НИУ «МЭИ»

Кафедра Радиотехнических систем

Расчетно-пояснительная записка к курсовому проекту по

дисциплине «Аппаратура потребителей СРНС»

Выполнил: Чепелев И.И.

Группа: ЭР-15-17

Проверил: Корогодин И.В.

Москва 2022

**Этап 2. Моделирование траектории движения**

На данном этапе требуется реализовать функцию расчета положения спутника GPS на заданный момент по шкале времени UTC на языке Matlab или Python. Значения, полученные на предыдущем этапе, нужны нам в качестве эфемерид для моделирования.

Построить трехмерные графики множества положений спутника GPS с системным номером, соответствующим номеру студента по списку. Графики в двух вариантах: в СК ECEF WGS84 и соответствующей ей инерциальной СК. Положения должны соответствовать временному интервалу с 00:00 МСК 14 февраля до 00:00 МСК 15 февраля 2022 года. Допускается использовать одни и те же эфемериды на весь рассматриваемый интервал. Для выполнения данного этапа я использовал пакет математического моделирования Matlab.

Исходные данные:

Значения эфемерид, полученные в исходном сигнале и обработанные в первом пункте. Уточнение и проверка данных выполнено в RTKNAVI.

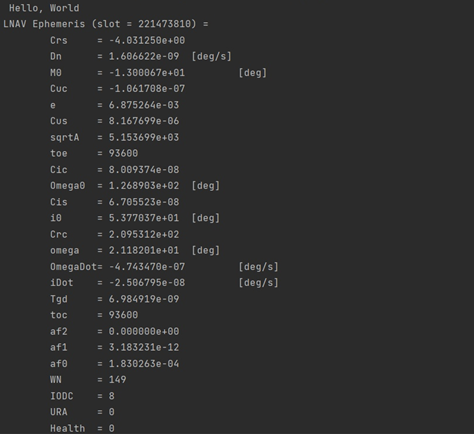
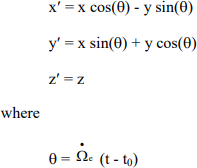




Рисунок 1 –– Значения эфемерид на заданную дату

На прошлом этапе эфемериды получены для даты 14.02.2022 0:00:00, Далее на суточном интервале (93600 секунд) по алгоритму из ИКД рассчитываются координаты спутника в системе ECEF WGS 84. Перевод координат в инерциальную систему осуществляется по заданным формулам:



Далее с помощью встроенной функции ecef2enu центр декартовой системы координат перемещается в точку приема, местоположение которой определяется с помощью RTKNAVI (рисунок 5). Полученные координаты спутника относительно приемника c учетом их знака пересчитываются в полярную систему координат с помощью известных соотношений:



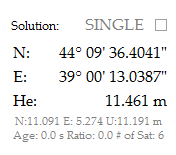


Рисунок 2 – Координаты приемника

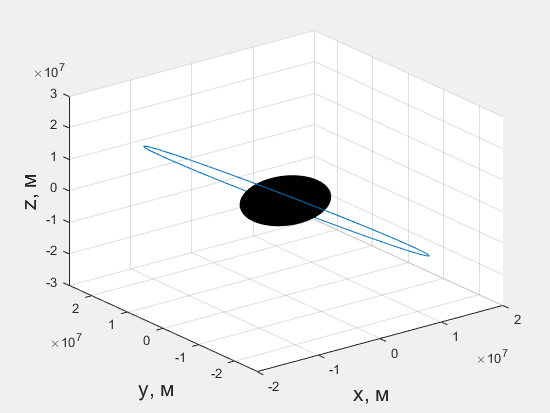


Рисунок 3 –– Множество положений спутника GPS в системе ECI

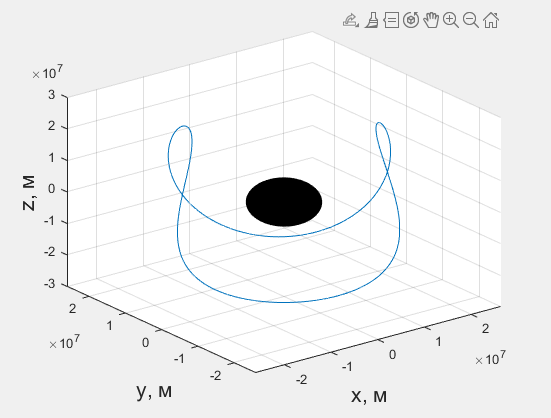


Рисунок 4 –– Множество положений спутника GPS в системе ECEF WGS84

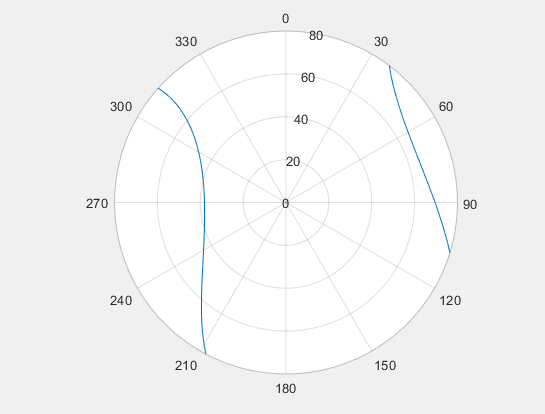


Рисунок 5 –– Sky Plot

Для построения SkyView с помощью Trimble GNSS Planning Online необходимо выбрать заданный спутник, дату, рассматриваемый период, а также координаты точки приема. Полученные настройки приведены на рисунке 6.

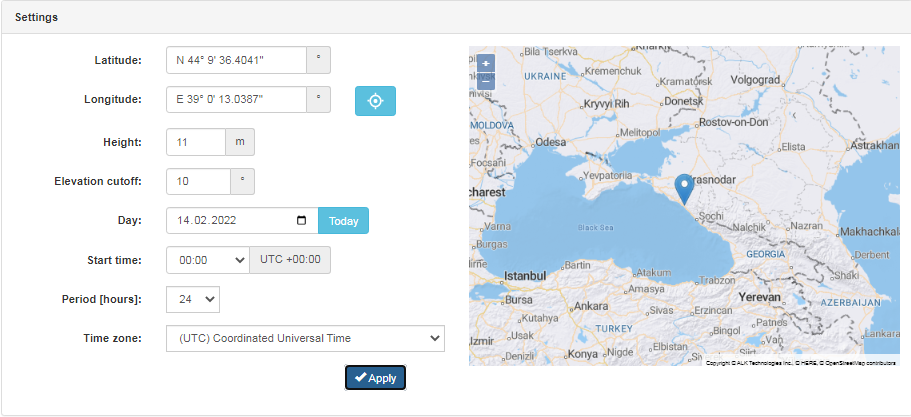


Рисунок 6 – Исходные данные для Trimble GNSS Planning Online

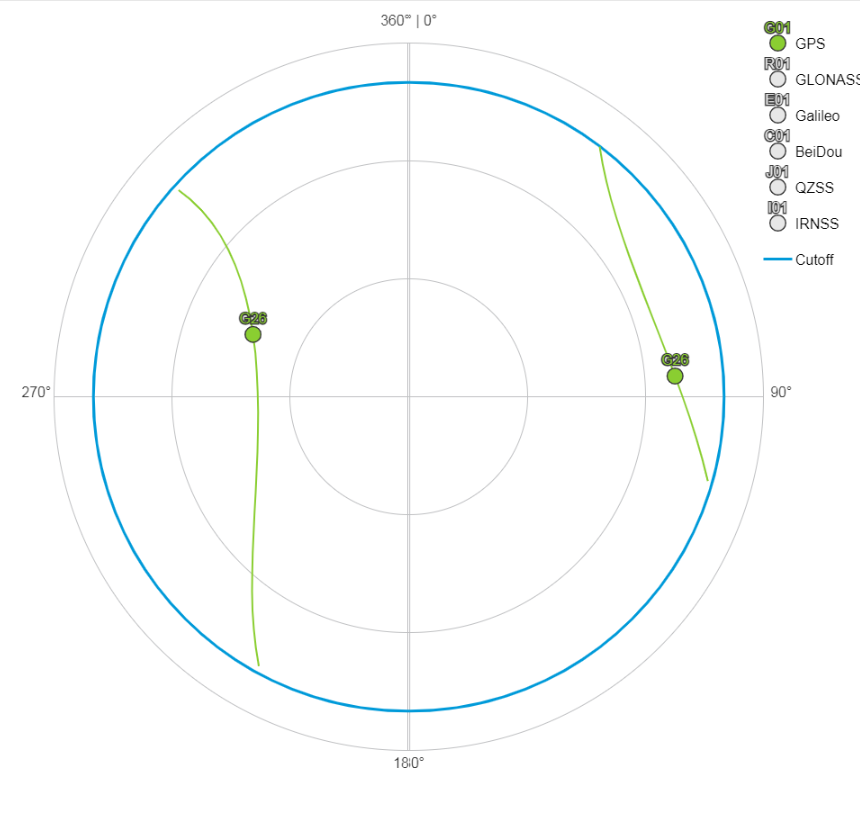


Рисунок 7 – SkyView из Trimble GNSS Planning Online

На данном этапе курсового проекта была разработана программа, выполняющая расчет положения спутника GPS, выводящая в файл out.txt значения координат спутника в заданном формате. При сравнении графиков на рисунках 5 и 7 видно сходство рассчитанного SkyView и полученного с помощью Trimble GNSS Planning Online.

**Приложение:**

clc

clear all

close all

% Эфемериды

C\_rs = 2.095312e+02; % Амплитуда поправочного члена синусоидальной гармоники к радиусу орбиты

delta\_n = deg2rad(1.606622e-9); % Среднее отклонение движения от вычисленного значения

M\_0 = deg2rad(-1.300067e+01); % Средняя аномалия

C\_uc = -1.061708e-07; % Амплитуда поправочного члена косинусной гармоники к аргументу широты

eks = 6.875264e-03; % Эксцентриситет

C\_us = 8.167699e-06; % Амплитуда поправочного члена синусоидальной гармоники к аргументу широты

rootA = 5.153699e+03; % Корень из большой полуоси

t\_oe = 93600; % Опорное время эфемерид

C\_ic = 8.009374e-08; % Амплитуда поправочного члена косинусной гармоники к углу наклона

Omega\_0 = deg2rad(1.268903e+02); % Долгота узла

C\_is = 6.705523e-08; % Амплитуда поправочного члена синусоидальной гармоники к углу наклона

i\_0 = deg2rad(5.377037e+01); % Наклонение

C\_rc = 2.095312e+02; % Амплитуда поправочного члена косинусной гармоники к радиусу орбиты

omega = deg2rad(2.118201e+01); % Аргумент перигея

Omega\_dot = deg2rad(-4.743470e-07); % Скорость прямого восхождения

IDOT = deg2rad(-2.506795e-08); % Скорость угла наклона

% Константы из ИКД

pi = 3.1415326535898;

mu = 3.986005e+14; % Гравитационная постоянная

Omega\_e\_dot = 7.2921151467e-5; % Скорость вращения Земли

% Расчет значений элементов Кеплера

A = rootA^2; % Большая полуось

n\_0 = sqrt(mu/A^3); % Расчетное среднее движение

for i = 1:86400 % Суточный интервал рассмотрения

t = 86400+18 + i; % Количество секунд от начала текущей недели при начале рассмотрения в 00:00:01

% Время от опорной эпохи эфемерид

t\_k = t - t\_oe;

if t\_k>302400

t\_k = t\_k - 604800;

elseif t\_k<-302400

t\_k = t\_k + 604800;

end

n = n\_0 + delta\_n; % Скорректированное среднее движение

M\_k = M\_0 + n \* t\_k; % Средняя аномалия

% Эксцентрическая аномалия

E\_0 = M\_k;

E\_k = 0;

k = 0;

while abs(E\_0 - E\_k)>1e-8

E\_k = E\_0;

E\_0 = E\_0 + (M\_k - E\_0 + eks \* sin(E\_0))/(1 - eks \* cos(E\_0));

k = k + 1;

end

nu\_k = atan2((sqrt(1 - eks^2) \* sin(E\_k)/(1 - eks \* cos(E\_k))), ((cos(E\_k) - eks)/(1 - eks \* cos(E\_k)))); % Истинная аномалия

Phi\_k = nu\_k + omega; % Аргумент широты

delta\_u\_k = C\_us\*sin(2\*Phi\_k) + C\_uc\*cos(2\*Phi\_k); % Аргумент поправки на широту

delta\_r\_k = C\_rs\*sin(2\*Phi\_k) + C\_rc\*cos(2\*Phi\_k); % Коррекция радиуса

delta\_i\_k = C\_is\*sin(2\*Phi\_k) + C\_ic\*cos(2\*Phi\_k); % Коррекция наклона

u\_k = Phi\_k + delta\_u\_k; % Скорректированный аргумент широты

r\_k = A \* (1 - eks \* cos(E\_k)) + delta\_r\_k; % Скорректированный радиус

i\_k = i\_0 + delta\_i\_k + IDOT \* t\_k; % Скорректированное наклонение

% Позиция в орбитальной плоскости

x\_k\_sh = r\_k \* cos(u\_k);

y\_k\_sh = r\_k \* sin(u\_k);

Omega\_k = Omega\_0 + (Omega\_dot - Omega\_e\_dot) \* t\_k - Omega\_e\_dot \* t\_oe; %Скорректированная широта восходящего узла

% Расчет координат спутника

x\_k = x\_k\_sh \* cos(Omega\_k) - y\_k\_sh \* cos(i\_k) \* sin(Omega\_k);

y\_k = x\_k\_sh \* sin(Omega\_k) + y\_k\_sh \* cos(i\_k) \* cos(Omega\_k);

z\_k = y\_k\_sh \* sin(i\_k);

x\_current(1,i) = x\_k;

y\_current(1,i) = y\_k;

z\_current(1,i) = z\_k;

% Инерциальная система координат

theta = Omega\_e\_dot \* t\_k;

x\_k\_1 = x\_k \* cos(theta) - y\_k \* sin(theta);

y\_k\_1 = x\_k \* sin(theta) + y\_k \* cos(theta);

z\_k\_1 = z\_k;

x\_converted(1,i) = x\_k\_1;

y\_converted(1,i) = y\_k\_1;

z\_converted(1,i) = z\_k\_1;

% Данные из RTKNAVI (координаты приемника)

B = deg2rad(44.09363261); % Широта

L = deg2rad(39.00130546); % Долгота

H = 1.247; % Высота

[x(i), y(i), z(i)] = ecef2enu(x\_current(1,i),y\_current(1,i),z\_current(1,i),B,L,H,wgs84Ellipsoid,'radians'); % Отображение координат спутника относительно точки приема

% Полярная система координат

if z(i) > 0

rho(i) = sqrt(x(i)^2 + y(i)^2 + z(i)^2);

theta\_1(i) = acos(z(1,i)/rho(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 theta\_1(i) = NaN;

rho(i) = NaN;

phi(i) = NaN;

end

end

% Вывод значений координат

out = fopen('../out.txt', 'w+');

for i = 1:length(x\_current)

fprintf(out, '%6.0f %10.10f %10.10f %10.10f\n', i, x\_current(1,i), y\_current(1,i), z\_current(1,i));

end

fclose(out);

% Вывод графиков

R = 6378136; % Радиус Земли

[x\_sphere, y\_sphere, z\_sphere] = sphere(100);

x\_Earth=R\*x\_sphere;

y\_Earth=R\*y\_sphere;

z\_Earth=R\*z\_sphere;

figure (1)

subplot (1,1,1)

surf(x\_Earth,y\_Earth,z\_Earth);

hold on;

plot3(x\_current(1,:), y\_current(1,:), z\_current(1,:));

xlabel('x, м','FontSize',16);

ylabel('y, м','FontSize',16);

zlabel('z, м','FontSize',16);

figure (2)

subplot (1,1,1)

surf(x\_Earth,y\_Earth,z\_Earth);

hold on;

plot3(x\_converted(1,:), y\_converted(1,:), z\_converted(1,:));

xlabel('x, м','FontSize',16);

ylabel('y, м','FontSize',16);

zlabel('z, м','FontSize',16);

figure (3)

axes = polaraxes;

polarplot(axes,phi,rad2deg(theta\_1))

axes.ThetaDir = 'clockwise';

axes.ThetaZeroLocation = 'top';

rlim([0 80]);