

周报

参数设定

将宇航楼楼顶的水平点设为原点构建了ENU坐标系，楼顶的坐标（LLA坐标系）为[30.534 114.361 50]，反射点的坐标为[30.534 114.351 0]。选取2025年1月1日8：00开始，通过这个参数设置：

My Settings	
Time of almanac:	2025-01-01
Time zone:	UTC +08:00
Visible period:	
2025-01-01 08:00 - 2025-01-01 14:00	
Latitude:	N 30° 32' 2.4"
Longitude:	E 114° 21' 3.6"
Height:	50 m
Elevation cutoff:	30 °

找到了符合条件的GPS卫星，并记录他们的六根参数：

名称	中文名称
Eccentricity:	离心率
Semimajor axis:	半长轴
Right ascension of ascending node:	升交点赤经
Argument of perigee:	近地点辐角
Mean anomaly:	平近点角
Inclination:	轨道倾角

计算位置，仍然采用aerospace toolbox。对于这个工具箱，需要在GCRF中的六根参数，接收的六根参数是：

名称	中文名称
Eccentricity:	离心率
Semimajor axis:	半长轴
Right ascension of ascending node:	升交点赤经
Argument of periapsis:	近心点辐角
True anomaly:	真近点角
Inclination:	轨道倾角

因此需要进行转换。（perigee和periapsis指同一量）

但平近点角需要转换为真近点角。设平近点角为 M ,偏近点角为 E , 真近点角为 θ ,首先通过迭代法求偏近点角的数值解：

$$M = E - e \sin E$$

再求真近点角：

$$\theta = \arctan \frac{\sqrt{1-e^2} \sin E}{\cos E - e}$$

其中 e 是离心率。

计算分辨率

计算完成之后，将卫星六根数导入并计算轨道和速度，将其带入分辨率计算函数，即能够获得相应的分辨率数据，通过循环迭代即可获得图像。

代码如下：

```
%% 参数设定
PR0=[30.534 114.361 50];%观测
PRflat=[30.534 114.361 0];%原点
TA=[30.534 114.351 0];%反射点
%GPS卫星六根数，顺序：Eccentricity, Semimajor axis, RAAN, Argument of perigee,
Mean
%anomaly, Inclination
G02=[0.016548 26559593.0 -111.623 -60.930 -42.868 -55.371];
G03=[0.005665 26559593.0 -46.336 65.644 120.500 56.587];
G04=[0.003235 26559593.0 15.315 -169.998 -68.494 55.403];
%时间
starttime=datetime(2025,1,1,8,0,0,"TimeZone","Asia/Shanghai");
durationtime=hours(4);
sampletime=60;% s
```

```

%% 设定和导入卫星
sc = satelliteScenario(starttime,starttime+durationtime,samptime);

%导入卫星
sc=GPSImport(sc,G02);
sc=GPSImport(sc,G03);
sc=GPSImport(sc,G04);

%% 获取位置和预报
[position,velocity]=states(sc,Satellites,CoordinateFrame='ecef');

% 获取地面站 ECEF 坐标
gsECEF = lla2ecef(PRflat);
% 转换卫星坐标为ENU
[xn_pos,ye_pos,zup_pos]=ecef2enu(position(1,:,:),position(2,:,:),position(3,:,:),PRflat(1),PRflat(2),PRflat(3),wgs84Ellipsoid);

[xn_vel,ye_vel,zup_vel]=ecef2enu(velocity(1,:,:),velocity(2,:,:),velocity(3,:,:),PRflat(1),PRflat(2),PRflat(3),wgs84Ellipsoid);

xn_pos=squeeze(xn_pos);
ye_pos=squeeze(ye_pos);
zup_pos=squeeze(zup_pos);

xn_vel=squeeze(xn_vel);
ye_vel=squeeze(ye_vel);
zup_vel=squeeze(zup_vel);

%% 计算分辨率和作图
% 以原点将发射机等转换成ENU坐标系
PR_enu=lla2enu(PR0,PRflat,"flat");
TA_enu=lla2enu(TA,PRflat,"flat");

x=starttime:seconds(samptime):starttime+durationtime;
S_ground=zeros(size(xn_pos,1),3);

for i=1:3
    for j=1:size(xn_pos,1)
        S_ground(j,i)=Calcu_resolution_fix_receiver([xn_pos(j,i); ye_pos(j,i); zup_pos(j,i)], [xn_vel(j,i); ye_vel(j,i); zup_vel(j,i)],TA_enu,PR_enu,[0;30;0]);%缓慢运动
    end
    figure;
    plot(x,S_ground(:,i));
    title(sprintf("卫星%s随时间分辨率变化趋势",sc.Satellites.Name(i)));
    xlabel("时间");

```

```

        ylabel("分辨率");
    end

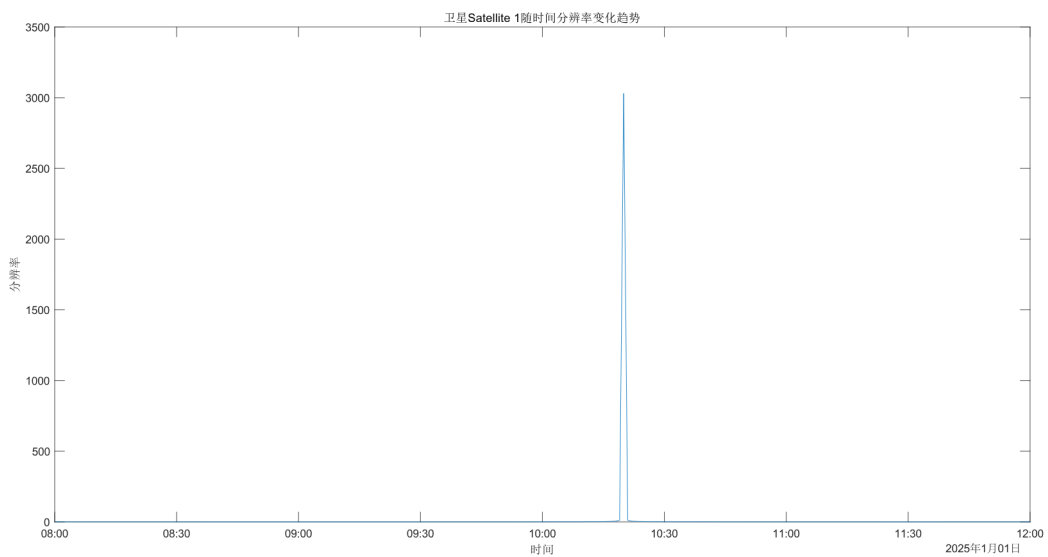
    figure;
    plot(x,sum(S_ground,2));
    title("总分辨率随时间变化趋势");
    xlabel("时间");
    ylabel("分辨率");

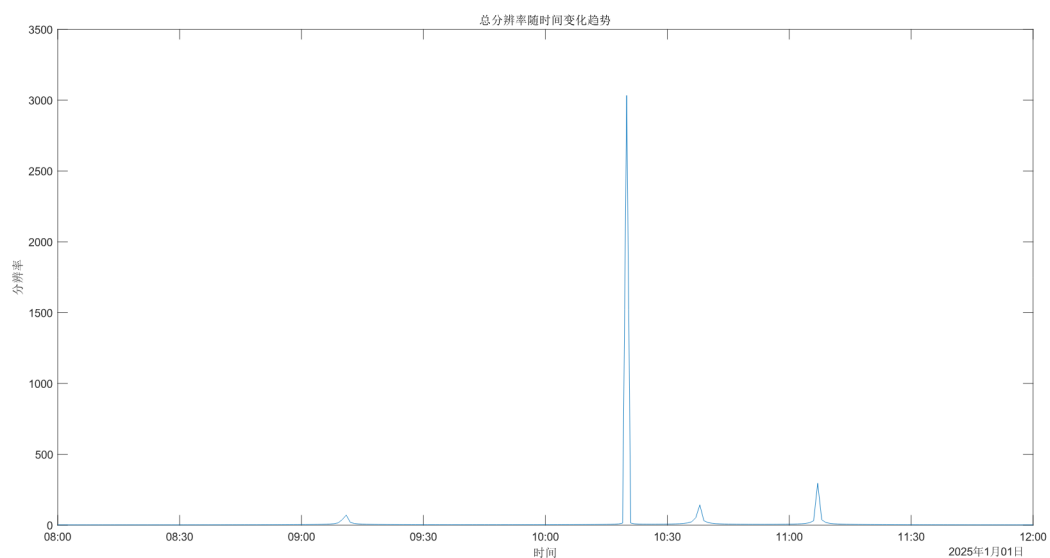
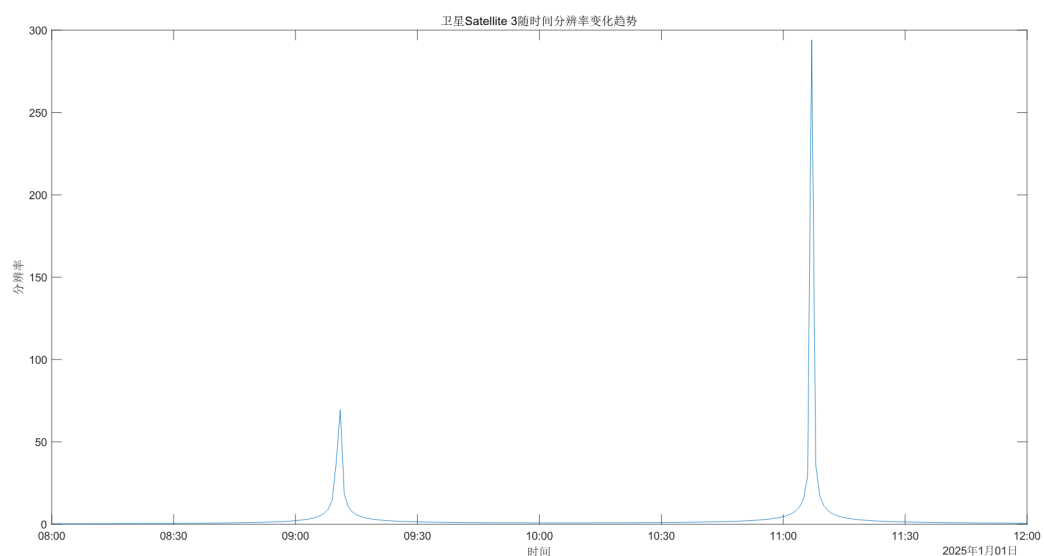
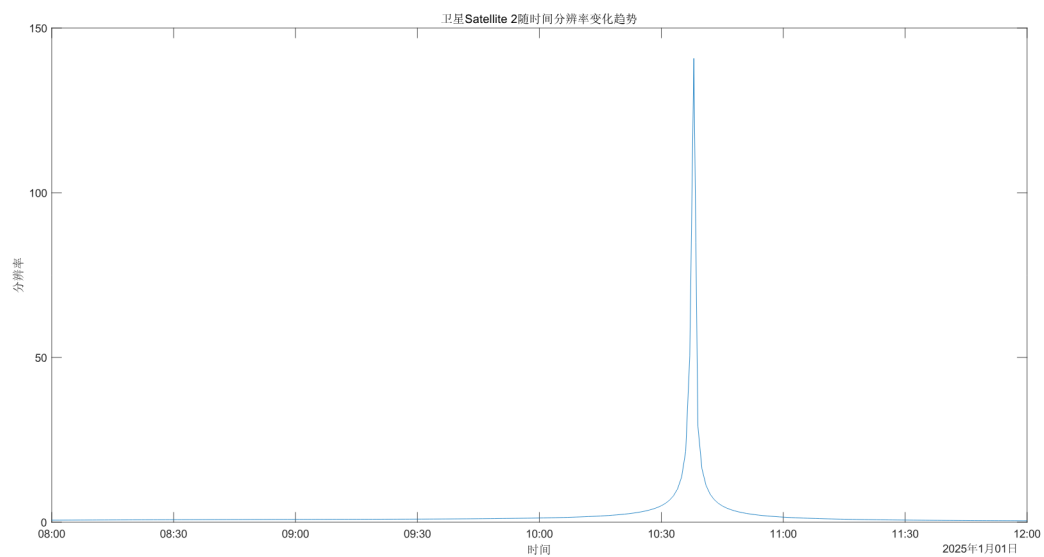
    function sc=GPSImport(sc,orbit)
    %计算真近地点
    e=orbit(1);
    M=orbit(5);
    fun=@(E) E-e*sind(E)-M;
    E=fzero(fun,M);
    theta=atand((sqrt(1-e^2)*sind(E))/(cosd(E)-e));
    satellite(sc,orbit(2),e,orbit(6),orbit(3),orbit(4),theta);

    end

```

得到的图像：





优化问题

在本问题中，假设了水平位置0的点进行测试，则目标优化的变量是ENU坐标系下的500m内的一点 $[x_e \ y_n]$ 和时间 t 共3个变量，目标是让三颗卫星的分辨率和最小。

使用Matlab Optimization Toolbox，首先构建一个外壳调用目标函数：

```
function [S_ground_sum] = Calcu_relsolution_optimization_shell(best_position,
time, PT_list,VT_list,PR,VR)
%CALCU_RELSOLUTION_OPTIMIZATION_SHELL 优化壳
% best_position: 优化量
% time: 优化量
% PT_list: 预测的发射卫星坐标
% VT_list: 预测的发射卫星速度
% 地面反射点的坐标TA、接收机的速度VR
% 输出分辨面积和

TA=[best_position; 0];
S_ground_sum=0;
for i=1:3

    S_ground_sum=S_ground_sum+Calcu_relsolution_fix_receiver(squeeze(PT_list(time,
i,:)),squeeze(VT_list(time,i,:)),TA,PR,VR);
end
end
```

通过优化控件设计一个优化目标，使用遗传算法求解：

优化

problem

solution

 = 在约束下最小化问题目标

创建优化变量

名称	维度	类型	下界	上界	初始点			
best_position	2x1	连续	-Inf	Inf	0	-	+	
time	1x1	整数	1	241	241	-	+	

定义问题

目的

最小化

最大化

可行性

求解方程

目标

函数文件

Calcu_relsolution_optimization_shell...

新建...

函数输入

Calcu_relsolution_optimization_shell (

best_position

 ,

time

 ,

PT_list

 , ...

VT_list

 ,

PR_enu

 ,

VR

)

约束

单行定义

norm([best_position;0]-PR_enu)

<=

500

-

+

指定与问题相关的求解器选项

显示结果

☒ 问题

☒ 解

☒ 求解器停止原因

☒ 目标值

选择任务模式

代码是：

```
PR0=[30.534 114.361 50];%观测
PRflat=[30.534 114.361 0];%原点
TA=[30.534 114.351 0];%反射点
%GPS卫星六根数, 顺序: Eccentricity, Semimajor axis, RAAN, Argument of perigee,
Mean
%anomaly, Inclination
G02=[0.016548 26559593.0 -111.623 -60.930 -42.868 -55.371];
G03=[0.005665 26559593.0 -46.336 65.644 120.500 56.587];
G04=[0.003235 26559593.0 15.315 -169.998 -68.494 55.403];
%时间
starttime=datetime(2025,1,1,8,0,0,'TimeZone','Asia/Shanghai');
durationtime=hours(4);
sampletime=60;% s
sc = satelliteScenario(starttime,starttime+durationtime,sampletime);
%导入卫星
sc=GPSImport(sc,G02);
sc=GPSImport(sc,G03);
sc=GPSImport(sc,G04);
%% 获取位置和预报
[position,velocity]=states(sc,Satellites,CoordinateFrame='ecef');
```

```

% 获取地面站 ECEF 坐标
gsECEF = lla2ecef(PRflat);
% 转换卫星坐标为ENU
[xn_pos,ye_pos,zup_pos]=ecef2enu(position(1,:,:),position(2,:,:),position(3,:,:),PRflat(1),PRflat(2),PRflat(3),wgs84Ellipsoid);

[xn_vel,ye_vel,zup_vel]=ecef2enu(velocity(1,:,:),velocity(2,:,:),velocity(3,:,:),PRflat(1),PRflat(2),PRflat(3),wgs84Ellipsoid);

xn_pos=squeeze(xn_pos);
ye_pos=squeeze(ye_pos);
zup_pos=squeeze(zup_pos);

xn_vel=squeeze(xn_vel);
ye_vel=squeeze(ye_vel);
zup_vel=squeeze(zup_vel);

% 以原点将发射机等转换成ENU坐标系
PR_enu=lla2enu(PR0,PRflat,"flat")';
TA_enu=lla2enu(TA,PRflat,"flat")';

x=starttime:seconds(sampletime):starttime+duriantime;
PT_list=cat(3,xn_pos, ye_pos, zup_pos);
VT_list=cat(3,xn_vel, ye_vel, zup_vel);
VR=[0;30;0];

% 创建优化变量
best_position = optimvar("best_position",2,1);
time = optimvar("time","Type","integer","LowerBound",1,"UpperBound",241);

% 设置求解器的初始起点
initialPoint(best_position) = zeros(size(best_position));
initialPoint(time) = repmat(241,size(time));

% 创建问题
problem = optimproblem;

% 定义问题目标
problem.Objective = fcn2optimexpr(@Calcu_relsolution_optimization_shell,...
    best_position,time,PT_list,VT_list,PR_enu,VR);

% 定义问题约束
problem.Constraints = norm([best_position;0]-PR_enu) ≤ 500;

% 显示问题信息
show(problem);

```



```

% 求解问题
[solution,objectiveValue,reasonSolverStopped] = solve(problem,initialPoint);

% 显示结果
solution
reasonSolverStopped
objectiveValue

% 清除变量
clearvars best_position time initialPoint reasonSolverStopped objectiveValue

timeline=starttime:seconds(sampletime):starttime+durationtime;
disp("最终解: 坐标: ");
disp([solution;best_position;0]);
disp("时间: "+string(timeline(solution;time)));

```

最后解得，在ENU坐标系下，坐标为[10.4486 -0.9424 0]，时间为1月1日早上8时，分辨率达到最小值0.1716.