密	More Precise
更加出彩	More Outstanding

Welcome To Follow!