

rtklib 深入解读

1 状态预测及周跳探测 (rtkpos/udstate)

`IO rtk_t *rtk : rtk solution structure`

`rtk->tt`: 更新时间间隔

`rtk->opt.mode`: 定位模式, 包括 `PMODE_KINEMA`, `PMODE_STATIC`, `PMODE_FIXED` 等

`rtk->opt.ru`: `PMODE_FIXED` 定位模式下流动站被固定的坐标

`rtk->opt.dynamics`: 状态参数里是否包括速度以及加速度

`rtk->opt.iono`: 是否为每颗卫星估计电离层

`rtk->ssat[i].outc[f]`: 每颗卫星各自频率的相位观测值接收数目?

`rtk->ssat[i].azel[1]`: 共视卫星在流动站站心坐标系下的高度角

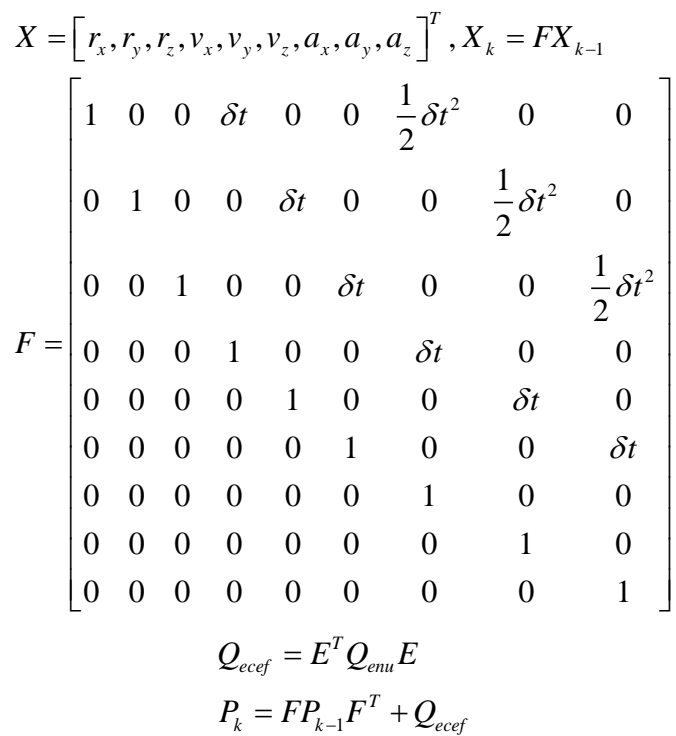
`rtk->ssat[i].slip[f]`: 最后一位表示周跳的标识, 1 为发生周跳; 前四位表示前一历元流动站和基准站的 LLI 标识

`rtk->ssat[i].gf[k]`: L1/L2 or L1/L5 的单差 GF 组合

`rtk->ssat[i].half[f]`: 只要任意站某颗卫星某个频点 LLI 前一位是 1, 设置为 0

`rtk->ssat[sat[i]-1].lock[k]`: ?

1.2 单差电离层状态量预测



1.2 单差电离层状态量预测

RTKLIB 为每一颗卫星赋予一个单差模糊度

1. 对所有卫星，如果 L1/L2 载波相位中断次数都大于 120，重置单差电离层状态量为 0
2. 对本次观测共视卫星，如果未初始化，设置其为 $1E-6$, $SQR(rtk->opt.std[1]*b1/1E4)$
3. 否则，加入过程噪声， $SQR(rtk->opt.prn[1]*b1/1E4*\cos(e1))*tt$

1.3 天顶方向对流层湿延迟状态量预测

RTKLIB 为每一个测站赋予一个天顶对流层湿延迟 (+2 个梯度)

TROPOPT_EST // TROPOPT_ESTG

1. 对基准站和流动站，如果未初始化，设置为 0.15 , $SQR(rtk->opt.std[2])$; 对于额外估计的梯度，设置为 $1E-6$, $SQR(0.001)$
2. 否则，加入过程噪声，对于天顶对流层湿延迟，设置为 $SQR(rtk->opt.prn[2])*tt$; 对于额外的梯度设置为 $SQR(rtk->opt.prn[2]*0.3)*fabs(rtk->tt)$

1.4 单差整周模糊度预测

opt.mode>PMODE_DGPS

```
#define PMODE_KINEMA 2          /* positioning mode: kinematic */
#define PMODE_STATIC 3         /* positioning mode: static */
#define PMODE_MOVEB 4          /* positioning mode: moving-base */
#define PMODE_FIXED 5          /* positioning mode: fixed */
```

1.4.1 周跳探测策略

RTKLIB 对两站所有共视卫星的所有存在频点分别探测

1. LLI 标识

&rtk->ssat[sat-1].slip[f]

LLI1:   ||||| LLI2:   ||||| 1: rover receiver; 2: base receiver

周跳情况 1: `obs[i].LLI[f]>0`

周跳情况 2: `LLI[i] || && !obs[i].LLI[f]`, 即前后历元前一位相悖

只要出现上述任一情况, 就可以对 **slip** 最后一位赋值 1, 即标识周跳

2. 单差 GF 组合 (L1/L2 or L1/L5)

①单频点单差相位②组建单差 GF 组合

$$GF = \lambda_1 \phi_1 - \lambda_2 \phi_2$$

周跳情况 3: 前后历元的 GF 组合差大于阈值

3. 多普勒禁用

1.4.2 重置单差模糊度 (0.0, 0.0) 的条件

对所有卫星

1. `opt.modear==ARMODE_INST`

2. `++rtk->ssat[i-1].outc[f]>(unsigned int)rtk->opt.maxout`

对共视卫星

3. 检测到周跳, 即**slip**最后一位是1; 另外, 如果使用IF组合观测值组建定位方程, 需要双频 (L1/L2) 周跳均无问题

2/3: `rtk->ssat[sat[i]-1].lock[f]=-rtk->opt.minlock;`

如果上述情况均未发生, 则保持模糊度, 仅仅增加过程噪声:

`rtk->P[j+j*rtk->nx]+=rtk->opt.prn[0]*rtk->opt.prn[0]*tt;`

1.4.3 设置未初始化的模糊度及修正已初始化的模糊度

`bias[i]=cp-pr/lami;`

`initx(rtk,bias[i],SQRT(rtk->opt.std[0]),IB(sat[i],f,&rtk->opt));`

✚ 计算 **bias** 和上一历元单差模糊度偏差 (有效) 的平均值, 加到已初始化的模糊度以进行修正?