第二次大作业 P2：PVT 解算

聂浩 (2013011280)

2016 年 6 月 19 日

1 单系统定位：自己实现一种定位解算算法，分别利用 GPS、北斗的数据文件计算用户的位置与速度。通过查阅地图，看看用户的位置在哪里？

a 单 GPS

中国境内大部分地图采用的为 GCJ02 火星坐标系，而百度地图使用的为 BD09 坐标系，而所提供的 GPS 数据为基于 WGS84 的 ECEF 坐标系。

本次大作业中首先完成的是对于以上坐标系的相互转换[1]以及 ECEF 与 LLH 坐标的转换[2]，具体可见 co_trans[3]中的文件。

随后进行 PVT 解算，由于本作业中没有提供 T 的数据，所以这里仅仅进行 PV 解算，代码如下

gps.m

```
clear;
close;
load L1.mat
[lng,lat]=gcj02towgs84(116.3324885066,40.0014938354);
pre=lla2ecef([lat,lng,0]);
%以清华大学主楼为起始点，坐标来自高德地图

position_ecef=zeros(3,200);
position_wgs84=zeros(2,200);
position_gcj02=zeros(2,200);
position_bd09=zeros(2,200);
for i=1:200
    rho=PseudoRange(:,i);
    sat=SatInfo(:,i);
    position_ecef(:,i)=pvt(rho,sat,pre);
    lla=ecef2lla(position_ecef(:,i)');%转换为wgs84
    lat=lla(1);
    lng=lla(2);
    position_wgs84(:,i)=[lng;lat];
    [lng,lat]=wgs84togcj02(lng,lat);
```

[1]这里参考了相关代码https://github.com/wandergis/coordTransform_py

[2]使用 matlab 2015b 自带函数

[3]执行本作业中的程序需要将 co_trans 文件夹添加到路径中，或者将其中的文件拷出

```
    position_gcj02(:,i)=[lng;lat];
    [lng,lat]=gcj02tobd09(lng,lat);
    position_bd09(:,i)=[lng;lat];
end
v=sum((position_ecef(:,2:200)-position_ecef(:,1:199)).^2).^0.5;
```

pvt.m

```
function [pre]=pvt(rho,sat,pre)
%%%输出 pre为最终得到的坐标
%%%输入 rho为伪距
%%%      sat为卫星坐标
%%%      pre为预测的用户位置

%获得卫星的数量
[n,~]=size(rho);

%将卫星位置从结构体转换为n*3的矩阵
sat=reshape(struct2array(sat),3,n)';

error=100;
%循环条件，当计算修正位置小于20m时停止迭代
count=1;
%循环条件，防止结果不收敛而导致死循环
while(error>20||count>20)
    count=count+1;
    tmp_pre=repmat(pre,n,1);
    %矩阵化，方便计算
    r=(sum((sat-tmp_pre).^2,2)).^0.5;
    delta_rho=rho-r;
    H=-(sat-tmp_pre).*repmat(r.^-1,1,3);
    delta_p=inv(H'*H)*H'*delta_rho;

    error=sum(delta_p.^2)^0.5;
    %error为delta_p的长度
    pre=pre+delta_p';
end
return;
```

计算得到的位置点如图1[4]，速率如图2，单位为 m/s

b 单北斗

根据相关参考文献，“WGS84 坐标与 CGCS2000 坐标的差值在 cm 级范围内”[5]，所以这里直接用 WGS84 坐标系对北斗信号进行结算，相关代码见附录所附 beidou.m。计算得到的位置点如图3，速率如图4，单位为 m/s

[4]该图为调用百度地图 API 绘制，相关代码见 map/gps.html, 请使用 Chrome 浏览器打开

[5]WGS84 和 CGCS2000 坐标转换研究彭小强

图 1: GPS 定位结果

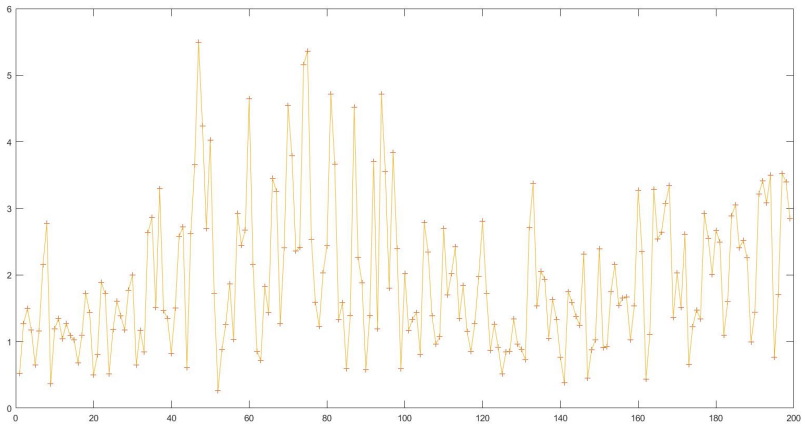

图 2: GPS 定位速率

图 3: 北斗定位结果

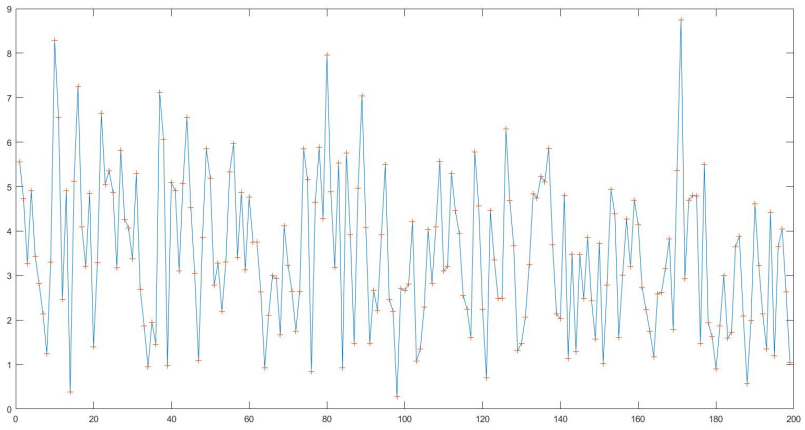

图 4: 北斗定位速率

图 5: 联合定位结果

2 联合定位（多模接收）：自己实现一种定位解算算法，联合使用 GPS、北斗的数据文件计算用户的位置与速度。并将该定位结果与单独使用 GPS、北斗时的定位解算结果进行比较？给出你的结论？

根据上一问的结论可以看出，200 个定位点之间的距离非常小，即整个定位过程中用户几乎没有移动，这里将北斗的 7 颗卫星信息直接附在 GPS 信号之后，即 PVT 解算中以 14 颗星进行解算。代码见附录 merge.m 计算得到的位置点如图5，速率如图6，单位为 m/s，从定位图可以看出，多模接受得到的 200 个定位点更加集中, 可见其精度更高。

3 在导航系统使用中，一般希望收到的卫星数越多越好，即希望采用更多的卫星来进行定位解算，这也是目前国际上开发新卫星导航系统的出发点之一。在此，试改变 GPS、北斗可用卫星数，然后再定位，并比较定位解算结果的变化？给出你的结论？

只是用 GPS 的前五颗卫星数据进行 PVT 解算，相关代码见附录 reduce.m.

所得结果如图7，速率如图8，单位为 m/s，从定位图可以看出，卫星减少后定位的区域变大，可见

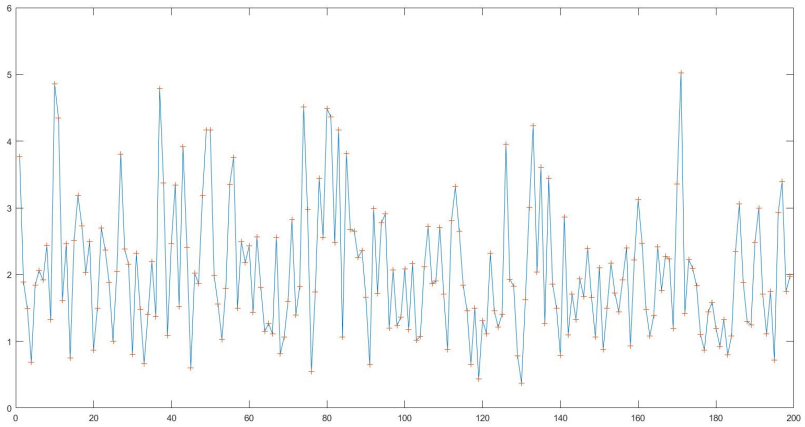

图 6: 联合定位速率

会出现更大的误差。

4 附录

beidou.m

```
clear;
close;
load B1.mat
[lng,lat]=gcj02towgs84(116.3324885066,40.0014938354);
pre=lla2ecef([lat,lng,0]);
%以清华大学主楼为起始点，坐标来自高德地图
position_ecef=zeros(3,200);
position_wgs84=zeros(2,200);
position_gcj02=zeros(2,200);
position_bd09=zeros(2,200);
for i=1:200
    rho=PseudoRange(:,i);
    sat=SatInfo(:,i);
    position_ecef(:,i)=pvt(rho,sat,pre);%PVT解算
    lla=ecef2lla(position_ecef(:,i)');
    lat=lla(1);
    lng=lla(2);
    position_wgs84(:,i)=[lng;lat];
    [lng,lat]=wgs84togcj02(lng,lat);%转换为gcj02
    position_gcj02(:,i)=[lng;lat];
    [lng,lat]=gcj02tobd09(lng,lat);%转换为bd09
    position_bd09(:,i)=[lng;lat];
end
v=sum((position_ecef(:,2:200)-position_ecef(:,1:199)).^2).^0.5;
```

图 7: 减少卫星后 GPS 定位结果

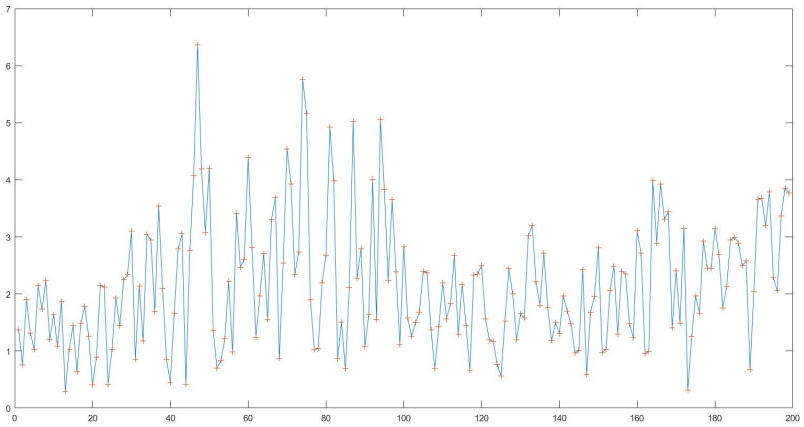

图 8: 减少卫星后 GPS 定位速率

merge.m

```m
clear;
close;
load L1.mat
P=PseudoRange;
sat=SatInfo;
load B1.mat;
PseudoRange=[P;PseudoRange];
SatInfo=[sat;SatInfo];
[lng,lat]=gcj02towgs84(116.3324885066,40.0014938354);
pre=lla2ecef([lat,lng,0]);
%以清华大学主楼为起始点，坐标来自高德地图

position_ecef=zeros(3,200);
position_wgs84=zeros(2,200);
position_gcj02=zeros(2,200);
position_bd09=zeros(2,200);
for i=1:200
    rho=PseudoRange(:,i);
    sat=SatInfo(:,i);
    position_ecef(:,i)=pvt(rho,sat,pre);
    lla=ecef2lla(position_ecef(:,i)');
    lat=lla(1);
    lng=lla(2);
    position_wgs84(:,i)=[lng;lat];
    [lng,lat]=wgs84togcj02(lng,lat);
    position_gcj02(:,i)=[lng;lat];
    [lng,lat]=gcj02tobd09(lng,lat);
    position_bd09(:,i)=[lng;lat];
end
v=sum((position_ecef(:,2:200)-position_ecef(:,1:199)).^2).^0.5;
```

reduce.m

```m
clear;
close;
load L1.mat
%仅仅保留前五个卫星的数据
PseudoRange=PseudoRange(1:5,:);
SatInfo=SatInfo(1:5,:);
[lng,lat]=gcj02towgs84(116.3324885066,40.0014938354);
pre=lla2ecef([lat,lng,0]);
%以清华大学主楼为起始点，坐标来自高德地图

position_ecef=zeros(3,200);
position_wgs84=zeros(2,200);
position_gcj02=zeros(2,200);
position_bd09=zeros(2,200);
for i=1:200
    rho=PseudoRange(:,i);
    sat=SatInfo(:,i);
    position_ecef(:,i)=pvt(rho,sat,pre);
    lla=ecef2lla(position_ecef(:,i)');
    lat=lla(1);
```

```
    lng=lla(2);
    position_wgs84(:,i)=[lng;lat];
    [lng,lat]=wgs84togcj02(lng,lat);
    position_gcj02(:,i)=[lng;lat];
    [lng,lat]=gcj02tobd09(lng,lat);
    position_bd09(:,i)=[lng;lat];
end
v=sum((position_ecef(:,2:200)-position_ecef(:,1:199)).^2).^0.5;
```