

GPS 定位技术与应用实验

——GPS 用户位置求解 Matlab 仿真

一、定位原理

GPS 用户对卫星 j 进行伪距测量，产生观测方程：

$$\rho_j = \sqrt{(x_j - x_u)^2 + (y_j - y_u)^2 + (z_j - z_u)^2} + c\tau_u \quad (1)$$

其中 (x_j, y_j, z_j) 表示第 j 颗卫星的位置坐标； (x_u, y_u, z_u) 是用户的位置坐标， τ_u 是用户接收机钟与 GPS 系统时钟的相对误差。设一共观测到 N 颗卫星，则得到方程组：

$$\begin{cases} \rho_1 = \sqrt{(x_1 - x_u)^2 + (y_1 - y_u)^2 + (z_1 - z_u)^2} + c\tau_u \\ \rho_2 = \sqrt{(x_2 - x_u)^2 + (y_2 - y_u)^2 + (z_2 - z_u)^2} + c\tau_u \\ \vdots \\ \rho_N = \sqrt{(x_N - x_u)^2 + (y_N - y_u)^2 + (z_N - z_u)^2} + c\tau_u \end{cases} \quad (2)$$

定位的目的就是计算 (x_u, y_u, z_u) 和 τ_u 。

直接求解上述非线性方程组十分困难。可以采用**牛顿迭代法**这种常用的数值计算方法，其中的关键思想是**线性化**及**最小二乘法**。具体过程如下：

对用户位置进行估计，得到估计位置坐标 (x_0, y_0, z_0) ，用 $(\Delta x, \Delta y, \Delta z)$ 表示估计位置与真实位置的偏移量，即

$$\begin{cases} x_u = x_0 + \Delta x \\ y_u = y_0 + \Delta y \\ z_u = z_0 + \Delta z \end{cases} \quad (3)$$

设 $f(x_u, y_u, z_u) = \sqrt{(x_j - x_u)^2 + (y_j - y_u)^2 + (z_j - z_u)^2}$ ，并把它在 (x_0, y_0, z_0) 做一阶泰勒级数展开得：

$$\begin{aligned} & f(x_u, y_u, z_u) \\ &= f(x_0, y_0, z_0) + \frac{\partial f(x_0, y_0, z_0)}{\partial x_0} \Delta x + \frac{\partial f(x_0, y_0, z_0)}{\partial y_0} \Delta y + \frac{\partial f(x_0, y_0, z_0)}{\partial z_0} \Delta z \\ &= f(x_0, y_0, z_0) - \frac{x_j - x_0}{f(x_0, y_0, z_0)} \Delta x - \frac{y_j - y_0}{f(x_0, y_0, z_0)} \Delta y - \frac{z_j - z_0}{f(x_0, y_0, z_0)} \Delta z \end{aligned} \quad (4)$$

因此，把对卫星 j 的观测方程线性化后得到：

$$\rho_j = f(x_0, y_0, z_0) - \frac{x_j - x_0}{f(x_0, y_0, z_0)} \Delta x - \frac{y_j - y_0}{f(x_0, y_0, z_0)} \Delta y - \frac{z_j - z_0}{f(x_0, y_0, z_0)} \Delta z + c\tau_u \quad (5)$$

令

$$\begin{cases} \Delta\rho_j = f(x_0, y_0, z_0) - \rho_j \\ a_{xj} = \frac{x_j - x_0}{f(x_0, y_0, z_0)} \\ a_{yj} = \frac{y_j - y_0}{f(x_0, y_0, z_0)} \\ a_{zj} = \frac{z_j - z_0}{f(x_0, y_0, z_0)} \end{cases} \quad (6)$$

伪距观测方程变化为:

$$\Delta\rho_j = a_{xj}\Delta x + a_{yj}\Delta y + a_{zj}\Delta z - c\tau_u \quad (7)$$

把方程组(2)中的每个方程线性化, 得到下面的线性方程组:

$$\begin{cases} \Delta\rho_1 = a_{x1}\Delta x + a_{y1}\Delta y + a_{z1}\Delta z - c\tau_u \\ \Delta\rho_2 = a_{x2}\Delta x + a_{y2}\Delta y + a_{z2}\Delta z - c\tau_u \\ \vdots \\ \Delta\rho_N = a_{xN}\Delta x + a_{yN}\Delta y + a_{zN}\Delta z - c\tau_u \end{cases} \quad (8)$$

把(8)写成矩阵形式, 可得:

$$\Delta\mathbf{p} = \mathbf{H}\Delta\mathbf{x} \quad (9)$$

其中

$$\Delta\mathbf{p} = \begin{bmatrix} \Delta\rho_1 \\ \Delta\rho_2 \\ \vdots \\ \Delta\rho_N \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} a_{x1} & a_{y1} & a_{z1} & 1 \\ a_{x2} & a_{y2} & a_{z2} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ a_{xN} & a_{yN} & a_{zN} & 1 \end{bmatrix}, \quad \Delta\mathbf{x} = \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ -c\tau_u \end{bmatrix}。$$

按照上述方法, 求解非线性方程组(2)的问题被转化为求解线性方程组 (见(8)和(9))。

如果只能观察到 4 颗卫星, 即 $N=4$, (8)和(9)是个根据 4 条线性方程求解 4 个未知数的问题, 具有唯一解:

$$\Delta\mathbf{x} = \mathbf{H}^{-1}\Delta\mathbf{p} \quad (10)$$

如果能观察到的卫星数量大于 4, 即 $N>4$, 求解(8)和(9)是个超定方程组 (即方程数量大于未知数的数量), 此时需要使用**最小二乘法**求, 解的形式为:

$$\Delta\mathbf{x} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \Delta\mathbf{p} \quad (11)$$

求出 $(\Delta x, \Delta y, \Delta z)$ 后, 使用(3)便求出用户坐标。

迭代: 因为线性化使用了一阶泰勒级数展开近似, 这种近似只有当估计坐标 (x_0, y_0, z_0) 非常接近真实坐标 (x_u, y_u, z_u) 时才有效。如果 $(\Delta x, \Delta y, \Delta z)$ 太大, 需要用本次计算得出的坐标 (x_u, y_u, z_u) 作为下一次计算的估计坐标 (x_0, y_0, z_0) , 重新迭代上述计算过程, 直到计算得到的 $(\Delta x, \Delta y, \Delta z)$ 的值比较小为止。

二、Matlab 程序代码

下面 Matlab 程序完成利用伪距测量用户位置的 Matlab 仿真计算。

1、主程序

```
SatellitePosition=[17746 17572 7365 1;  
12127 -9774 21091 1;  
13324 -18178 14392 1;  
14000 -13073 19058 1;  
19376 -15756 -7365 1;  
zeros(19, 4)]; %卫星位置  
%卫星位置坐标，每一行数据的前三列分别表示卫星的x、y、z坐标值，第4列数据表示本颗卫星是否可见，1为可见，0为不可见。  
  
UserPosition=[6400 0 0];  
%用户真实位置（注意：定位程序并未用到此参数）  
  
Prange=CalculatePseudoRange(SatellitePosition, UserPosition);  
%函数CalculatePseudoRange用于计算机模拟伪距测量结果，计算结果Prange是一个矢量，其中的不同元素表示对不同卫星的伪距测量结果  
  
[CalUserPosition, OK]=CalculateUserPosition2(SatellitePosition, Prange);  
%调用函数CalculateUserPosition2，进行定位计算，计算CalUserPosition中包含位置计算结果。
```

2、伪距测量模拟函数 CalculatePseudoRange

```
function Prange=CalculatePseudoRange(SatellitePosition,UserPosition) %计算机模拟伪距测量  
  
c=3e5; %光速，单位：km/s;  
DeltaT=1e-4; %钟差为 1e-4 数量级秒，假设卫星钟间时钟一致，DeltaT=Tu-Ts；钟差不宜超过 3e-4，否则不收敛；  
VisSatNum=0;  
  
%首先找出可以观测到的卫星数量  
SatellitePosNew=[];  
for k=1:24  
    if SatellitePosition(k,4)==1  
        VisSatNum=VisSatNum+1;  
        SatellitePosNew=[SatellitePosNew; SatellitePosition(k,1:3)];  
    end  
end
```

```

        end    %if
    end    %for

    Prange=ones(1,VisSatNum);

    %求解用户接收机收到的伪距信息
    for n=1:VisSatNum
        Prange(1,n)=sqrt(          (SatellitePosNew(n,:)-UserPosition)          *
        (SatellitePosNew(n,:)-UserPosition)' + c*DeltaT );
    end

```

3、定位计算函数 CalculateUserPosition2

```

function                                [CalUserPosition,
CalculateOK]=CalculateUserPosition2(SatellitePosition,Prange)
%输入参数:
%卫星位置坐标SatellitePosition, 对每一颗可见卫星的伪距测量结果Prange
%输出参数:
%用户位置坐标: CalUserPosition, 是一个矩阵, 第一行表示最终定位结果, 后面几行显示定位
计算的中间过程结果;
%参数CalculateOK表示用户位置计算是否成功, 1为成功, 0为失败;
%该程序用线性化方法求解四个或多个卫星的伪距、钟差方程, 具体算法见课本
%假设我们接收到4个或者更多伪距后, 有如下方程
%PR = sqrt( (xi-x)^2 + (yi-y)^2 + (zi-z)^2 ) + ct , i=1,2,3,4
%使用最小二乘法求解

c=3e5;          %光速, 单位: km/s;
DeltaT=1e-3;    %钟差为 1e-4 数量级秒, 假设卫星钟间时钟一致, DeltaT=Tu-Ts; 钟差不
宜超过 3e-4, 否则不收敛;
VisSatNum=0;
CalculateOK=1;

%首先找出可以接收到的卫星, 多于4颗继续运算, 否则返回
SatellitePosNew=[];
for k=1:24
    if SatellitePosition(k,4)==1
        VisSatNum=VisSatNum+1;
        SatellitePosNew=[SatellitePosNew; SatellitePosition(k,1:3)];
    end
end
if VisSatNum<4    %不足 4 颗可见卫星
    CalculateOK=0;
    CalUserPosition=[0 0 0];
    return
end

```

```

XYZ0=[0 0 0];           %给用户位置赋初值
CalculateRecord=XYZ0;    %此变量用于保存每一步迭代计算的中间结果
DeltaT0=0;              %时钟差初始值
Wxyz=SatellitePosNew;   %卫星位置坐标
Error=1000;
ComputeTime=0;
while (Error>0.01) && (ComputeTime<1000) %开始迭代运算
    ComputeTime=ComputeTime+1;
    R=ones(1,VisSatNum);
    for n=1:VisSatNum
        R(1,n)=sqrt( (Wxyz(n,:)-XYZ0) * (Wxyz(n,:)-XYZ0)' ) + DeltaT0*c;
    end %for
    DeltaP=R-Prange;
    A=ones(VisSatNum,3);
    for n=1:VisSatNum
        A(n,:)=(Wxyz(n,:)-XYZ0)./R(1,n);
    end
    H=[A ones(VisSatNum,1)];
    DeltaX=inv(H'*H) * H' * DeltaP'; %最小二乘法求卫星位置
    TempDeltaX=DeltaX(1:3,:);
    Error=max(abs(TempDeltaX));
    XYZ0=XYZ0+DeltaX(1:3,:);
    if ComputeTime<10
        CalculateRecord=[CalculateRecord; XYZ0];
    end
    DeltaT0=DeltaX(4,1)/(-c);
end %while
if ComputeTime==1000
    CalUserPosition=[0 0 0];
    CalculateOK=0;
else
    CalUserPosition=[XYZ0; CalculateRecord];
end

```

三、实验内容

- 1、熟悉 Matlab 编程的语法、环境。
- 2、定位程序是一个迭代运算程序，卫星坐标和用户坐标由上述语句给出，运行程序，那么
 - a) 写出最终算出的用户坐标结果（要求精确到小数点后 4 位）；
 - b) 写出程序的迭代计算次数；
 - c) 通过编程，计算出每次迭代计算的中间结果与用户真实位置的距离。（要求：给出具体程序语句。）

提示：CalculateUserPosition2 程序返回的变量 CalUserPosition 包含了每一次迭代计算的中间结果。设用户真实坐标为 (x_u, y_u, z_u) ，迭代中间结果的坐标为 (x_c, y_c, z_c) ，则两者的距离为： $\sqrt{(x_u - x_c)^2 + (y_u - y_c)^2 + (z_u - z_c)^2}$ ，这反映了计算结果的误差。

3、把用户真实坐标设为(你的学号后四位+4000, 0, 0)，保持卫星坐标不变，重复上述的 a)、b)、c)三步，分别给出程序运行结果。

四、思考题

1. 程序中的坐标是在哪个坐标系下的坐标？坐标的单位是什么？
2. 在实验中，卫星的位置坐标是已知量。问：在实际使用中，用户如何获知卫星的坐标信息？
3. 定位计算，除了要知道卫星的坐标，还要知道伪距信息。问：a)什么叫伪距？b)伪距信息如何获得？
4. 程序 CalculateUserPosition2 中，语句“XYZ0=[0 0 0];”给用户坐标附的初始值为(0, 0, 0)。问(0, 0, 0)是否最佳的用户坐标初始值？有更好的用户坐标初始值吗？为什么？
5. 程序 CalculateUserPosition2 中，while 语句的判断条件为“(Error>1) && (ComputeTime<1000)”，即要求两个判断条件同时成立才循环，问：a)判断条件 Error>1 是什么意思？解释其物理含义。b)判断条件 ComputeTime<1000 是什么意思？解释其在程序中的作用。
6. 程序 CalculateUserPosition2 中，语句“A(n,:)=(Wxyz(n,:)-XYZ0)./R(1,n);”是什么意思？请用定位原理中的数学公式来解释。
7. 程序 CalculateUserPosition2 中，语句“XYZ0=XYZ0+DeltaX(1:3,:);”是什么意思？请用定位原理中的内容解释。
8. 指出程序需要改善的地方。

实验报告内容要求

- 1、结合 Matlab 程序叙述 GPS 的定位原理。
- 2、给出实验内容第 2、3 步的结果。
- 3、回答思考题。