



武汉大学

WUHAN UNIVERSITY



北斗单点定位解算 编程实验报告

院 系	测 绘 学 院
学 号	2021302141134
年 级	2 0 2 1 级
班 级	导 航 工 程 1 班
姓 名	刘 恒 祯
时 间	二〇二三年四月



目录

1. 任务要求	- 3 -
1.1 单历元最小二乘估计解算	- 3 -
1.2 静态卡尔曼滤波解算（静态逐次滤）	- 3 -
2. 算法原理	- 4 -
2.1 单历元最小二乘估计(Least Squares Estimation of Single Epoch) ..	- 4 -
2.2 静态卡尔曼滤波(Static Kalman Filter)	- 7 -
3. 程序设计	- 9 -
3.1 脚本调用结构	- 9 -
3.2 关键算法实现	- 10 -
①文件操作	- 10 -
②历元迭代与值更新	- 10 -
③数据处理	- 13 -
④绘图：柱形图、3d 散点图、散点-折线图	- 14 -
4. 运行结果	- 16 -
4.1 关键中间运行结果	- 16 -
4.2 单点定位解算结果	- 17 -
5. 结果可视化与评定	- 20 -
5.1 单历元最小二乘估计	- 20 -
5.2 静态卡尔曼滤波	- 23 -
5.3 精度评定	- 26 -
6. 实习感想	- 31 -
7. 附录	- 32 -
①main.m	- 32 -
②xyz2blh.m	- 40 -
③xyz2neu.m	- 41 -
④xyzDraw.m	- 42 -
⑤RclockdiffDraw.m	- 43 -
⑥neuDraw.m	- 44 -
⑦m0Draw.m	- 45 -
⑧blhpos3d.m	- 46 -
⑨histogram3.m	- 47 -
⑩histogram6.m	- 48 -



1. 任务要求

1.1 单历元最小二乘估计解算

- (1) 读入一个历元的所有观测值，建立观测方程和随机模型；
- (2) 设参数的近值，第一个历元可设为 (0 0 0 0)，线性化并迭代计算得到最小二乘估计（观测值独立），求验后单位权中误差，并利用验后单位权方差进行估计参数的精度评定；
- (3) 输出结果 X(k) Y(k) Z(k) T(k) 和验后方差矩阵按格式化输出到 txt 文档（格式化如下图）。

历元编号	GPS周	周秒	观测值数	
# 1	1547	5544.0000	7	
X (m)		Y (m)	Z (m)	T(m)
解算数值		解算数值	解算数值	解算数值
验后单位权中误差:		解算数值 (m)		
验后估计方差(m ²)				
解算数值		解算数值	解算数值	解算数值
解算数值		解算数值	解算数值	解算数值
解算数值		解算数值	解算数值	解算数值
解算数值		解算数值	解算数值	解算数值

# 2	1547	5545.0000	7	
X (m)		Y (m)	Z (m)	T(m)
解算数值		解算数值	解算数值	解算数值
验后单位权中误差:		解算数值 (m)		
验后估计方差(m ²)				
解算数值		解算数值	解算数值	解算数值
解算数值		解算数值	解算数值	解算数值
解算数值		解算数值	解算数值	解算数值
解算数值		解算数值	解算数值	解算数值

绿色文章部分为
解算结果

- (4) 重复(1)和(2)的步骤，（从第二个历元开始，可以将上一个历元解算的结果作为近似值）直到最后一个历元解算完成；

1.2 静态卡尔曼滤波解算（静态逐次滤）

静态逐次滤波解算的要求与（1.1）相同，输出并绘制结果，观测“静态逐次滤波”结果与“单历元解算”结果的差异（接收机坐标估计误差的差异和精度的差异），并分析为什么会有这样的差异。



2. 算法原理

单点定位问题抽象至数学问题：

①已知：卫星坐标矢量（星历给出）： $\vec{X}^s=(x^s, y^s, z^s)$

伪距观测值（实际观测）： ρ_r^s

②待求：接收机坐标矢量： $\vec{X}_r=(x_r, y_r, z_r)$

③关键：建立观测量与未知参数之间的函数关系（观测方程）：

$$\rho_r^s=f(\vec{X}^s, \vec{X}_r)$$

④解算：观测方程线性化，最小二乘模型解算。

2.1 单历元最小二乘估计 (Least Squares Estimation of Single Epoch)

伪距观测方程：

$$\tilde{\rho} = \rho - \delta_{t_R} \times c + \delta_{t^s} \times c - V_{ion} - V_{trop} \quad (1.1)$$

其中，伪距观测值：

$$\rho = \sqrt{(X_s - X)^2 + (Y_s - Y)^2 + (Z_s - Z)^2} \quad (1.2)$$

δ_{t^s} 经过星历改正， V_{ion} 与 V_{trop} 经过模型改正，故可视为已知量，得：

$$\tilde{\rho}^i = \sqrt{(X_s^i - X)^2 + (Y_s^i - Y)^2 + (Z_s^i - Z)^2} - \delta_{t_R} \times c + S^i \quad (1.3)$$

由于 X_s, Y_s, Z_s 为星历提供的已知量， c 为光速(常量)，

故 X, Y, Z, δ_{t_R} 为四个需要设参估计的量，即必要观测量为4，

故单历元内至少观测四个卫星，列出四个观测方程，即 $i \geq 4$ 。

公式(1.3)线性化得：

$$\tilde{\rho}^i = \rho_0^i - \frac{X_s^i - X_0}{\rho_0^i} x - \frac{Y_s^i - Y_0}{\rho_0^i} y - \frac{Z_s^i - Z_0}{\rho_0^i} z - \delta_{t_R} \times c + S^i \quad (1.4)$$



其中,

$$\rho_0^i = \sqrt{(X_s^i - X_0)^2 + (Y_s^i - Y_0)^2 + (Z_s^i - Z_0)^2} \quad (1.5)$$

若解算第1个历元, $(X_0 \ Y_0 \ Z_0)$ 的初值可设置为 $(0 \ 0 \ 0)$;

若解算第 n 个历元 $(n \geq 2)$, $(X_0 \ Y_0 \ Z_0)$ 的初值可设置为 $(X_{n-1} \ Y_{n-1} \ Z_{n-1})$;

结合公式 (1.4)、(1.5) 得:

$$\tilde{\rho}^i = \rho_0^i + l^i \cdot x + m^i \cdot y + n^i \cdot z - \delta_{t_R} \times c + S^i \quad (1.6)$$

公式 (1.6) 写为矩阵形式则有:

$$\begin{bmatrix} \tilde{\rho}^1 \\ \tilde{\rho}^2 \\ \dots \\ \tilde{\rho}^n \end{bmatrix} = \begin{bmatrix} l^1 & m^1 & n^1 & -1 \\ l^2 & m^2 & n^2 & -1 \\ \dots & \dots & \dots & \dots \\ l^n & m^n & n^n & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \delta_{t_R} \times c \end{bmatrix} + \begin{bmatrix} \rho_0^1 \\ \rho_0^2 \\ \dots \\ \rho_0^n \end{bmatrix} + \begin{bmatrix} S^1 \\ S^2 \\ \dots \\ S^n \end{bmatrix} \quad (1.7)$$

公式 (1.7) 写为误差方程, 有:

$$\begin{bmatrix} V_1 \\ V_2 \\ \dots \\ V_n \end{bmatrix} = \begin{bmatrix} l^1 & m^1 & n^1 & -1 \\ l^2 & m^2 & n^2 & -1 \\ \dots & \dots & \dots & \dots \\ l^n & m^n & n^n & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \delta_{t_R} \times c \end{bmatrix} - \left(\begin{bmatrix} \tilde{\rho}^1 \\ \tilde{\rho}^2 \\ \dots \\ \tilde{\rho}^n \end{bmatrix} - \begin{bmatrix} \rho_0^1 \\ \rho_0^2 \\ \dots \\ \rho_0^n \end{bmatrix} - \begin{bmatrix} S^1 \\ S^2 \\ \dots \\ S^n \end{bmatrix} \right) \quad (1.8)$$

用间接平差模型可表示为:

$$V = B\hat{x} - l \quad (1.9)$$

对给定的观测值方差阵 D , 取 σ_0 为某一合适值, 则 $P = Q^{-1} = (\frac{1}{\sigma_0^2} D)^{-1}$, 得到:

$$\hat{x} = (B^T P B)^{-1} B^T P l \quad (1.10)$$

单历元内逐次迭代 $(i=1, 2, 3, \dots)$, P 矩阵不变, 更新 $B_i \ l_i$ 矩阵, 计算改正数 \hat{x}_i :

$$\hat{x}_i = (B_i^T P B_i)^{-1} B_i^T P l_i \quad (1.11)$$

更新下次迭代初值:

$$X_{i+1}^0 = X_i^0 + \hat{x}_i \quad (1.12)$$



逐次收敛至第 n 次循环, $\hat{x} < threshold$ 时终止迭代, 得:

$$X_n = X_{n-1}^0 + \hat{x}_n \quad (1.13)$$

最终得到:

$$X = X_n \quad (1.14)$$

即为该历元解算的接收机位置坐标.

精度评定:

验后单位权中误差为:

$$\hat{\sigma}_0 = \sqrt{\frac{V^T P V}{n-t}} \quad (1.15)$$

参数的协因数矩阵为:

$$Q_{\hat{x}\hat{x}} = N_{BB}^{-1} = \begin{bmatrix} q_{XX} & q_{XY} & q_{XZ} & q_{Xt} \\ q_{YX} & q_{YY} & q_{YZ} & q_{Yt} \\ q_{ZX} & q_{ZY} & q_{ZZ} & q_{Zt} \\ q_{tX} & q_{tY} & q_{tZ} & q_{tt} \end{bmatrix} \quad (1.16)$$

参数的方差协方差阵为:

$$D_{\hat{x}\hat{x}} = \hat{\sigma}_0^2 Q_{\hat{x}\hat{x}} = \begin{bmatrix} \sigma_{XX}^2 & \sigma_{XY} & \sigma_{XZ} & \sigma_{Xt} \\ \sigma_{YX} & \sigma_{YY}^2 & \sigma_{YZ} & \sigma_{Yt} \\ \sigma_{ZX} & \sigma_{ZY} & \sigma_{ZZ}^2 & \sigma_{Zt} \\ \sigma_{tX} & \sigma_{tY} & \sigma_{tZ} & \sigma_{tt}^2 \end{bmatrix} \quad (1.17)$$



2.2 静态卡尔曼滤波(Static Kalman Filter)

递推公式总结：（线性化过程与单历元最小二乘估计解算相同，不再赘述）

$$X_{k+1}^* = \hat{X}_k$$

$$\Delta z_{k+1} = z_{k+1} - h_{k+1} \hat{X}_k$$

$$K_{k+1} = Q_{\hat{X}_k} h_{k+1}^T (w_{k+1}^{-1} + h_{k+1} Q_{\hat{X}_k} h_{k+1}^T)^{-1}$$

$$\hat{X}_{k+1} = \hat{X}_k + K_{k+1} \Delta z_{k+1}$$

$$Q_{\hat{X}_{k+1}} = Q_{\hat{X}_k} - K_{k+1} h_{k+1} Q_{\hat{X}_k}$$

$$v_{k+1} = h_{k+1} \hat{X}_{k+1} - z_{k+1}$$

$$\hat{\sigma}_{0,k+1}^2 = \frac{1}{(\ell_{k+1} - n)} \left[\hat{\sigma}_{0,k}^2 (\ell_k - n) + \Delta z_{k+1}^T K_{k+1}^T Q_{\hat{X}_k} K_{k+1} \Delta z_{k+1} + v_{k+1}^T w_{k+1} v_{k+1} \right]$$

$$\text{var}(\hat{X}_{k+1}) = \hat{\sigma}_{0,k+1}^2 Q_{\hat{X}_{k+1}}$$

$$\hat{\sigma}_0^2 = \frac{v_{k+1}^T w_{k+1} v_{k+1}}{\sum_{i=1}^{k+1} \ell_i - n}$$

$$\text{var}(\hat{X}) = \hat{\sigma}_0^2 Q_{\hat{X}_{k+1}}$$

递推最小二乘流程如下：

α) 第 1 组观测值解算

初组观测值解算与单历元最小二乘估计原理相同，解算得到：

$$\hat{X}_1 = X^0 + \hat{x}_1 \quad (2.1)$$

$$\hat{\sigma}_{0,1} = \sqrt{\frac{V_1^T W_1 V_1}{\ell_1 - n}} \quad (2.2)$$

$$Q_{\hat{x}_1 \hat{x}_1} = N_{B_1 B_1}^{-1} = (H_1^T W_1 H_1)^{-1} \quad (2.3)$$

$$D_{\hat{x}_1 \hat{x}_1} = \hat{\sigma}_{0,1}^2 Q_{\hat{x}_1 \hat{x}_1} \quad (2.4)$$



$\beta)$ 第 $k+1$ 组观测值解算($k \geq 1$)

\hat{X}_k 为第 k 组参数(改正数)解算估计值, 为第 $k+1$ 组参数(改正数)预测值:

$$X_{k+1}^* = \hat{X}_k \quad (2.5)$$

计算预测残差:

$$\Delta z_{k+1} = z_{k+1} - h_{k+1} \hat{X}_k \quad (2.6)$$

计算增益矩阵 K_{k+1} :

$$K_{k+1} = Q_{\hat{X}_k} h_{k+1}^T (w_{k+1}^{-1} + h_{k+1} Q_{\hat{X}_k} h_{k+1}^T)^{-1} \quad (2.7)$$

计算参数(改正数)更新值 \hat{X}_{k+1} :

$$\hat{X}_{k+1} = \hat{X}_k + K_{k+1} \Delta z_{k+1} \quad (2.8)$$

计算协因数矩阵 $Q_{\hat{X}_{k+1}}$:

$$Q_{\hat{X}_{k+1}} = Q_{\hat{X}_k} - K_{k+1} h_{k+1} Q_{\hat{X}_k} \quad (2.9)$$

计算验后单位权中误差 $\hat{\sigma}_{0,k+1}^2$:

$$\hat{\sigma}_{0,k+1}^2 = \frac{1}{(\ell_{k+1} - n)} \left[\hat{\sigma}_{0,k}^2 (\ell_k - n) + \Delta z_{k+1}^T K_{k+1}^T Q_{\hat{X}_k} K_{k+1} \Delta z_{k+1} + v_{k+1}^T w_{k+1} v_{k+1} \right] \quad (2.10)$$

组内最小二乘解算, 判断是否迭代同组数据:

if $\max |\hat{X}_{k+1} - X_{k+1}^*| > threshold$, *then* perform the iteration process.

else we have $k = k + 1$, output the \hat{X}_{k+1} $Q_{\hat{X}_{k+1}}$ $\hat{\sigma}_{0,k+1}$.

$\gamma)$ 静态逐次滤波解算完成, 得到最终参数改正数估计值 \hat{x}

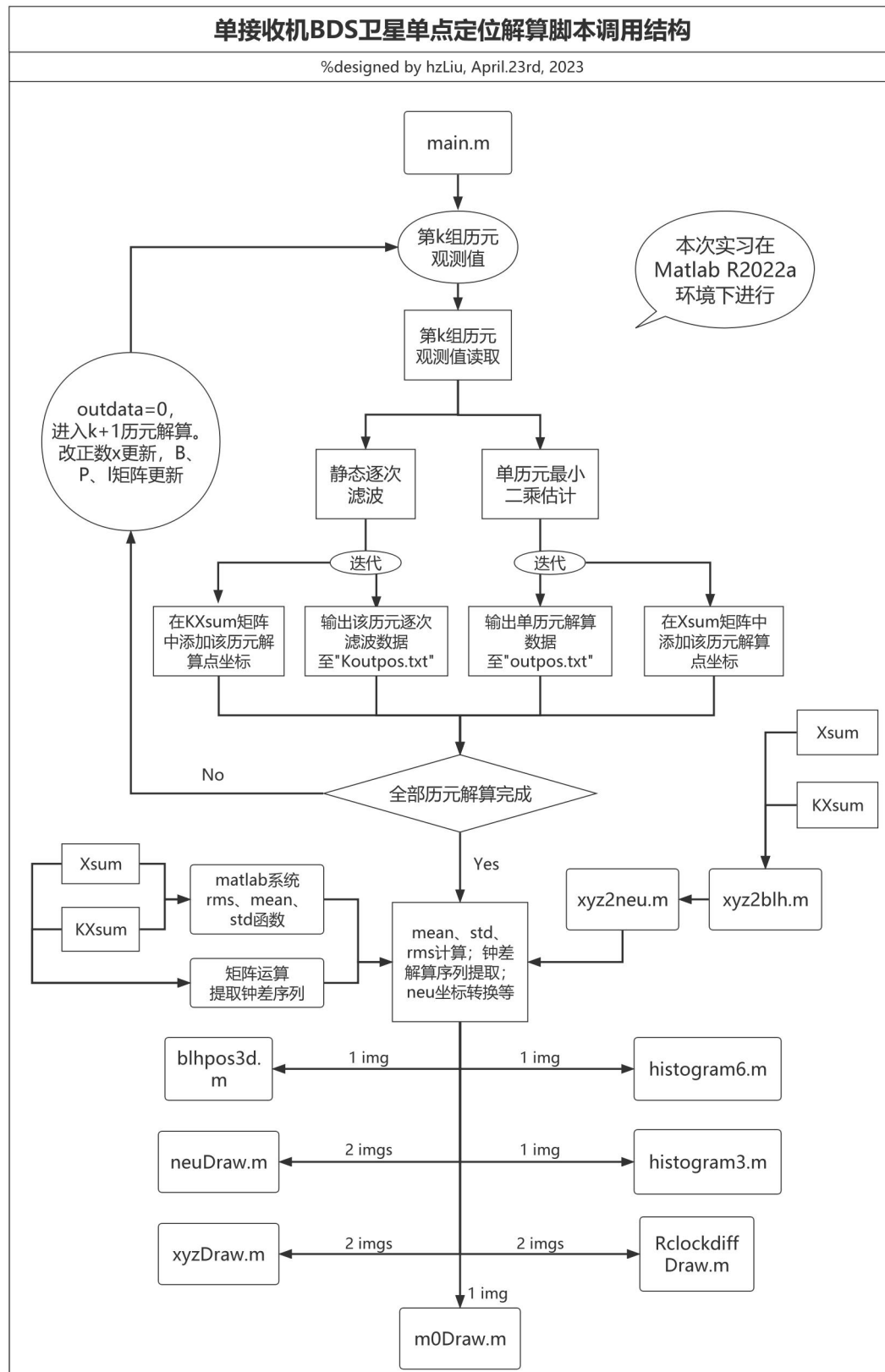
对给定初值 X^0 , 我们前面对改正数 \hat{x} 进行逐次滤波, 最终:

$$\hat{X} = X^0 + \hat{x} \quad (2.11)$$

得到静态逐次滤波的解 \hat{X} .

3. 程序设计

3.1 脚本调用结构





3.2 关键算法实现

①文件操作

```
%% read file
while ~feof(fid)
    nepoch=nepoch+1;
    tline=fgetl(fid);
    firstline=tline;%copy the first line
    if(strcmp(tline(1),'#'))
        %epoch, gpsweek, gpssecond, number of obs
        obshead=sscanf(tline(2:end),"%f");
        for i=1:obshead(4)
            tline=fgetl(fid);
            % satid=tline(1:3); %omit prn
            % X(m),Y(m),Z(m),pseudorange(m),variance(m)
            obsdata(i,:)=sscanf(tline(4:end),"%f",[1,5]);
        end
    end
    .....
end
```

重点是逐历元的**读取**。由于静态逐次滤波与单历元最小二乘的特性，逐历元计算具有吻合性与便捷性；同时，边读数据边计算避免了所有数据全部读入造成巨额的内存开支，提高了**计算效率**。

②历元迭代与值更新

```
%% calculate
%Static Kalman Filter
if nepoch>=2
    if nepoch==2
        X_est=kfX0.';
        Qxk=kfQx0;
        x_est=kfx0;
        m0k=kfm0;
        allsatsum=kfsatnum;
    end
    m=0;
    %construct B Q l
    for i=1:obshead(4)
        .....
    end
```



```
allsatsum=allsatsum+obshead(4);
w=inv(q);
%iteration: one time is enough in terms of precision
while abs(kmaxCorr)>1e-9&&m<1
    allsatsum=allsatsum+obshead(4);
    h=[kvecbx kvecby kvecbz kvecbc];
    deltaz=z-h*x_est;
    %deltaz=z-h*X_est;%wrong
    K=Qxk*h.'/(inv(w)+h*Qxk*h. ');
    xk1=x_est+K*deltaz;
    %Xk1=X_est+K*deltaz;%wrong
    Xk1=X_est+xk1;
    Qk1=Qxk-K*h*Qxk;
    vk1=h*xk1-z;
    m0k1=sqrt(1/(allsatsum-4)*(m0k*m0k*(allsatsum-obshead(4)-4)+deltaz.
        '*K.*Qxk*K*deltaz+vk1.*w*vk1));
    Dk1=m0k1*m0k1*Qk1;
    %renewal
    x_est=xk1;
    %X_est=Xk1;%no need for the estimated value renewal
    Qxk=Qk1;
    kmaxCorr=max(abs(K*deltaz));
    m=m+1;
end
end

%Single Epoch Cauculation by Least Squares
.....
while abs(maxCorr)>1e-9&&n<10
    %construct B Q l
    for i=1:obshead(4)
        .....
    end
    %P=inv(Q);
    x=((B. ')/Q*B)\((B. ')/Q*1);
    X(1)=X(1)+x(1);
    X(2)=X(2)+x(2);
    X(3)=X(3)+x(3);
    X(4)=X(4)+x(4);
    maxCorr=max(x(1:3));
    n=n+1;
end
V=B*x-l;
m0=sqrt((V. '/Q*V)/(obshead(4)-4));
```



```
Qxx=inv(B.' /Q*B);  
Dxx=m0*m0.*Qxx;  
Xsum=[Xsum;X];  
  
%extra code for 1st epoch of static kalman filter  
if nepoch==1  
    kfX0=X;  
    kfQx0=Qxx;  
    kfx0=x;  
    kfm0=m0;  
    kfDxx=(m0*m0). *kfQx0;  
    kfsatnum=obshead(4);  
    allsatsum=kfsatnum;  
    outdata1=[X(1),X(2),X(3),-1.*X(4)];  
    KXsum=[KXsum;X]  
end  
%% Clear the data and prepare the next epoch  
obsdata = obsdata * 0;
```

首先，迭代终止条件的设置。由于代码展示有限，我在 main.m 中设置了每历元（epoch）disp（“... ..”）展示静态卡尔曼滤波与单历元最小二乘迭代次数（iteration time），**命令行输出每历元迭代次数以可视化收敛过程**，这在（4.1）的中间运行结果的图片中可以看到命令行输出的迭代次数，便于我更改 threshold 调整精度。事实上，通过我 debug 调整改正数阈值然后对数据精度的分析，对单历元最小二乘设置为 $1e-7$ 便已足够（1-2 次迭代就足够了），对静态卡尔曼滤波设置 $1e-5$ ，迭代 1 次就够了。还需要注意到的是，迭代终止条件设置为两个之一不满足其中之一时，便跳出迭代。一个条件是改正数绝对值最大的一个 $\text{abs}(\text{maxCorr})$ ，体现收敛过程达到精度便跳出；和迭代次数 $n < 10$ ，避免非正常收敛导致无限次迭代。

其次，解算数据的更新。卡尔曼滤波的逐历元更新都是改正数更新，而单历元最小二乘的 B、1 矩阵的迭代是变化的，需要用到估计



值的更新，即我们的初值设定加上更新的改正数。也就是说 LS 与 KF 的迭代是有一点点不同的。对于单历元最小二乘，实际上单历元的初值可以任给合理值， $(0, 0, 0, 0)$ 也可以；但考虑到迭代速度和程序运行速度，便采用利用上一历元的解算估计值作为下一历元解算的初值 X_0 ，以加快收敛速度、程序运行速度与效率。

另外，对于静态卡尔曼滤波，还有：第一组观测值平差完全等同于单历元最小二乘，故我用来 `nepoch` 来标记历元次数，用 `if` 判断当 `nepoch==2` 时进入 `kalman filter`，而将单历元最小二乘 `nepoch==1` 的结果赋给 `kalman filter`，同时对于 `kalman filter` 的 `nepoch ≥ 2`，用 `nepoch==1` 的数据作 X^0 初值（我们说过，不论是哪种模型，迭代更新的只是改正数）。关键的还有变量作用域问题，需要全局声明一下空（矩阵）变量，详见代码。

③数据处理

```
%data processing
%transform negative clock difference to positive ones
Xsum(:,4)=-1.*Xsum(:,4);
for i=1:length(Xsum)
    neuPos=[neuPos;xyz2neu(StaPos,Xsum(i,1:3))];
    clock=[clock;Xsum(i,4)];
End
%transform negative clock difference to positive ones
KXsum(:,4)=-1.*KXsum(:,4);
for i=1:length(KXsum)
    KneuPos=[KneuPos;xyz2neu(KStaPos,KXsum(i,1:3))];
    Kclock=[Kclock;KXsum(i,4)];
end
Nmean=mean(neuPos(:,1));
Emean=mean(neuPos(:,2));
Umean=mean(neuPos(:,3));
rmsN=rms(neuPos(:,1));
rmsE=rms(neuPos(:,2));
```



```
rmsU=rms(neuPos(:,3));  
neuPos=[timeSeq neuPos];  
xyzPos=[timeSeq Xsum];  
clock=[timeSeq clock];  
m0sum=[timeSeq m0vec Km0vec];  
KNmean=mean(KneuPos(:,1));  
KEmean=mean(KneuPos(:,2));  
KUmean=mean(KneuPos(:,3));  
KrmsN=rms(KneuPos(:,1));  
KrmsE=rms(KneuPos(:,2));  
KrmsU=rms(KneuPos(:,3));  
KneuPos=[timeSeq KneuPos];  
KxyzPos=[timeSeq KXsum];  
Kclock=[timeSeq Kclock];
```

A. 为了绘图比较，我们需要构造出钟差的时序矩阵。因此，我们将周内秒提取出来单独作为列向量，与 Xsum（单历元最小二乘结果）、KXsum（静态卡尔曼滤波结果）中提取的钟差列向量合并，得到钟差时序矩阵 Kclock/clock。

B. 在两种解算方式的每个历元的解算中，将两种解算方式计算出的后验方差储存至列向量 m0vec/Km0vec 中。

C. 在遍历 Xsum/KXsum 的同时，将 xyz 坐标转换得到 neu 坐标（站心坐标为参考真值）。

D. 同样地，将时间序列合并得到失序坐标序列，得到 KxyzPos/xyzPos、KneuPos/neuPos。

另外，用 matlab 内置 mean、rms、std 函数，计算 neu 的均值与均方差 rms（因为测站真值已给出，站心坐标已知无秩亏），计算 xyz 的均值与标准差 std（样本无真值，自由度-1）。

④绘图：柱形图、3d 散点图、散点-折线图

```
% draw imgs  
KneuName="KneuPos";
```



```
neuDraw(KneuPos,KneuName);
KclockName="KReceiver clock difference";
RclockdiffDraw(Kclock,KclockName);
m0name="Posterior Unit Weight Error";
m0Draw(m0sum,m0name);
xyzName="xyzPos";
xyzDraw(xyzPos,xyzName);
KxyzName="KxyzPos";
xyzDraw(KxyzPos,KxyzName);
pos3dname="blhPos3d";
blhpos3d(Xsum(:,1),Xsum(:,2),Xsum(:,3),pos3dname,StaPos);
Kpos3dname="KblhPos3d";
blhpos3d(Xsum(:,1),Xsum(:,2),Xsum(:,3),Kpos3dname,KStaPos);
cats6=['stdX';'stdY';'stdZ';'rmsN';'rmsE';'rmsU'];
vals6=[stdX,stdY,stdZ,rmsN,rmsE,rmsU; ...
KstdX,KstdY,KstdZ,KrmsN,KrmsE,KrmsU];
histogramname6="Hist1_Kalman_SingleLeastSquares";
histogram6(cats6,vals6,histogramname6);
cats3=['Xmean';'Ymean';'Zmean';'Nmean';'Emean';'Umean'];
vals3=[Xmean,Ymean,Zmean,Nmean,Emean,Umean; ...
KXmean,KYmean,KZmean,KNmean,KEmean,KUmean];
histogramname3="Hist2_Kalman_SingleLeastSquares";
histogram3(cats3,vals3,histogramname3);
```

xyzDraw 画 xyz 时序图；neuDraw 画 neu 时序图；histogram3 画 1*2 的柱状图，体现精度信息；histogram6 画 1*1 的柱状图；blhpos3d 画 blh（WGS84）椭球下的 3D 坐标图，并以颜色深浅衡量高程；RclockdiffDraw 画钟差时序图；m0Draw 画验后单位权中误差时序图。

在具体实现的过程中，我花了很长时间优化界面的美观性，还有一点实现了消除时间中断折线图粘连的代码（这次的数据用不到，是卫星导航原理这门课画连续时段图的时候写的，拿过来注释一下）。具体函数详见代码。

我的代码文件由 1 个 main 脚本文件+9 个函数组成。除报告中附录给出以外，更多关于不连续时间处理以及写代码中的修改与注释的信息，具体请详见 '[./codeDir](#)'。



4. 运行结果

4.1 关键中间运行结果

当前文件夹

名称
animated.m
hist.fig
hist2.fig
histogram3.m
histogram6.m
main.m
neuDraw.m
pos3d.m
RclockdiffDraw.m
xyz2blh.m
xyz2neu.m

详细信息

工作区

名称	值
allsatsum	70259
ans	0
B	24x4 double
c	299792458
cats3	6x5 char
cats6	6x4 char
clock	2880x2 dou...
clockName	"Receiver clo...
deltaz	24x1 double
Dk1	4x4 double
Dxx	4x4 double
Emean	-0.1392
fid	3
filename	../txtDir/CU...
firstline	# 2880 217...

命令行窗口

```

KalmanEpoch 2877 Iteration Times: 1
Epoch 2877 Iteration Times: 3
KalmanEpoch 2878 Iteration Times: 1
Epoch 2878 Iteration Times: 6
KalmanEpoch 2879 Iteration Times: 1
Epoch 2879 Iteration Times: 2
KalmanEpoch 2880 Iteration Times: 1
Epoch 2880 Iteration Times: 7
Xmean=-1132915.0448m
Ymean=6092529.4173m
Zmean=1504633.46m
stdX=0.52645m
stdY=0.7874m
stdZ=0.28395m
Nmean=0.069443m
Emean=-0.13917m
Umean=0.94676m
rmsN=0.32m
rmsE=0.48908m
rmsU=1.2476m
KXmean=-1132914.9437m
KYmean=6092529.5397m
KZmean=1504633.3794m
KstdX=0.32284m
KstdY=0.30971m
KstdZ=0.17173m
KNmean=-0.033057m
KEmean=-0.2609m

```

KXsum

绘图		变量		视图	
+	打开	行	列		
根据所选内容新建	打印	1	1		
变量		所选内容			
2880x4 double					
	1	2	3	4	
1	-1.132914...	6.092528...	1.504633...	24.26412...	
2	-1.132914...	6.092528...	1.504632...	24.22530...	
3	-1.132915...	6.092529...	1.504632...	24.28628...	
4	-1.132915...	6.092529...	1.504632...	24.28633...	
5	-1.132915...	6.092529...	1.504632...	24.23787...	
6	-1.132915...	6.092529...	1.504632...	24.19821...	
7	-1.132915...	6.092529...	1.504632...	24.15139...	
8	-1.132915...	6.092529...	1.504632...	24.08869...	
9	-1.132915...	6.092529...	1.504632...	24.06278...	
10	-1.132915...	6.092529...	1.504632...	24.05496...	
11	-1.132915...	6.092529...	1.504632...	24.05131...	
12	-1.132915...	6.092529...	1.504632...	24.04265...	
13	-1.132915...	6.092529...	1.504632...	24.04213...	
14	-1.132915...	6.092529...	1.504632...	24.04563...	
15	-1.132915...	6.092529...	1.504632...	24.05536...	
16	-1.132915...	6.092529...	1.504632...	24.06840...	

Xsum

绘图		变量		视图	
+	打开	行	列		
根据所选内容新建	打印	1	1		
变量		所选内容			
2880x4 double					
	1	2	3	4	
1	-1.132914...	6.092528...	1.504633...	24.26412...	
2	-1.132914...	6.092529...	1.504632...	24.20020...	
3	-1.132915...	6.092529...	1.504632...	24.39440...	
4	-1.132915...	6.092529...	1.504632...	24.30742...	
5	-1.132915...	6.092529...	1.504632...	24.09269...	
6	-1.132915...	6.092529...	1.504632...	24.03726...	
7	-1.132915...	6.092529...	1.504632...	23.91733...	
8	-1.132915...	6.092529...	1.504632...	23.71532...	
9	-1.132915...	6.092529...	1.504632...	23.88719...	
10	-1.132915...	6.092529...	1.504632...	24.00266...	
11	-1.132915...	6.092529...	1.504632...	24.02643...	
12	-1.132915...	6.092529...	1.504632...	23.96851...	
13	-1.132915...	6.092529...	1.504632...	24.04453...	
14	-1.132915...	6.092529...	1.504632...	24.09277...	
15	-1.132915...	6.092529...	1.504632...	24.18733...	
16	-1.132915...	6.092529...	1.504632...	24.25093...	

4.2 单点定位解算结果

下面是我的工作目录文件展示：

最优估计\2021302141134-刘恒祯-北斗单点定位解算\txtDir

查看

电脑 > MyWork (G:) > 大二下 > 最优估计 > 2021302141134-刘恒祯-北斗单点定位解算 > txtDir

名称	修改日期	类型	大小
CUSV_20212220_BDS_sagnac.txt	2023/4/20 18:35	文本文档	5,575 KB
CUSV_20212220_BDS_sgnac_no_eph_error.t...	2023/4/20 18:46	文本文档	5,583 KB
CUSV20210222_2.txt	2023/4/20 16:48	文本文档	5,018 KB
CUSV20212220_BDS.txt	2023/4/17 11:44	文本文档	5,575 KB
Koutpos.txt	2023/4/29 23:19	文本文档	1,338 KB
outpos.txt	2023/4/29 23:19	文本文档	1,338 KB

图 1 观测值文件与输出文件

blhPos3d.png	2023/4/23 1:11	PNG 图片文件	186 KB
Hist1_Kalman_SingleL...	2023/4/23 1:11	PNG 图片文件	53 KB
Hist2_Kalman_SingleL...	2023/4/23 1:11	PNG 图片文件	84 KB
KblhPos3d.png	2023/4/23 1:11	PNG 图片文件	186 KB
KneuPos.png	2023/4/23 1:10	PNG 图片文件	56 KB
KReceiver clock differ...	2023/4/23 1:10	PNG 图片文件	49 KB
KxyzPos.png	2023/4/23 1:11	PNG 图片文件	91 KB
neuPos.png	2023/4/23 1:10	PNG 图片文件	134 KB
Posterior Unit Weight...	2023/4/29 23:19	PNG 图片文件	95 KB
Receiver clock differe...	2023/4/23 1:10	PNG 图片文件	82 KB
xyzPos.png	2023/4/23 1:10	PNG 图片文件	135 KB

图 2 图片绘制文件

blhpos3d.m	2023/4/23 18:52	MATLAB Code	2 KB
histogram3.m	2023/4/22 19:58	MATLAB Code	2 KB
histogram6.m	2023/4/23 1:25	MATLAB Code	1 KB
m0Draw.m	2023/4/29 23:18	MATLAB Code	3 KB
main.m	2023/4/29 23:19	MATLAB Code	11 KB
neuDraw.m	2023/4/22 16:29	MATLAB Code	3 KB
RclockdiffDraw.m	2023/4/23 2:02	MATLAB Code	2 KB
xyz2blh.m	2023/4/22 23:34	MATLAB Code	1 KB
xyz2neu.m	2023/4/20 21:58	MATLAB Code	1 KB
xyzDraw.m	2023/4/21 17:11	MATLAB Code	4 KB

图 3 代码文件



下面是我的结果输出文件展示（部分历元解算结果）：

Koutpos.txt	×	outpos.txt	×
0	10	20	30
3743	-0.0000	0.0001	0.0002
3744	0.0001	-0.0005	-0.0001
3745	# 417 2170 185280.000 24		
3746	X(m)	Y(m)	Z(m)
3747	-1132915.1396	6092529.7995	1504633.1752
3748	Posterior Unit Weight Error(m):	0.41990733	
3749	Posterior Estimated Variance(m^2):		
3750	0.0002	-0.0002	-0.0000
3751	-0.0002	0.0009	0.0001
3752	-0.0000	0.0001	0.0002
3753	0.0001	-0.0005	-0.0001
3754	# 418 2170 185310.000 24		
3755	X(m)	Y(m)	Z(m)
3756	-1132915.1404	6092529.7994	1504633.1757
3757	Posterior Unit Weight Error(m):	0.41957512	
3758	Posterior Estimated Variance(m^2):		
3759	0.0002	-0.0002	-0.0000
3760	-0.0002	0.0009	0.0001
3761	-0.0000	0.0001	0.0002
3762	0.0001	-0.0005	-0.0001
3763	# 419 2170 185340.000 24		
3764	X(m)	Y(m)	Z(m)
3765	-1132915.1411	6092529.7993	1504633.1762
3766	Posterior Unit Weight Error(m):	0.41955181	
3767	Posterior Estimated Variance(m^2):		
3768	0.0002	-0.0002	-0.0000
3769	-0.0002	0.0009	0.0001
3770	-0.0000	0.0001	0.0002
3771	0.0001	-0.0005	-0.0001
3772	# 420 2170 185370.000 24		
3773	X(m)	Y(m)	Z(m)
3774	-1132915.1419	6092529.7993	1504633.1767
3775	Posterior Unit Weight Error(m):	0.41945827	
3776	Posterior Estimated Variance(m^2):		
3777	0.0002	-0.0002	-0.0000
3778	-0.0002	0.0009	0.0001
3779	-0.0000	0.0001	0.0002
3780	0.0001	-0.0005	-0.0001

图 4 静态卡尔曼滤波解算结果图



Koutpos.txt	×	outpos.txt	×	
				0 10 20 30 40 50 60
11831		Posterior Estimated Variance(m ²):		
11832		0.0916	-0.0431	-0.0089 0.0244
11833		-0.0431	0.4971	0.1107 -0.2811
11834		-0.0089	0.1107	0.1124 -0.0747
11835		0.0244	-0.2811	-0.0747 0.1879
11836		# 1316 2170 212250.000 25		
11837		X(m)	Y(m)	Z(m) T(m)
11838		-1132915.4239	6092529.7821	1504633.7488 -19.2792
11839		Posterior Unit Weight Error(m): 0.45572193		
11840		Posterior Estimated Variance(m ²):		
11841		0.0814	-0.0371	-0.0078 0.0208
11842		-0.0371	0.4427	0.0994 -0.2499
11843		-0.0078	0.0994	0.1010 -0.0671
11844		0.0208	-0.2499	-0.0671 0.1670
11845		# 1317 2170 212280.000 25		
11846		X(m)	Y(m)	Z(m) T(m)
11847		-1132915.3730	6092529.6768	1504633.7456 -19.1764
11848		Posterior Unit Weight Error(m): 0.46972816		
11849		Posterior Estimated Variance(m ²):		
11850		0.0859	-0.0383	-0.0081 0.0213
11851		-0.0383	0.4677	0.1057 -0.2637
11852		-0.0081	0.1057	0.1075 -0.0713
11853		0.0213	-0.2637	-0.0713 0.1763
11854		# 1318 2170 212310.000 25		
11855		X(m)	Y(m)	Z(m) T(m)
11856		-1132915.4223	6092529.7839	1504633.7449 -19.2278
11857		Posterior Unit Weight Error(m): 0.44441068		
11858		Posterior Estimated Variance(m ²):		
11859		0.0766	-0.0335	-0.0071 0.0185
11860		-0.0335	0.4169	0.0947 -0.2349
11861		-0.0071	0.0947	0.0964 -0.0639
11862		0.0185	-0.2349	-0.0639 0.1570
11863		# 1319 2170 212340.000 25		
11864		X(m)	Y(m)	Z(m) T(m)
11865		-1132915.3947	6092529.7037	1504633.7380 -19.0887
11866		Posterior Unit Weight Error(m): 0.45266634		
11867		Posterior Estimated Variance(m ²):		
11868		0.0791	-0.0342	-0.0072 0.0188

图 5 单历元最小二乘估计解算结果图



5. 结果可视化与评定

5.1 单历元最小二乘估计

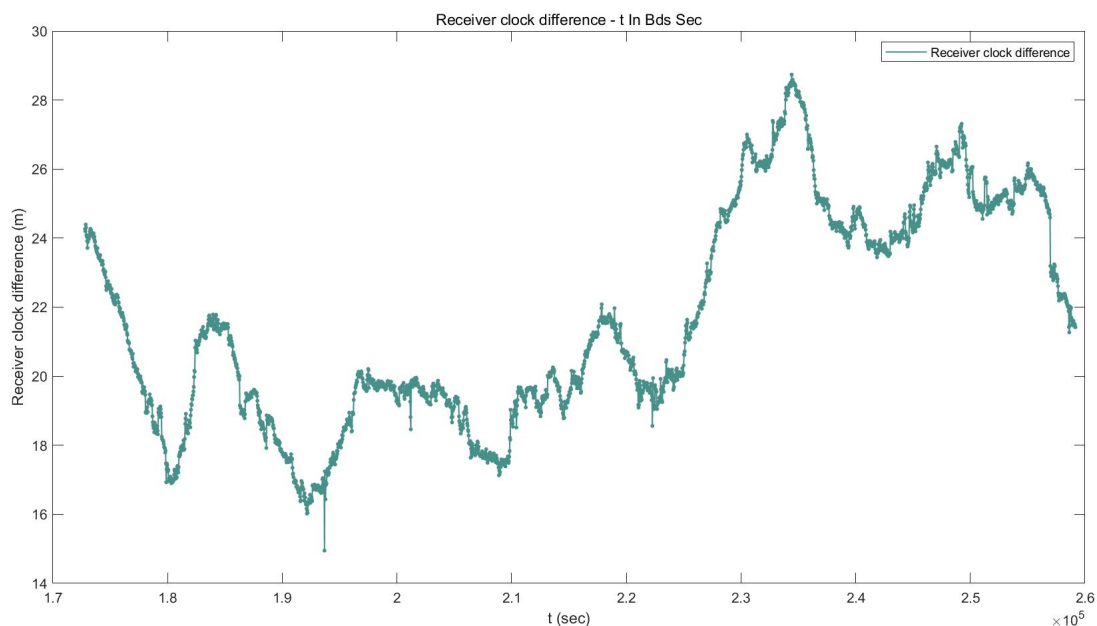


图 1 钟差时序图

从钟差时序图中可以看出，单历元解算的观测值方程较少，认为历元间观测值是独立的。由于噪声等因素，曲线不平滑，抖动跳跃较大。20m 左右的量级除以光速 $c=299792458\text{m/s}$ ，也是合理的。

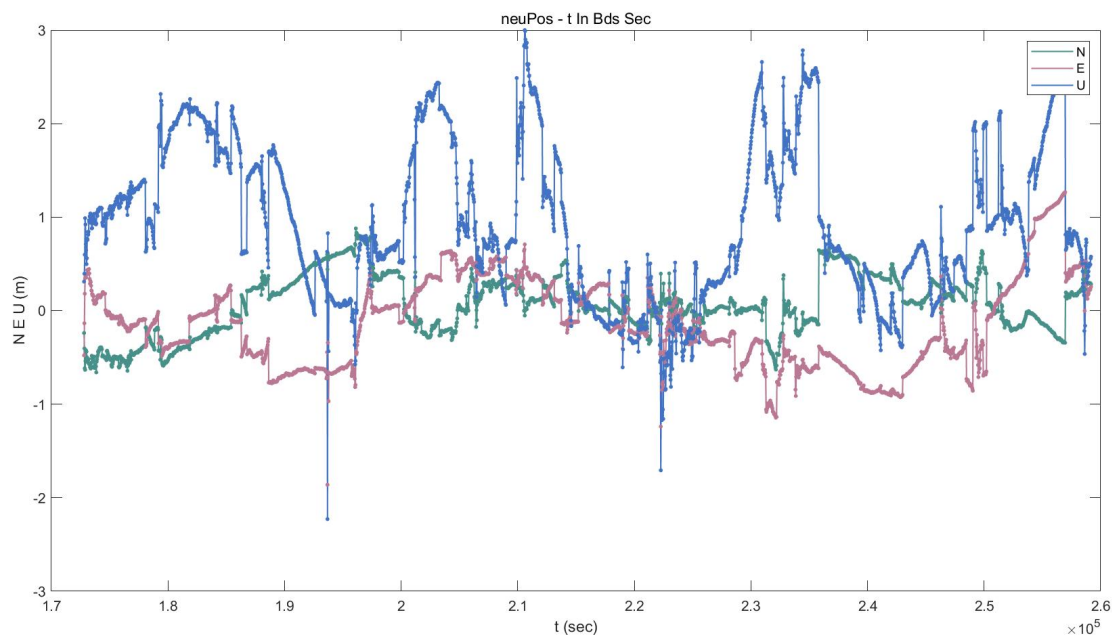


图 2 neu 坐标时序图



以给定的接收机坐标真值为站心，neu 曲线的纵轴绝对值的大小正比与观测误差。我们可以看到，北方向 N、东方向 E 的误差在均值 0 附近抖动，而天顶方向的 U 坐标却总体均值明显不为零，而且随时间变化抖动幅度更大。

分析其原因，天顶方向的信号（电磁波）需要穿过完整的大气。主要矛盾在于天顶方向的大气延迟：对流层、电离层误差。

对于对流层误差，大气折射效应主要体现在电磁波信号在穿过大气时，速度将发生变化，传播路径也会发生弯曲。对流层延迟在天顶方向可达 2.5m；在低高度角时可达 30-40m。对流层延迟引起的平面定位误差 $\pm 2\text{m}$ 左右，且呈随机性变化；而在高程方向引起的偏差可达到 $\sim 10\text{m}$ ，且表现为系统性误差。即便已经使用 Hopfield 等模型等进行系统性误差改正（经验模型改正的主要是占 80% 的干分量），而湿分量虽小却由于水汽的不确定因素和不规则变化导致的湿分量的模型改正精度和可信度受到限制，仍在天顶方向表现出较大误差。

对于电离层误差，若经过经验模型改正（Klobuchar 模型）也仅仅能达到改正 60% 左右的延迟，而且每时每刻天顶方向的总电子含量 TEC 也是变化的，更导致了误差的增加。若使用其他方式改正（我不知道是不是双频接收机，如果是的话）由误差传播定律与正态分布高斯噪声的特性，在噪声方面也会增大，使误差增大。

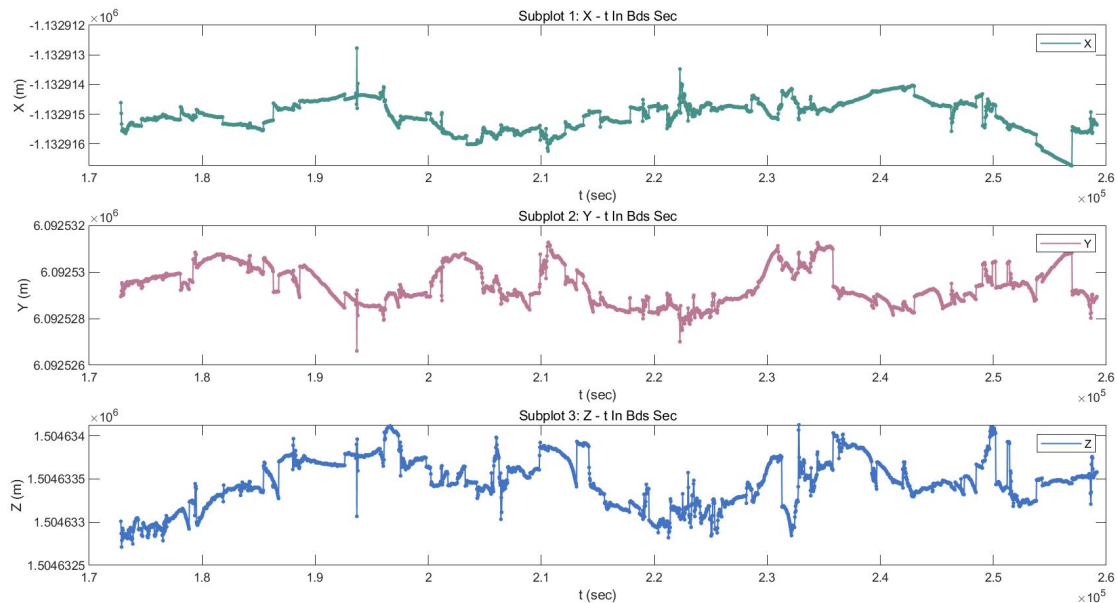


图 3 xyz 坐标时序图

单历元解算的结果由于仅受本历元内观测值的影响，而默认历元间观测值独立。故出现噪声等因素，历元间的实际存在一定相关性的影响，到图像上表现出非平滑的特性。

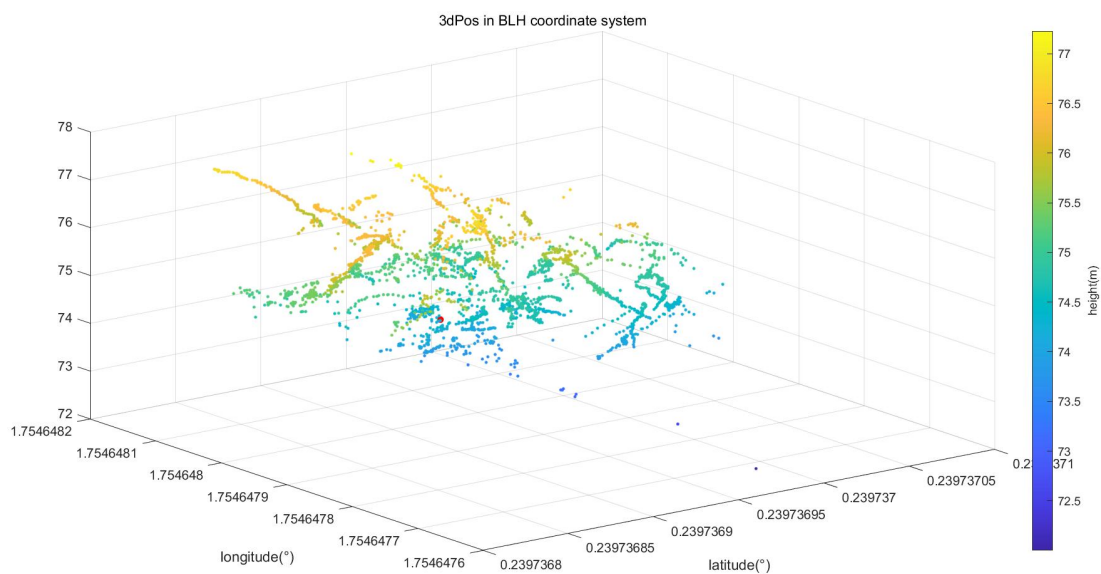


图 4 b1h3d 坐标时序图

中心红色原点为接收机坐标真值的 b1h 坐标位置。三维图中可以直观地看出解算的估计值坐标散落在真值点附近；散点的颜色对应 colorbar 的高程标识（大地高），更直观（从 z 轴坐标也可看出）。

5.2 静态卡尔曼滤波

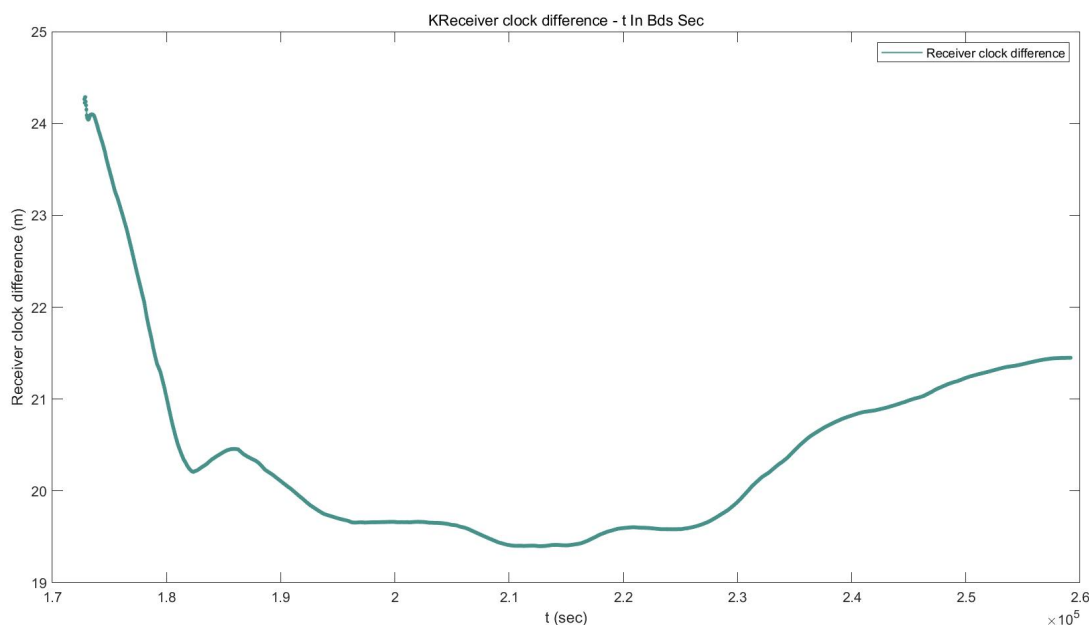


图 1 钟差时序图

我们在逻辑上如果这样想：如果我们可以类比于单历元最小二乘估计为以观测值数量为一大步的解算过程的记录，那么因而可以类比基于最小二乘的静态卡尔曼滤波是类比于一步步解算记录的累积过程。

由于累积的缘故，随累积的增加，静态卡尔曼滤波的自由度在逐步增加积累，而解算的必要观测值只有 4 个。也就是说，随时间积累受新增加的观测值的影响越小，曲线成更弱小的变化。因而，卡尔曼滤波的曲线是较为很平滑的，是积累的平滑过程。

此外，在数值与变化趋势来看，考虑到累积特性，静态卡尔曼滤波的钟差时序图与单历元最小二乘估计的钟差时序图是总体上对应的，因为我们的解算观测数据是相同的。

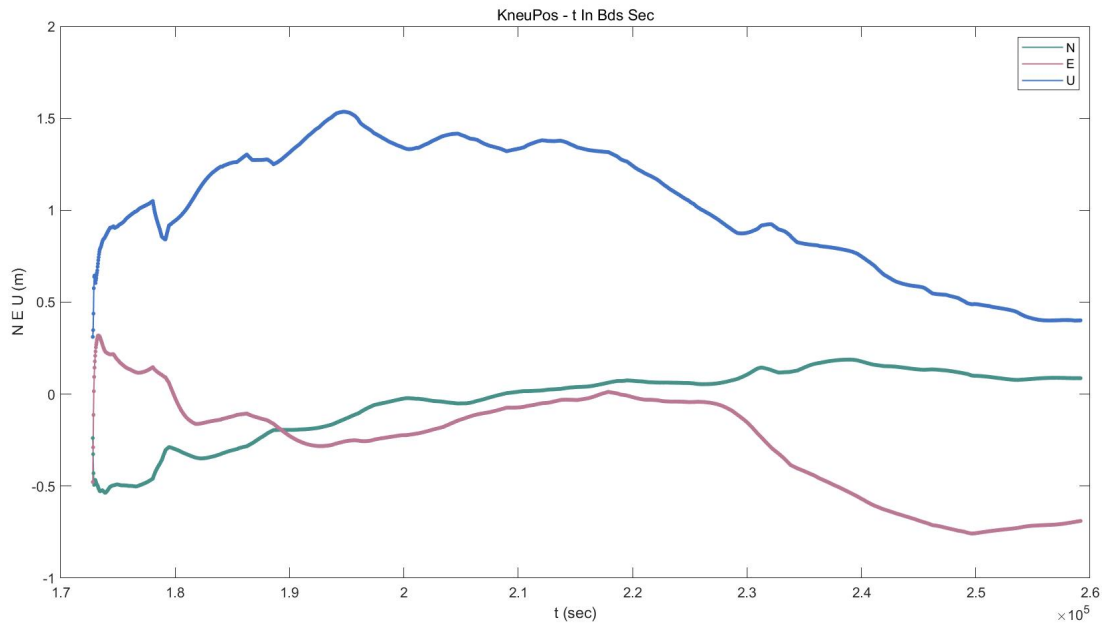


图 2 neu 坐标时序图

同样地，依然可以看出不论是卡尔曼滤波还是单历元最小二乘解算估计，天顶方向 U 的误差仍然是最大的，原因分析是相同的，故不再赘述。

由于作业中体现的是特殊的静态 kalman 滤波，我试着另外在本作业以外考虑 kalman 滤波平滑曲线的原理。

具有用新的观测值更新修正上一步的预测值，同时作为与下一步的预测值，通过考虑这两类的信息（预测值、观测值）的方差予以设权重（方差小权重大，反之同理），体现在 kalman 增益矩阵中，进而体现在对预测值的修正。这样的随时间的递推过程使得滤波后的观测值曲线平滑特性是明显的。

另外，通过不断迭代预测和更新过程，卡尔曼滤波能够有效地减小估计值的方差，使得在下次的滤波中体现出方差的影响；同时 kalman 滤波对噪声和不确定性的处理也能够降低噪声对估计值的影响，从而达到进一步平滑曲线的效果。

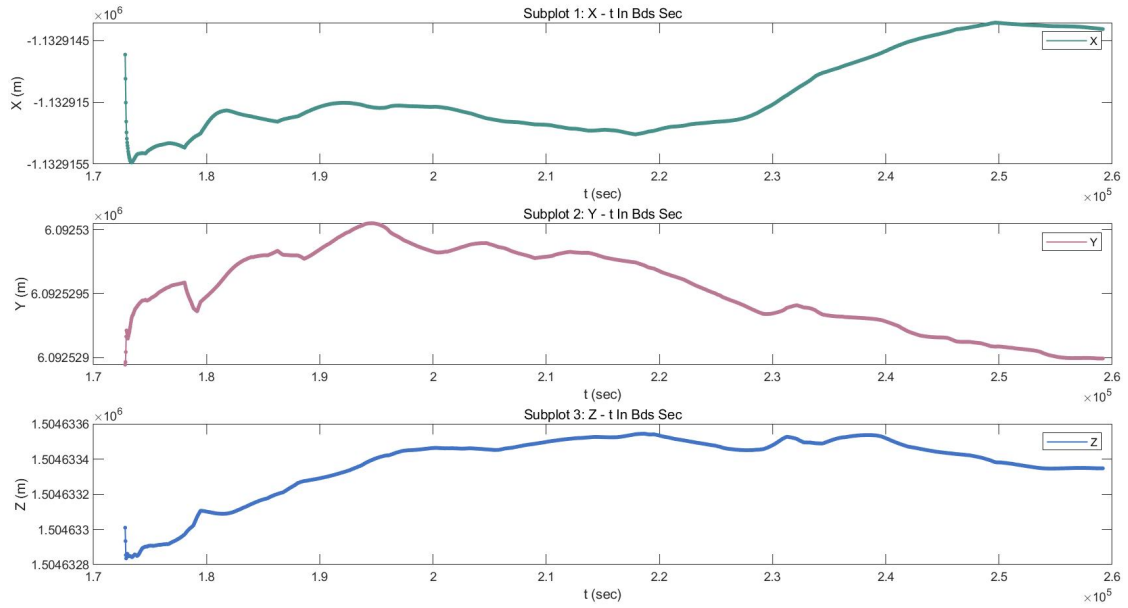


图 3 xyz 坐标时序图

同样是平滑的曲线，对比真值我们可以发现 Y、Z 方向的曲线变化趋势吻合与给定接收机坐标真值的接近，而 X 方向相对背离。但误差均在 1m 以内，也是合理的。

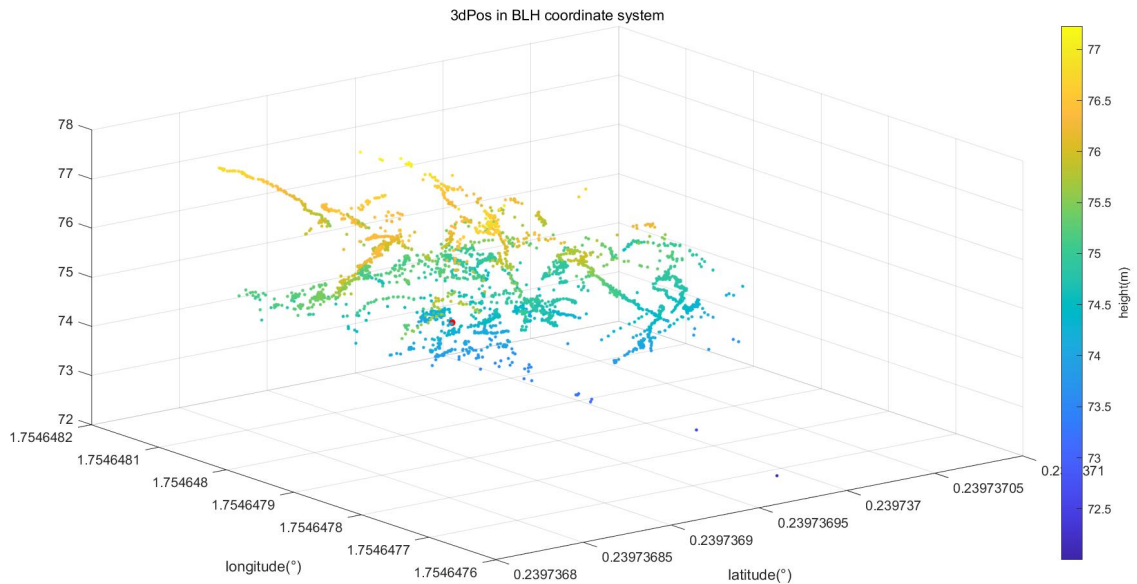


图 4 b1h3d 坐标时序图

同样地，红色圆点代表接收机真值的 b1h（WGS84 系）坐标，旁边的散点代表解算值的分布，与单历元解算对比二者有明显相似性。



5.3 精度评定

首先需要注意的一个点，我们单点定位解算出的坐标是地心地固系，而北斗卫星解算（由于星历播发时卫星位置便以 CGCS2000 作为参考）是 CGCS2000 坐标系下的，最终我们与接收机位置参考真值比较也应该是 CGCS2000 坐标系。但实习报告中给出的参考点是这样的，如下图所示：

The table below contains the ITRF2014 coordinates of the selected stations propagated at epoch 2021/08/10.
Contact : itrfr@ign.fr

DOMES	ID	PT	X [m]	Y [m]	Z [m]	sigma_X [m]	sigma_Y [m]	sigma_Z [m]
21904S001	CUSV	A	-1.13291501648681e+06	6.09252850388968e+06	1.50463316777129e+06	9.93333e-04	1.35206e-03	8.56407e-04

可以看到，接收机位置参考真值是在 ITRF2014 框架下的，使用的是 WGS84 椭球，如果不加以转换，会增大部分误差。我们来看一下两种参考系的参数差异：（我的 xyz2blh.m 函数中）

```
编辑器 - G:\大二下\最优估计\2021302141134-刘恒祯-北斗单点
+1 histogram6.m x pos3d.m x neuDraw.m x h
10 %WGS84
11 a=6378137;
12 e2=0.0066943799013; %第一偏心率的平方
13
14 %CGCS2000
15 %a=6378137;
16 %e2=0.00669438002290;
17
```

可以看到，在椭球的差异很小很小（根本上是短半轴长度的差异，很小很小，但我没有写，半长轴几乎是相等的，因而第一偏心率有微小差异），实际上，将接收机位置真值由 WGS84 系转为 CGCS2000 系来进行精度评定是最合理的。由于老师没有另给 WGS84 与 CGCS2000 坐标系的转化关系，便默认认为二者的差异可以忽略。

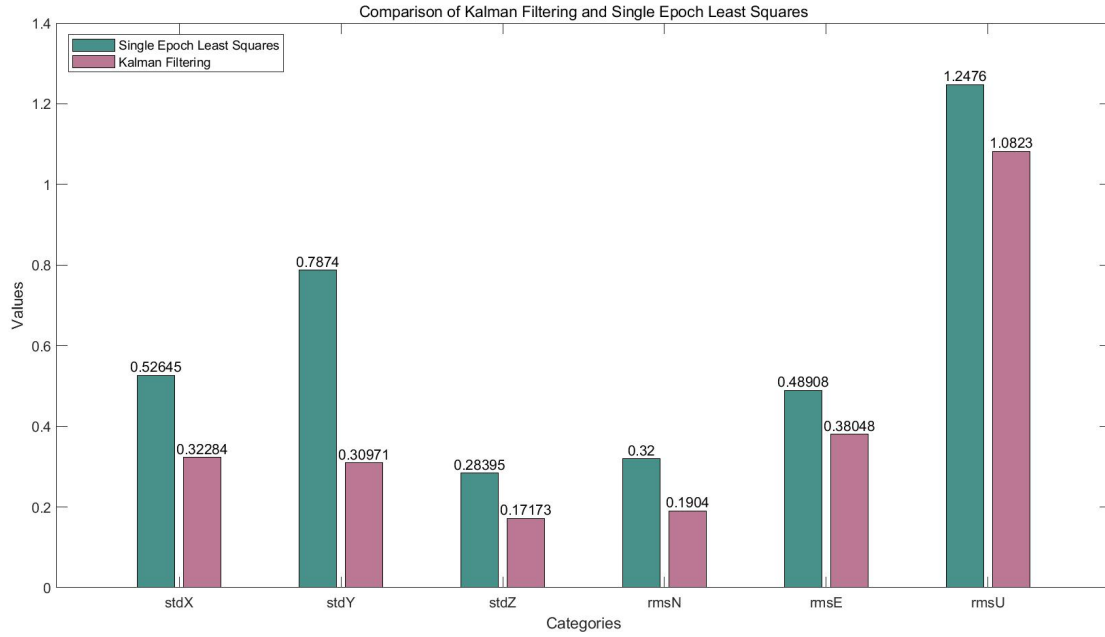


图 1 单历元最小二乘估计与静态卡尔曼滤波精度对比（一）

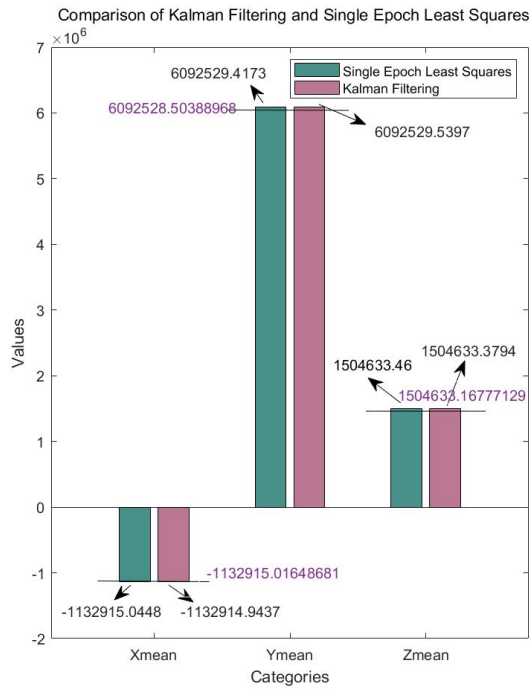


图 2 LS/KF 精度对比（二）

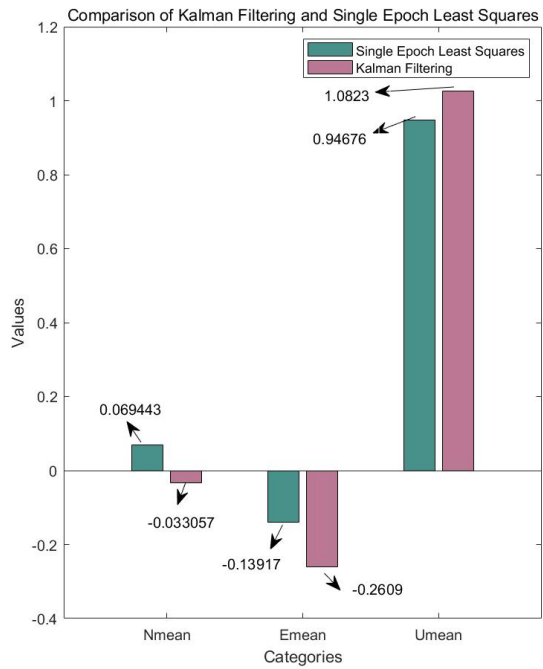


图 3 LS/KF 精度对比（三）

表 1 LS/KF 在 XYZ 坐标的精度比较

	meanX (m)	meanY (m)	meanZ (m)	StdX (m)	StdY (m)	StdZ (m)
LS	-1132915.0448	6092529.4173	1504633.46	0.52645	0.7874	0.28395
KF	-1132914.9437	6092529.5397	1504633.3794	0.32284	0.30971	0.17173



表 2 LS/KF 在 NEU 坐标的精度比较

	meanN (m)	meanE (m)	meanU (m)	rmsN (m)	rmsE (m)	rmsU (m)
LS	0.069443	-0.13917	0.94676	0.32	0.48908	1.2476
KF	-0.033057	-0.2609	1.0265	0.1904	0.38048	1.0823

首先，说明一下我的数据呈现方式。3 个柱状图与两个表格反映了单历元最小二乘估计解算与静态逐次滤波解算的精度对比。柱状图在相对大小的视觉上更直观，而表格在数值上更直观，各有优势。二者相互配合，利于我们的精度评定。

其次，观察 std 与 rms 数据。从 std 与 rms 的对比下我们可以明显发现，可以发现静态逐次滤波（静态卡尔曼滤波）解算估计值的稳定性明显优于单历元最小二乘解算，即所有的 stdX/stdY/stdZ 与 rmsN/rmsE/rmsU，静态逐次滤波的结果都比单历元解算小，结果更好。这正体现出 kalman 滤波优化方差（减小）的平滑特性，等价考虑了所有历元的观测数据，使解算值在概率上更可靠。

再者，观察 mean 数据与真值的对比（柱状图中紫色数据为接收机坐标真值，以及表格中数据的直接比对作差）。发现 X 方向上结果 LS 可以达到 cm 级精度（2.5cm），KF 也可以达到 cm 级（7.0cm），LS 比 KF 略优；在 Y 方向上 LS 与 KF 均表现出误差约为 1m 的精度，二者不相上下；在 Z 方向上 LS 表现出 0.3m 的误差，KF 表现出 0.2m 的误差，KF 优于 LS。这样看来，二种方法的解算可靠性不相上下。

然后，同样地观察 NEU 方向 LS 与 KF 的对比。由于 NEU 值的绝对值大小正相关于解算估计值于真值的差距，绝对值越大精度越差。我们可以看到 KF 的误差要大于 LS，这一方面也是因为 KF 如果在解算

初期观测值不理想需要花费大量的时间收敛平滑一点点拉回趋向真值，过程中缓慢平滑的变化（相对于单历元解算）导致参与计算的很大一连续部分的解算点未与真值逼近，其累积影响则体现在了 NEU 的均值上，使得它相对偏大（要注意的是，可能在某方向上观测值不理想，另一些方向误差就较小，比如 N 方向的均值绝对值 KF 的便明显小于 LS 的）。因此，KF 方法不是坏的，依然起到很有效的解算作用。

另外，分析解算全部数据 txt 文件结果。在此，为了便于展示解算精度，绘制了两种算法在同一时间轴的验后单位权中误差 \hat{m}_0 的时序图，如下所示。

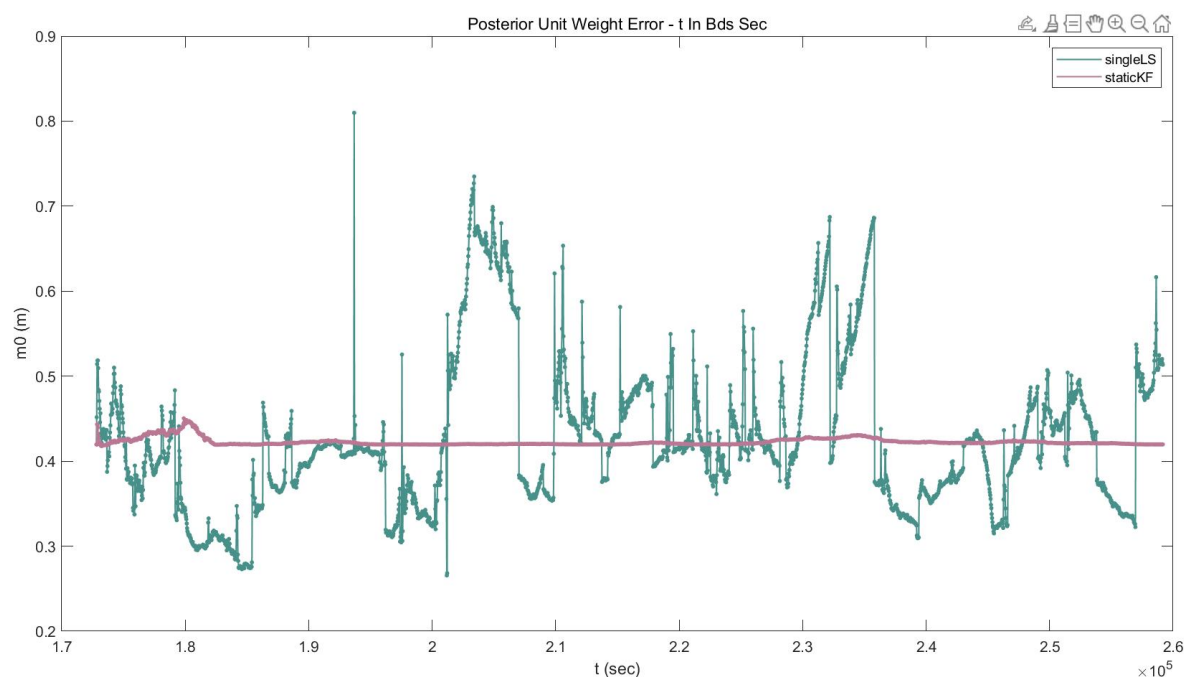


图 4 LS/KF 验后单位权中误差历元时序图

结合图 4 与 txt 文件数据（部分可见在（4.2）单点定位结算结果中部分历元的解算数据），我们可以看到：

在静态卡尔曼滤波解算的“Koutpos.txt”中的验后单位权中误



差是稳定在 0.42m 左右小范围波动。实际上，kalman 滤波的结果文件中，验后单位权中误差是在由初始历元小范围内微弱增大，然后递减收敛，后保持平稳，这符合 kalman 滤波解算方差收敛的特点。另外，对比解算的坐标与给定接收机坐标真值可知，解算误差也在 dm 级，精度是合理的，足够的，优秀的。

在单历元最小二乘解算的结果文件“outpos.txt”中，验后单位权中误差在 0.3-0.7m 之间抖动，总体上在均值 0.4 附近。这也能够反映出观测方程（观测值）较好，单历元最小二乘解算的数据较稳定，不含粗差以及明显的系统误差等，而主要影响解算精度的就是各种随机噪声。以及，比较坐标真值，单历元 LS 解算的结果精度也是相当好的（误差主要在 dm 级，cm 级的也相当一部分）。



6. 实习感想

完成这次实习，我收获蛮多。其实写代码画的时间与优化可视化结果、撰写报告的时间相比，是明显更少的。

讲到代码，代码不难，但必须弄清唯一的是在 kalman 滤波中一直迭代更新的是改正数而非改正数加初值（身边有的同学来问我，他们把逐次的初值都更新了，跳动特别大，输出的验后单位权中误差不收敛）。迭代的流程是清晰的，迭代最大次数与改正数最小阈值两个条件之一不满足便跳出迭代。

在原理上 kalman 滤波也要求在逐次滤波的每一次中也要判断迭代，我是写了这一部分的。但结果发现 kalman 滤波的改正数相对比较大，大概在 $1e-5$ 这个量级上，算一次就够了。我调了迭代次数，也确实发现每个“逐次”的单次内多次迭代没有起到收敛的作用（与给的观测值有关，不是算法问题），干脆为了执行效率迭代一次了。也就相当于只做递推的最小二乘（逐次滤波），单次内不迭代，算一次。可视化的结果及分析也证明了我们的结果是优秀的合理的可信的。

然后是可视化输出的优化，我一直觉得画图就要画好看，于是就画了柱状图、散点-折线图、3d 散点图（可以拟合曲面预测测站周围变化趋势，但对于此次作业没有必要）……以及研究好看的配色。做成之后是有很大的成就感的。

我觉得，写报告是对实习任务的总结与汇报，是知识体系的汇总与思路的总体回顾与梳理。虽然时间长，既是自己汗水的证明，也是交给辛苦审阅的助教与老师的答卷。总之，愿再接再厉，不忘初心！



7. 附录

①main.m

```
clc;clear;close all;%designed by hzLiu, Apr.20th,2023

MAXOBS=60;%PRN<=60
% X(m),Y(m),Z(m),pseudorange(m)
obsdata=zeros([MAXOBS,5]);
satid=nan([MAXOBS,1]);
X=zeros([1,4]);
nepoch=0;
c=299792458;

m0sum=[];
KXsum=[];
Xsum=[];
m0vec=[];
Km0vec=[];
neuPos=[];
KneuPos=[];
clock=[];
Kclock=[];
timeSeq=[];
kfQx0=[];
kfX0=zeros(1,4);
kfx=zeros(4,1);
kfm0=[];

filename='../txtDir/CUSV_20212220_BDS_sgnac_no_eph_error.txt';
outfile='../txtDir/outpos.txt';
outfile1='../txtDir/Koutpos.txt';
fid=fopen(filename,'r');
fop=fopen(outfile,'w+');
fop1=fopen(outfile1,'w+');

disp(filename);
if(fid==-1 || fop==-1)
    disp('can not find or open the file: %')
    'wrong';
else
    %% read file
    while ~feof(fid)
        nepoch=nepoch+1;
```




```
tline=fgetl(fid);
firstline=tline;%copy the first line
if(strcmp(tline(1),'#'))
    %epoch, gpsweek, gpssecond, number of obs
    obshead=sscanf(tline(2:end),"%f");
    for i=1:obshead(4)
        tline=fgetl(fid);
        % satid=tline(1:3); %omit prn
        % X(m),Y(m),Z(m),pseudorange(m),variance(m)
        obsdata(i,:)=sscanf(tline(4:end),"%f",[1,5]);
    end
end
timeSeq=[timeSeq;obshead(3)];

%% calculate
%Static Kalman Filter
if nepoch>=2
    if nepoch==2
        X_est=kfX0.';
        Qxk=kfQx0;
        x_est=kfx0;
        m0k=kfm0;
        allsatsum=kfsatnum;
        Km0vec=[Km0vec;m0k];
    end
    m=0;
    kvecbx=zeros(obshead(4),1);
    kvecby=zeros(obshead(4),1);
    kvecbz=zeros(obshead(4),1);
    kvecbc=-1.*ones(obshead(4),1);
    z=zeros(obshead(4),1);
    ks0=zeros(obshead(4),1);
    %default:sigma0=1,then Q=D
    q=zeros(obshead(4));
    kmaxCorr=1;
    for i=1:obshead(4)
        Xs=obsdata(i,1);
        Ys=obsdata(i,2);
        Zs=obsdata(i,3);

        ks0(i,1)=sqrt((Xs-X_est(1))*(Xs-X_est(1))+(Ys-X_est(2))*(Ys-X_est(2))+(
        Zs-X_est(3))*(Zs-X_est(3)));
        kvecbx(i,1)=-(Xs-X_est(1))/ks0(i,1);
        kvecby(i,1)=-(Ys-X_est(2))/ks0(i,1);
```



```
kvecbz(i,1)=-(Zs-X_est(3))/ks0(i,1);
z(i,1)=obsdata(i,4)-ks0(i,1)+X_est(4);
q(i,i)=obsdata(i,5);
end
w=inv(q);
allsatsum=allsatsum+obshead(4);
%iteration: one time is enough
while abs(kmaxCorr)>1e-9&&m<1
    h=[kvecbx kvecby kvecbz kvecbc];
    deltaz=z-h*x_est;
    %deltaz=z-h*X_est;%wrong
    K=Qxk*h.'/(inv(w)+h*Qxk*h.');
    xk1=x_est+K*deltaz;
    %Xk1=X_est+K*deltaz;%wrong
    Xk1=X_est+xk1;
    Qk1=Qxk-K*h*Qxk;
    vk1=h*xk1-z;

    m0k1=sqrt(1/(allsatsum-4)*(m0k*m0k*(allsatsum-obshead(4)-4)+deltaz.'*K.
    '*Qxk*K*deltaz+vk1.'*w*vk1));
    Dk1=m0k1*m0k1*Qk1;
    %renewal
    x_est=xk1;
    %X_est=Xk1;%no need for the estimated value renewal
    Qxk=Qk1;
    kmaxCorr=max(abs(K*deltaz));
    m=m+1;
end
disp("KalmanEpoch "+string(nepoch)+" Iteration Times:
"+string(m));
KXsum=[KXsum;Xk1.'];
Km0vec=[Km0vec;m0k1];
KXmean=mean(KXsum(:,1));
KYmean=mean(KXsum(:,2));
KZmean=mean(KXsum(:,3));
%StaPos=[Xmean Ymean Zmean];
KStaPos=[-1132915.01648681 6092528.50388968 1504633.16777129];
KstdX=std(KXsum(:,1));
KstdY=std(KXsum(:,2));
KstdZ=std(KXsum(:,3));

outdata1=[Xk1(1),Xk1(2),Xk1(3),-1.*Xk1(4)];
fprintf(fop1,string(firstline)+'\r\n');
```



```

fprintf(fop1,'%14s %14s %14s %14s\r\n','X(m)','Y(m)','Z(m)','T(m)');
    fprintf(fop1,'%14.4f %14.4f %14.4f %14.4f\r\n',outdata1);
    fprintf(fop1,"Posterior Unit Weight
Error(m):%13.8f\r\n",m0k1);
    fprintf(fop1,"Posterior Estimated Variance(m^2):\r\n");
    fprintf(fop1,'%14.4f %14.4f %14.4f %14.4f\r\n',Dk1(1,:));
    fprintf(fop1,'%14.4f %14.4f %14.4f %14.4f\r\n',Dk1(2,:));
    fprintf(fop1,'%14.4f %14.4f %14.4f %14.4f\r\n',Dk1(3,:));
    fprintf(fop1,'%14.4f %14.4f %14.4f %14.4f\r\n',Dk1(4,:));
end

%Single Epoch Cauculation by Least Squares
vecbx=zeros(obshead(4),1);
vecby=zeros(obshead(4),1);
vecbz=zeros(obshead(4),1);
vecbc=-1.*ones(obshead(4),1);
l=zeros(obshead(4),1);
s0=zeros(obshead(4),1);
%default:sigma0=1,then Q=D
Q=zeros(obshead(4));
x=zeros(4,1);
n=0;
maxCorr=1;
while abs(maxCorr)>1e-9&&n<10
    for i=1:obshead(4)
        Xs=obsdata(i,1);
        Ys=obsdata(i,2);
        Zs=obsdata(i,3);

s0(i,1)=sqrt((Xs-X(1))*(Xs-X(1))+(Ys-X(2))*(Ys-X(2))+(Zs-X(3))*(Zs-X(3))
);
        vecbx(i,1)=-(Xs-X(1))/s0(i,1);
        vecby(i,1)=-(Ys-X(2))/s0(i,1);
        vecbz(i,1)=-(Zs-X(3))/s0(i,1);
        l(i,1)=obsdata(i,4)-s0(i,1)+X(4);
        Q(i,i)=obsdata(i,5);
    end
    B=[vecbx vecby vecbz vecbc];
    %P=inv(Q);
    x=((B. ')/Q*B)\((B. ')/Q*1);
    X(1)=X(1)+x(1);
    X(2)=X(2)+x(2);
    X(3)=X(3)+x(3);
    X(4)=X(4)+x(4);
end

```



```
        maxCorr=max(x(1:3));
        n=n+1;
    end
    V=B*x-1;
    m0=sqrt((V.' /Q*V)/(obshead(4)-4));
    Qxx=inv(B.' /Q*B);
    Dxx=m0*m0.*Qxx;
    Xsum=[Xsum;X];
    m0vec=[m0vec;m0];

    Xmean=mean(Xsum(:,1));
    Ymean=mean(Xsum(:,2));
    Zmean=mean(Xsum(:,3));
    %StaPos=[Xmean Ymean Zmean];
    StaPos=[ -1132915.01648681  6092528.50388968  1504633.16777129];
    stdX=std(Xsum(:,1));
    stdY=std(Xsum(:,2));
    stdZ=std(Xsum(:,3));

    if nepoch==1
        kfX0=X;
        kfQx0=Qxx;
        kfx0=x;
        kfm0=m0;
        kfDxx=(m0*m0).*kfQx0;
        kfsatnum=obshead(4);
        allsatsum=kfsatnum;
        outdata1=[X(1),X(2),X(3),-1.*X(4)];
        KXsum=[KXsum;X];

        fprintf(fop1,string(firstline)+'\r\n');

        fprintf(fop1,'%14s %14s %14s %14s\r\n','X(m)','Y(m)','Z(m)','T(m)');
        fprintf(fop1,'%14.4f %14.4f %14.4f %14.4f\r\n',outdata1);
        fprintf(fop1,"Posterior Unit Weight
Error(m):%13.8f\r\n",kfm0);
        fprintf(fop1,"Posterior Estimated Variance(m^2):\r\n");
        fprintf(fop1,'%14.4f %14.4f %14.4f %14.4f\r\n',kfDxx(1,:));
        fprintf(fop1,'%14.4f %14.4f %14.4f %14.4f\r\n',kfDxx(2,:));
        fprintf(fop1,'%14.4f %14.4f %14.4f %14.4f\r\n',kfDxx(3,:));
        fprintf(fop1,'%14.4f %14.4f %14.4f %14.4f\r\n',kfDxx(4,:));
    end
    disp("Epoch "+string(nepoch)+" Iteration Times: "+string(n));
    outdata=[X(1),X(2),X(3),X(4)];
```



```
fprintf(fop,string(firstline)+'\r\n');

fprintf(fop,'%14s %14s %14s %14s\r\n','X(m)','Y(m)','Z(m)','T(m)');
fprintf(fop,'%14.4f %14.4f %14.4f %14.4f\r\n',outdata);
fprintf(fop,"Posterior Unit Weight Error(m):%13.8f\r\n",m0);
fprintf(fop,"Posterior Estimated Variance(m^2):\r\n");
fprintf(fop,'%14.4f %14.4f %14.4f %14.4f\r\n',Dxx(1,:));
fprintf(fop,'%14.4f %14.4f %14.4f %14.4f\r\n',Dxx(2,:));
fprintf(fop,'%14.4f %14.4f %14.4f %14.4f\r\n',Dxx(3,:));
fprintf(fop,'%14.4f %14.4f %14.4f %14.4f\r\n',Dxx(4,:));

%% Clear the data and prepare the next epoch
obsdata = obsdata * 0;

end

%% draw imgs
Xsum(:,4)=-1.*Xsum(:,4);
for i=1:length(Xsum)
    neuPos=[neuPos;xyz2neu(StaPos,Xsum(i,1:3))];
    clock=[clock;Xsum(i,4)];
end
KXsum(:,4)=-1.*KXsum(:,4);
for i=1:length(KXsum)
    KneuPos=[KneuPos;xyz2neu(KStaPos,KXsum(i,1:3))];
    Kclock=[Kclock;KXsum(i,4)];
end

Nmean=mean(neuPos(:,1));
Emean=mean(neuPos(:,2));
Umean=mean(neuPos(:,3));
rmsN=rms(neuPos(:,1));
rmsE=rms(neuPos(:,2));
rmsU=rms(neuPos(:,3));
neuPos=[timeSeq neuPos];
xyzPos=[timeSeq Xsum];
clock=[timeSeq clock];
m0sum=[timeSeq m0vec Km0vec];

neuName="neuPos";
neuDraw(neuPos,neuName);
clockName="Receiver clock difference";
RclockdiffDraw(clock,clockName);
```



```
m0name="Posterior Unit Weight Error";
m0Draw(m0sum,m0name);

KNmean=mean(KneuPos(:,1));
KEmean=mean(KneuPos(:,2));
KUmean=mean(KneuPos(:,3));
KrmsN=rms(KneuPos(:,1));
KrmsE=rms(KneuPos(:,2));
KrmsU=rms(KneuPos(:,3));
KneuPos=[timeSeq KneuPos];
KxyzPos=[timeSeq KXsum];
Kclock=[timeSeq Kclock];
KneuName="KneuPos";
neuDraw(KneuPos,KneuName);
KclockName="KReceiver clock difference";
RclockdiffDraw(Kclock,KclockName);

xyzName="xyzPos";
xyzDraw(xyzPos,xyzName);
KxyzName="KxyzPos";
xyzDraw(KxyzPos,KxyzName);

pos3dname="blhPos3d";
blhpos3d(Xsum(:,1),Xsum(:,2),Xsum(:,3),pos3dname,StaPos);
Kpos3dname="KblhPos3d";
blhpos3d(Xsum(:,1),Xsum(:,2),Xsum(:,3),Kpos3dname,KStaPos);

cats6=['stdX';'stdY';'stdZ';'rmsN';'rmsE';'rmsU'];
vals6=[stdX,stdY,stdZ,rmsN,rmsE,rmsU; ...
        KstdX,KstdY,KstdZ,KrmsN,KrmsE,KrmsU];
histogramname6="Hist1_Kalman_SingleLeastSquares";
histogram6(cats6,vals6,histogramname6);

cats3=['Xmean';'Ymean';'Zmean';'Nmean';'Emean';'Umean'];
vals3=[Xmean,Ymean,Zmean,Nmean,Emean,Umean; ...
        KXmean,KYmean,KZmean,KNmean,KEmean,KUmean];
histogramname3="Hist2_Kalman_SingleLeastSquares";
histogram3(cats3,vals3,histogramname3);

%% console display
disp('Xmean='+string(Xmean)+'m');
disp('Ymean='+string(Ymean)+'m');
disp('Zmean='+string(Zmean)+'m');
disp('stdX='+string(stdX)+'m');
```



```
disp('stdY='+string(stdY)+'m');
disp('stdZ='+string(stdZ)+'m');
disp('Nmean='+string(Nmean)+'m');
disp('Emean='+string(Emean)+'m');
disp('Umean='+string(Umean)+'m');
disp('rmsN='+string(rmsN)+'m');
disp('rmsE='+string(rmsE)+'m');
disp('rmsU='+string(rmsU)+'m');

disp('KXmean='+string(KXmean)+'m');
disp('KYmean='+string(KYmean)+'m');
disp('KZmean='+string(KZmean)+'m');
disp('KstdX='+string(KstdX)+'m');
disp('KstdY='+string(KstdY)+'m');
disp('KstdZ='+string(KstdZ)+'m');
disp('KNmean='+string(KNmean)+'m');
disp('KEmean='+string(KEmean)+'m');
disp('KUmean='+string(KUmean)+'m');
disp('KrmsN='+string(KrmsN)+'m');
disp('KrmsE='+string(KrmsE)+'m');
disp('KrmsU='+string(KrmsU)+'m');

disp('process over');
end
fclose(fid); fclose(fop);fclose(fop1);
```



②xyz2blh.m

```
function [B,L,H] = xyz2blh(X,Y,Z)

%function:
% 将坐标由空间直角坐标系转换为地心坐标系
% input:
% X、Y、Z-->空间直角坐标系坐标
% output:
% B、L、H-->地心坐标系坐标

%WGS84
a=6378137;
e2=0.0066943799013;    %第一偏心率的平方

%CGCS2000
a=6378137;
e2=0.00669438002290;

%1.计算大地经度 L
if (X==0) && (Y>0)
    L=90;
elseif X==0 && Y<0
    L=-90;
else
    L = atan2(Y,X);
end

%2.利用迭代法计算大地纬度 B
%1).赋初值
t_B0=Z/sqrt(X^2+Y^2);           %tanB
t_B1=(a*e2*t_B0/sqrt(1+t_B0^2-e2*t_B0^2)+Z)/sqrt(X^2+Y^2);

%2).进行迭代
while abs(t_B1-t_B0)>1e-10
    t_B0=t_B1;
    t_B1=(a*e2*t_B0/sqrt(1+t_B0^2-e2*t_B0^2)+Z)/sqrt(X^2+Y^2);
end
B=atan2(t_B1,1);

%3).计算大地高
N=a/sqrt(1-e2*(sin(B))^2);
H=sqrt(X^2+Y^2)/cos(B)-N;
```




③xyz2neu.m

```
function pos_NEU = xyz2neu(p0_XYZ,p1_XYZ)

%function:
% 将 p1 点坐标由空间直角坐标系转换为站心坐标系(站心坐标系原点为 p)
% input:
% p0_XYZ-->测站点坐标
% p1_XYZ-->空间点坐标
% output:
% N、E、U-->地心坐标系坐标

% 1.构造旋转矩阵 R
x0=p0_XYZ(1); y0=p0_XYZ(2); z0=p0_XYZ(3);
x1=p1_XYZ(1); y1=p1_XYZ(2); z1=p1_XYZ(3);

%调用函数得到纬度 B、经度 L
[B,L,~]=xyz2blh(x0,y0,z0);

R=[-sin(B)*cos(L) -sin(B)*sin(L) cos(B)
    -sin(L)      cos(L)      0
    cos(B)*cos(L) cos(B)*sin(L) sin(B)];

% 2.实现转换
pos_NEU=(R*[x1-x0;y1-y0;z1-z0]).';

end
```



④xyzDraw.m

```
function [] = xyzDraw(xyzPosmat,name)

clf;
set(gcf, 'Position', get(0, 'ScreenSize'))

%% 连续时间不需处理中断
subplot(3,1,1)
plot(xyzPosmat(:,1),xyzPosmat(:,2), 'Color', [0.28 0.57
0.54], LineWidth=1, DisplayName='X'); hold on
scatter(xyzPosmat(:,1),xyzPosmat(:,2),8,[0.28 0.57 0.54], "filled");
ylabel({'X (m)'});
xlabel({'t (sec)'});
%legend('$X$', 'Interpreter', 'latex');
legend('X');
title('Subplot 1: X - t In Bds Sec')

subplot(3,1,2)
plot(xyzPosmat(:,1),xyzPosmat(:,3), 'Color', [0.73 0.47
0.58], LineWidth=1, DisplayName='Y'); hold on
scatter(xyzPosmat(:,1),xyzPosmat(:,3),8,[0.73 0.47 0.58], "filled");
ylabel({'Y (m)'});
xlabel({'t (sec)'});
legend('Y');
title('Subplot 2: Y - t In Bds Sec')

subplot(3,1,3)
plot(xyzPosmat(:,1),xyzPosmat(:,4), 'Color', [0.26 0.45
0.77], LineWidth=1, DisplayName='Z'); hold on
scatter(xyzPosmat(:,1),xyzPosmat(:,4),8,[0.26 0.45 0.77], "filled");
ylabel({'Z (m)'});
xlabel({'t (sec)'});
legend('Z');
title('Subplot 3: Z - t In Bds Sec')

cd ../imgDir\
saveas(gcf, name, 'png');
cd ../codeDir\
hold off
end
```



⑤RclockdiffDraw.m

```
function [] = RclockdiffDraw(clockmat,name)

clf;
set(gcf, 'Position',get(0, 'ScreenSize'))

%% 连续时间不需处理中断
plot(clockmat(:,1),clockmat(:,2), 'Color',[0.28 0.57
0.54],LineWidth=1);hold on
scatter(clockmat(:,1),clockmat(:,2),8,[0.28 0.57 0.54], "filled");hold on

legend('Receiver clock difference');
ylabel({'Receiver clock difference (m)'});
xlabel({'t (sec)'});
title({name+ ' - t In Bds Sec'});

cd ../imgDir\
saveas(gcf, name, 'png');
cd ../codeDir\
hold off
end
```



⑥neuDraw.m

```
function [] = neuDraw(neuPosmat,name)

clf;
set(gcf, 'Position',get(0, 'ScreenSize'))

%% 连续时间观测无需处理中断
plot(neuPosmat(:,1),neuPosmat(:,2), 'Color',[0.28 0.57
0.54],LineWidth=1);hold on
plot(neuPosmat(:,1),neuPosmat(:,3), 'Color',[0.73 0.47 0.58],LineWidth=1);
plot(neuPosmat(:,1),neuPosmat(:,4), 'Color',[0.26 0.45 0.77],LineWidth=1);

scatter(neuPosmat(:,1),neuPosmat(:,2),8,[0.28 0.57 0.54],"filled");hold
on
scatter(neuPosmat(:,1),neuPosmat(:,3),8,[0.73 0.47 0.58],"filled")
scatter(neuPosmat(:,1),neuPosmat(:,4),8,[0.26 0.45 0.77],"filled")

legend('N','E','U');
ylabel({'N E U (m)'});
xlabel({'t (sec)'});
title({name+' - t In Bds Sec'});

cd ../imgDir\
saveas(gcf, name, 'png');
cd ../codeDir\
hold off
end
```



⑦m0Draw.m

```
function [] = m0Draw(m0,name)

clf;
set(gcf, 'Position',get(0, 'ScreenSize'))

%% 连续时间观测无需处理中断
plot(m0(:,1),m0(:,2), 'Color',[0.28 0.57 0.54],LineWidth=1);hold on
plot(m0(:,1),m0(:,3), 'Color',[0.73 0.47 0.58],LineWidth=1);

scatter(m0(:,1),m0(:,2),8,[0.28 0.57 0.54],"filled");hold on
scatter(m0(:,1),m0(:,3),8,[0.73 0.47 0.58],"filled")

legend('singleLS','staticKF');
ylabel({'m0 (m)'});
xlabel({'t (sec)'});
title({name+' - t In Bds Sec'});

cd ../imgDir\
saveas(gcf, name, 'png');
cd ../codeDir\
hold off
end
```



⑧blhpos3d.m

```
function [] = blhpos3d(x,y,z,name,refpos)
%三维散点着色图
clf;

len=length(x);
[bref,lref,href]=xyz2blh(refpos(1),refpos(2),refpos(3));
b=zeros(len,1);
l=zeros(len,1);
h=zeros(len,1);
for i=1:len
    [b(i),l(i),h(i)]=xyz2blh(x(i),y(i),z(i));
end
c=h;%c 表示对 z 轴进行着色

scatter3(b,l,h,50,c, '. ');%50 表示点的大小, c 表示着色情况, '.'表示点的形状
hold on
scatter3(bref,lref,href,300,'red', '. ');

xlabel('latitude(°)')
ylabel('longitude(°)')
title('3dPos in BLH coordinate system')
grid on
h = colorbar;%右侧颜色栏
set(get(h,'label'),'string','height(m)');%给右侧颜色栏命名

cd ..\imgDir\
saveas(gcf, name, 'png');
cd ..\codeDir\
hold off

end
```




⑨histogram3.m

```
function [] = histogram3(ncats,Y,name)
clf;

X1 = categorical({ncats(1,:),ncats(2,:),ncats(3,:)});
X1 = reordercats(X1,{ncats(1,:),ncats(2,:),ncats(3,:)});
X2 = categorical({ncats(4,:),ncats(5,:),ncats(6,:)});
X2 = reordercats(X2,{ncats(4,:),ncats(5,:),ncats(6,:)});

subplot(1,2,1)
b1=bar(X1,Y(:,1:3));hold on;
b1(1).FaceColor=[0.28 0.57 0.54];
b1(2).FaceColor=[0.73 0.47 0.58];
legend('Single Epoch Least Squares','Kalman
Filtering','Orientation','vertical');
xlabel('Categories')
ylabel('Values')
title('Comparison of Kalman Filtering and Single Epoch Least Squares')

subplot(1,2,2)
b2=bar(X2,Y(:,4:6));hold on;
b2(1).FaceColor=[0.28 0.57 0.54];
b2(2).FaceColor=[0.73 0.47 0.58];
legend('Single Epoch Least Squares','Kalman
Filtering','Orientation','vertical');
xlabel('Categories')
ylabel('Values')
title('Comparison of Kalman Filtering and Single Epoch Least Squares')

cd ../imgDir\
saveas(gcf, name, 'png');
cd ../codeDir\
hold off
end
```



⑩ histogram6.m

```
function [] = histogram6(ncats,Y,name)
clf;

X =
categorical({ncats(1,:),ncats(2,:),ncats(3,:),ncats(4,:),ncats(5,:),ncats(6,:)});
X =
reordercats(X,{ncats(1,:),ncats(2,:),ncats(3,:),ncats(4,:),ncats(5,:),ncats(6,:)});

b=bar(X,Y);
b(1).FaceColor=[0.28 0.57 0.54];
xtips1 = b(1).XEndPoints;
ytips1 = b(1).YEndPoints;
labels1 = string(b(1).YData);
text(xtips1,ytips1,labels1,'HorizontalAlignment','center',...
     'VerticalAlignment','bottom');
b(2).FaceColor=[0.73 0.47 0.58];
xtips2 = b(2).XEndPoints;
ytips2 = b(2).YEndPoints;
labels2 = string(b(2).YData);
text(xtips2,ytips2,labels2,'HorizontalAlignment','center',...
     'VerticalAlignment','bottom');
legend('Single Epoch Least Squares','Kalman
Filtering','Orientation','vertical','Location','northwest');

xlabel('Categories')
ylabel('Values')
title('Comparison of Kalman Filtering and Single Epoch Least Squares')
cd ../imgDir\
saveas(gcf, name, 'png');
cd ../codeDir\
hold off
end
```