

Этап 2. Моделирование.

Требуется реализовать на языке Matlab или Python функцию расчета положения спутника Beidou на заданный момент по шкале времени UTC. В качестве эфемерид использовать данные, полученные на предыдущем этапе.

Построить трехмерные графики множества положений спутника Beidou с системным номером. Графики в двух вариантах: в СК ECEF WGS84 и соответствующей ей инерциальной СК. Положения должны соответствовать временному интервалу с 18:00 МСК 16 февраля до 06:00 МСК 17 февраля 2021 года. Допускается использовать одни и те же эфемериды на весь рассматриваемый интервал.

Построить SkyView за указанный временной интервал и сравнить результат с Trimble GNSS Planning Online, полученный на прошлом этапе.

Рассчитаем количество секунд от начала текущей недели:

$$(24 \cdot 3 + 15) \cdot 3600 = 313200 \text{ с}$$

Моделирование производим в программе MatLab.

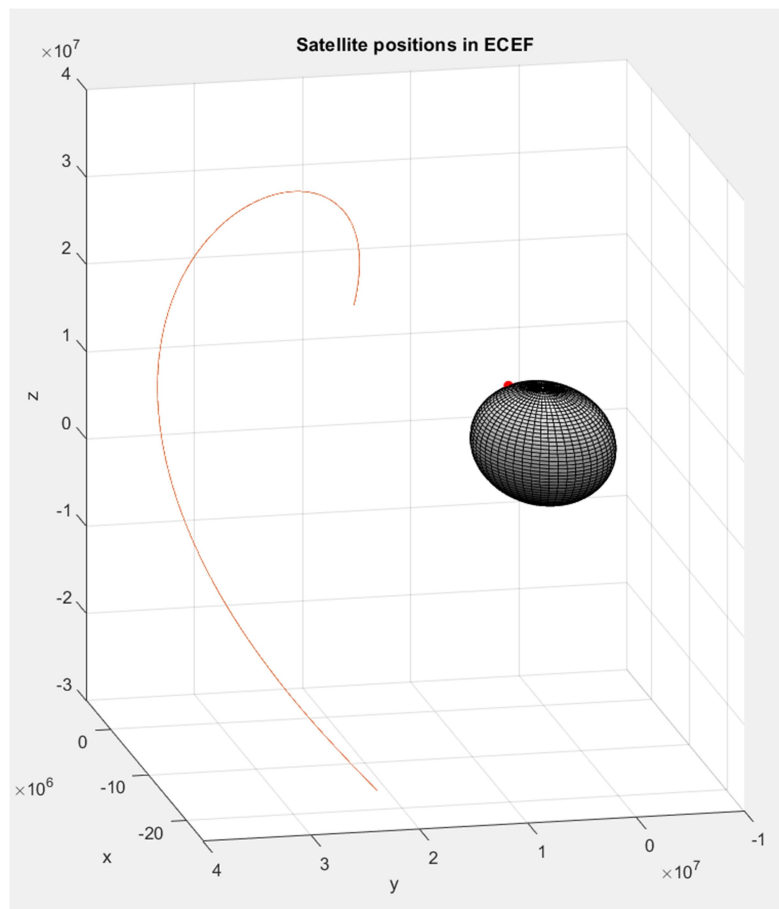


Рисунок 11 - Траектория движения спутника Beidou в системе ECEF.

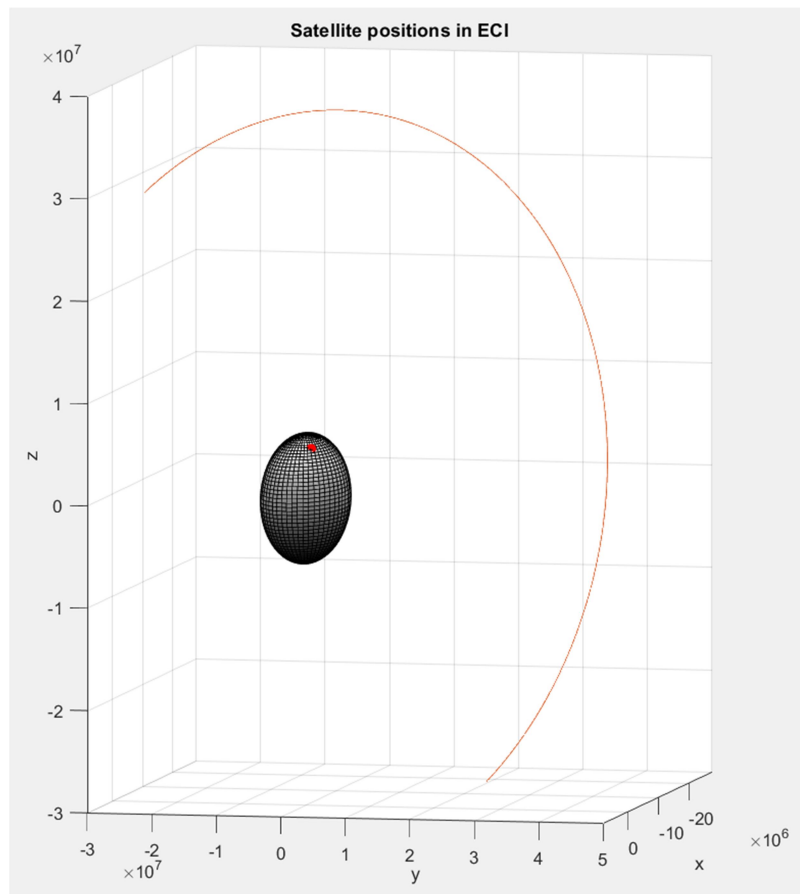


Рисунок 12 - Траектория движения спутника Beidou в системе ECI.

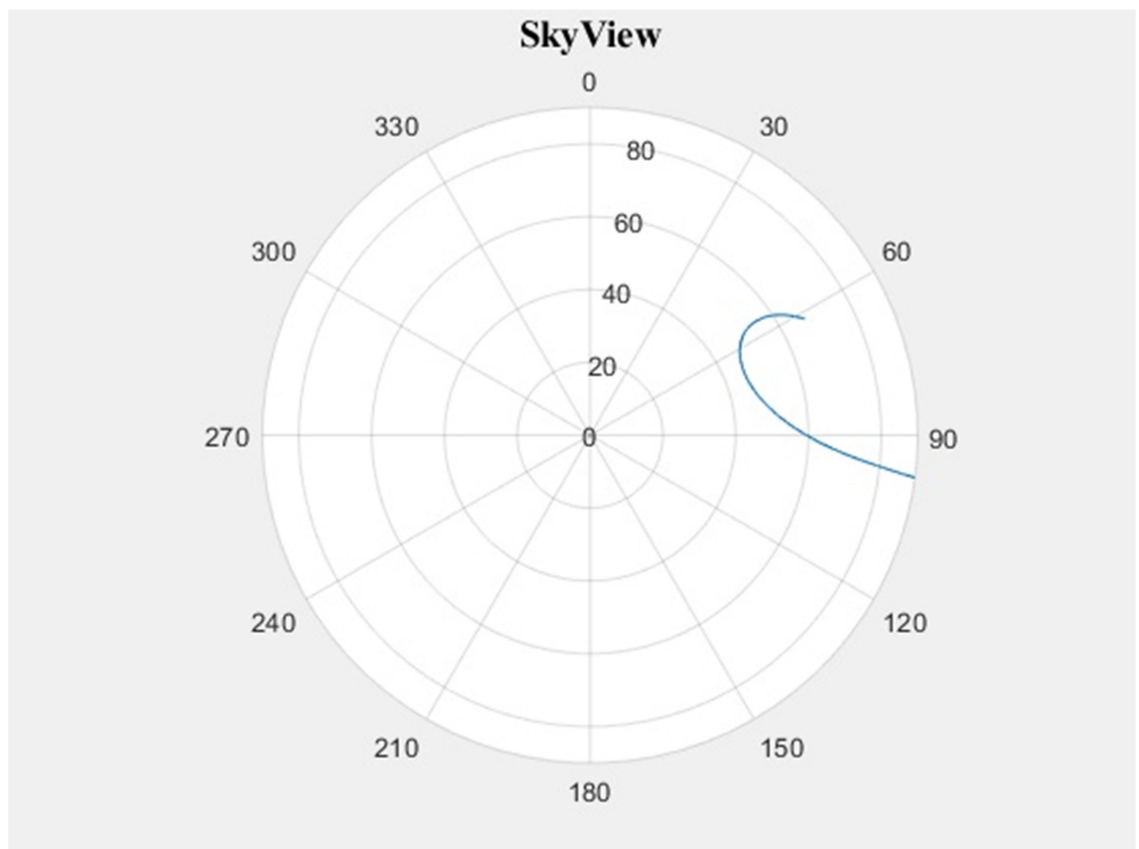


Рисунок 13 - SkyView спутника Beidou.

Сравним графики Sky View на рисунках 6 и 13, можно сказать что результаты совпадают с малейшими расхождениями, это связано с использованием одних эфемерид и особенностями алгоритмов программы MatLab.

Код программы в приложении.

Приложение.

```
close all;
```

```
clear all;
```

```
clc;
```

```
format long
```

```
%% Эфемериды
```

```
SatNum = 6;
```

```
toe = 241200;
```

```
Crs = -9.287500000000000e+01;
```

```
Dn = 8.71107710704449589e-13;
```

```
M0 = 2.32726368913121773e+00;
```

```
Cuc = -2.62074172496795654e-06;
```

```
e = 1.05765871703624725e-02;
```

```
Cus = 2.34702602028846741e-05;
```

```
sqrtA = 6.49287138557434082e+03;
```

```
Cic = -1.12690031528472900e-07;
```

```
Omega0 = 6.63759799965142852e-01;
```

```
Cis = 3.25962901115417480e-09;
```

```
i0 = 9.46015118241178121e-01;
```

```
Crc = -4.8214062500000000e+02;
```

```
omega = -2.20504767262928070e+00;
```

$\Omega_{\text{Dot}} = -1.773288508356074e-12;$

$i_{\text{Dot}} = -2.00008331149807446e-14;$

$T_{\text{gd}} = 9.750000000000000e+05;$

$t_{\text{oc}} = 2.196000000000000e+08;$

$a_{\text{f}2} = 1.48307593848345250e-22;$

$a_{\text{f}1} = 5.06794606280891458e-12;$

$a_{\text{f}0} = 3.53220045566558838e-01;$

$U_{\text{RA}} = 0;$

$I_{\text{ODE}} = 257;$

$I_{\text{ODC}} = 1;$

$\text{codeL2} = 0;$

$L2P = 0;$

$W_{\text{N}} = 789;$

%% Значения констант

$\mu = 3.986004418e14;$ % гравитационная постоянная

$\omega_e = 7.2921151467e-5;$ % скорость вращения

%% Временной промежуток

$\text{begin_time} = (24*2+18-3)*60*60;$ % время начала 8:00 по МСК 16 февраля

$\text{end_time} = (24*3+6-3)*60*60;$ % время окончания 6:00 по МСК 17 февраля

%% Длина временного промежутка

$t_{\text{arr}} = \text{begin_time}:1:\text{end_time};$

%% Большая полуось

$A = \sqrt{A^2};$

%% Среднее движение

```
n0 = sqrt(mu/A^3);
```

```
n = n0+Dn;
```

```
for k = 1:length(t_arr)
```

```
    % Vremya
```

```
    t(k) = t_arr(k)-toe;
```

```
    if t(k) > 302400
```

```
        t(k) = t(k)-604800;
```

```
    end
```

```
    if t(k) < -302400
```

```
        t(k) = t(k)+604800;
```

```
    end
```

```
    % Средняя аномалия
```

```
    M(k) = M0+n*t(k);
```

```
    % Решение уравнения Кеплера
```

```
    E(k) = M(k);
```

```
    E_old(k) = M(k)+1;
```

```
    epsilon = 1e-6;
```

```
    while abs(E(k)-E_old(k)) > epsilon
```

```
        E_old(k) = E(k);
```

```
        E(k) = M(k)+e*sin(E(k));
```

```
    end
```

```
    % Истинная аномалия
```

```
    nu(k) = atan2(sqrt(1-e^2)*sin(E(k)),cos(E(k))-e);
```

% Коэффициент коррекции

$\cos_correction(k) = \cos(2 * (\omega + \nu(k)))$;

$\sin_correction(k) = \sin(2 * (\omega + \nu(k)))$;

% Аргумент широты

$u(k) = \omega + \nu(k) + C_{uc} * \cos_correction(k) + C_{us} * \sin_correction(k)$;

% Радиус

$r(k) = A * (1 - e * \cos(E(k))) + C_{rc} * \cos_correction(k) + C_{rs} * \sin_correction(k)$;

% Наклон

$i(k) = i_0 + i_{dot} * t(k) + C_{ic} * \cos_correction(k) + C_{is} * \sin_correction(k)$;

% Долгота восходящего угла

$\lambda(k) = \Omega_0 + (\Omega_{dot} - \omega_e) * t(k) - \omega_e * t_0$;

% Положение на орбите

$x = r(k) * \cos(u(k))$;

$y = r(k) * \sin(u(k))$;

% Координаты

$X_0(k) = x * \cos(\lambda(k)) - y * \cos(i(k)) * \sin(\lambda(k))$;

$Y_0(k) = x * \sin(\lambda(k)) + y * \cos(i(k)) * \cos(\lambda(k))$;

$Z_0(k) = y * \sin(i(k))$;

%

$X(k) = X_0(k) * \cos(\lambda(k)) + Y_0(k) * \sin(\lambda(k))$;

$Y(k) = -X_0(k) * \sin(\lambda(k)) + Y_0(k) * \cos(\lambda(k))$;

```

Z(k) = Z0(k);

end

%% Из HKA в WGS84

ppb = 1e-9;

mas = 1e-3/206264.8; % [radian]

MATRIX_WGS_84 = [-3*ppb -353*mas -4*mas;
    353*mas -3*ppb 19*mas;
    4*mas -19*mas -3*ppb];

crd_WGS_84 = [X0; Y0; Z0];

for i = 1:length(crd_WGS_84(1,:))
    crd_WGS_84(:,i) = crd_WGS_84(:,i) + MATRIX_WGS_84 * crd_WGS_84(:,i) + [0.07; -0; -0.77];
end

crd_WGS_84 = crd_WGS_84.'; % perekhod k vektoru-stroke

%% postroenie grafikov

R_Earth = 6371e3;

[XE,YE,ZE] = sphere(10);

figure

surf(XE*R_Earth,YE*R_Earth,ZE*R_Earth)

hold on

grid on

plot3(crd_WGS_84(:,1), crd_WGS_84(:,2), crd_WGS_84(:,3))

plot3(X, Y, Z)

```

```
title('Satellite trajectory', 'FontName', 'Times New Roman', 'FontSize',14)
```

```
xlabel('X, m', 'FontName', 'Times New Roman', 'FontSize',14)
```

```
ylabel('Y, m', 'FontName', 'Times New Roman', 'FontSize',14)
```

```
zlabel('Z, m', 'FontName', 'Times New Roman', 'FontSize',14)
```

```
hold off
```

```
lgd = legend('Earth','CK ECEF WGS84','Inertial Coordinate System');
```

```
lgd.FontName = 'Times New Roman';
```

```
%% Перевод координат корпуса E в систему WGS84
```

```
% Широта(сначала идут значения - затем перевод)
```

```
N_gr = 55;
```

```
N_min = 45;
```

```
N_sec = 23.8178;
```

```
N = N_gr*pi/180+N_min/3437.747+N_sec/206264.8;
```

```
% Долгота(сначала идут значения - затем перевод)
```

```
E_gr = 37;
```

```
E_min = 42;
```

```
E_sec = 12.2608;
```

```
E = E_gr*pi/180+E_min/3437.747+E_sec/206264.8;
```

```
H = 500; % Приблизительное значение высоты расположения антенны на корпусе E(высота над  
уровнем моря + высота корпуса E)
```

```
llh = [N E H];
```

```
crd_PRM = llh2xyz(llh);
```



```
%% Построение SkyPlot
```

```
for i = 1:length(crd_WGS_84(:,1))
```

```
    [X(i) Y(i) Z(i)] =  
    ecef2enu(crd_WGS_84(i,1),crd_WGS_84(i,2),crd_WGS_84(i,3),N,E,H,wgs84Ellipsoid,'radians');
```

```
    if Z(i) > 0
```

```
        r(i) = sqrt(X(i)^2 + Y(i)^2 + Z(i)^2);
```

```
        teta(i) = acos(Z(i)/r(i));
```

```
        if X(i) > 0
```

```
            phi(i) = -atan(Y(i)/X(i))+pi/2;
```

```
        elseif (X(i)<0)&&(Y(i)>0)
```

```
            phi(i) = -atan(Y(i)/X(i))+3*pi/2;
```

```
        elseif (X(i)<0)&&(Y(i)<0)
```

```
            phi(i) = -atan(Y(i)/X(i))-pi/2;
```

```
        end
```

```
    else teta(i) = NaN;
```

```
        r(i) = NaN;
```

```
        phi(i) = NaN;
```

```
    end
```

```
end
```

```
%% skyplot
```

```
figure
```

```
ax = polaraxes;
```

```
polarplot(ax,phi,teta*180/pi)
```

```
ax.ThetaDir = 'clockwise';
```

```
ax.ThetaZeroLocation = 'top';
```

```
title('SkyView', 'FontName', 'Times New Roman', 'FontSize',14)
```

```
%% Построение графика угла места
```

```
th = hours(t_arr/3660 - 68); % перевод временной оси в формат hh:mm
```

```
figure
```

```
grid on
```

```
hold on
```

```
plot(th,(-teta)*180/pi+90,'DurationTickFormat','hh:mm') % временная ось
```

```
xlim([th(1) th(end)])
```

```
title('Elevation', 'FontName', 'Times New Roman', 'FontSize',14) % отображение названия графика
```

```
xlabel('Time', 'FontName', 'Times New Roman', 'FontSize',14) % отображение названия  
горизонтальной оси
```

```
ylabel('Elevation, deg', 'FontName', 'Times New Roman', 'FontSize',14) % отображение названия  
вертикальной оси
```

```
%% функция преобразования координат из WGS84 в ECEF
```

```
function xyz = llh2xyz(llh)
```

```
phi = llh(1); % llh(1) = широта в радианах
```

```
lambda = llh(2); % llh(2) = долгота в радианах
```

```
h = llh(3); % llh(3) = высота над уровнем моря в метрах
```

```
a = 6378137.0000; % полуось земли в метрах
```

```
b = 6356752.3142; % полуось земли в метрах
```

```
e = sqrt(1-(b/a).^2);
```

```
sinphi = sin(phi);
```

```
cosphi = cos(phi);
```

```
coslam = cos(lambda);
```

```
sinlam = sin(lambda);
```

```
tan2phi = (tan(phi))^2;
```

```
tmp = 1-e*e;
```

```
tmpden = sqrt(1+tmp*tan2phi);
```

```
x = (a*coslam)/tmpden+h*coslam*cosphi;
```

```
y = (a*sinlam)/tmpden+h*sinlam*cosphi;
```

```
tmp2 = sqrt(1-e*e*sinphi*sinphi);
```

```
z = (a*tmp*sinphi)/tmp2+h*sinphi;
```

```
xyz(1) = x; %    xyz(1) = ECEF x-координата в метрах
```

```
xyz(2) = y; %    xyz(2) = ECEF y-координата в метрах
```

```
xyz(3) = z; %    xyz(3) = ECEF z-координата в метрах
```

```
end
```