

Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Национальный исследовательский университет «МЭИ»

Институт: *ИРЭ*

Кафедра: *РТС*

Направление подготовки:

*11.05.01 Радиоэлектронные системы и  
комплексы*

**ОТЧЕТ по курсовой работе**

**Разработка модуля расчёта координат спутника Beidou**

**СТУДЕНТ**



(подпись)

/ **Лихачёв М.С.**

(Фамилия и инициалы)

**Группа**

**ЭР-15-16**

(номер учебной группы)

**ЗАЩИТА КУРСОВОЙ РАБОТЫ**

(отлично, хорошо, удовлетворительно, неудовлетворительно,  
зачтено, не зачтено)

/ **Корогодин И.В.** /

(подпись)

(Фамилия и инициалы члена комиссии)

/ **Шатилов А.Ю.** /

(подпись)

(Фамилия и инициалы члена комиссии)

**Москва  
2021**

## ПОСТАНОВКА ЗАДАЧИ

**Цель проекта** - добавление в программное обеспечение приемника функции расчета положения спутника Beidou на заданное время по данным его эфемерид.

**Исходные данные:** Спутник № 12 системы Beidou.

Требования к разрабатываемому программному модулю:

- требования назначения;
- отсутствие утечек памяти;
- малое время выполнения;
- низкий расход памяти;
- корректное выполнение при аномальных входных данных.

Для достижения цели выполняется ряд задач, соответствующих этапам проекта и контрольным мероприятиям:

- обработка данных от приемника, работа со сторонними сервисами для подготовки входных и проверочных данных для разрабатываемого модуля;
- моделирование модуля в Matlab/Python;
- реализация программного модуля на C/C++, включая юнит-тестирование в Check.

### **Этапы работы:**

На первом этапе подготовим вспомогательные данные для разработки: эфемериды и оценки положения спутника от сторонних сервисов.

На втором этапе требуется реализовать расчет координат для заданного спутника, также получить сравнимые значения и графические отображения траекторий основных характеристик.

На третьем этапе требуется разработать на языке C/C++ функцию расчета положения спутника Beidou на заданное время по шкале UTC, минимизируя время её исполнения и количество затрачиваемой оперативной памяти. Вызов функции не должен приводить к выбросу исключений или утечкам памяти при любом наборе входных данных.

## ЭТАП 1. ИСПОЛЬЗОВАНИЕ СТОРОННИХ СРЕДСТВ

**Цель курсового проекта** - получить библиотеку функций на Си++, позволяющую рассчитывать положение спутника Beidou по его эфемеридам.)

На крыше корпуса Е МЭИ установлена трехдиапазонная антенна Narxon NX-CSX601A. Она через 50-метровый кабель, сплиттер, bias-tee и усилитель подключена к трем навигационным приемникам:

- Javad Lexon LGDD,
- SwiftNavigation Piksi Multi,
- Clonicus разработки ЛНС МЭИ.

Эти приемники осуществляют первичную обработку сигналов Beidou B1I, выдавая по интерфейсам соответствующие потоки данных - наблюдения псевдодальностей и эфемериды спутников. Данные от приемника Clonicus, записанные вечером 16 февраля 2021 года.

С помощью «Википедии» определим номер НОРАД и сравним его с номером из «Информационно-аналитического центра координатно-временного и навигационного обеспечения».

12	Компас М3	C11	29.04.2012 20:50	CZ-3B/E	2012-018A	38250	COO, ~21 500 км	действующий
13	Компас М4	C12			2012-018B	38251	COO, ~21 500 км	

Рисунок 1 – Состояние космического аппарата Beidou на Википедии

СОСТОЯНИЕ КОСМИЧЕСКИХ АППАРАТОВ БЭЙДОУ НА 05.03.21

PRN	НОРАД	Тип КА	Тип системы	Дата запуска	Факт. суц. (дней)	Примечание
C12	38251	МEO-4	BDS-2	30.04.12	3231	Используется по ЦН

Рисунок 2 – Состояние космического аппарата Beidou на «Информационно-аналитического центра координатно-временного и навигационного обеспечения»

Номера спутника совпадают и равны 38251, название спутника - «Компас М4».

## **Определение орбиты и положения спутника на ней с помощью сервиса CelesTrak.**

При помощи сервиса CelesTrak можно получить изображение формы орбиты и положение спутника на ней. Изображения по требованию в задании необходимо получить на период 18:00 МСК 16 февраля 2021. 18:00 по МСК соответствует 15:00 по UTC (UTC +3). Так как сервис CelesTrak работает в формате времени UTC, установим время 15:00 UTC 16 февраля 2021.

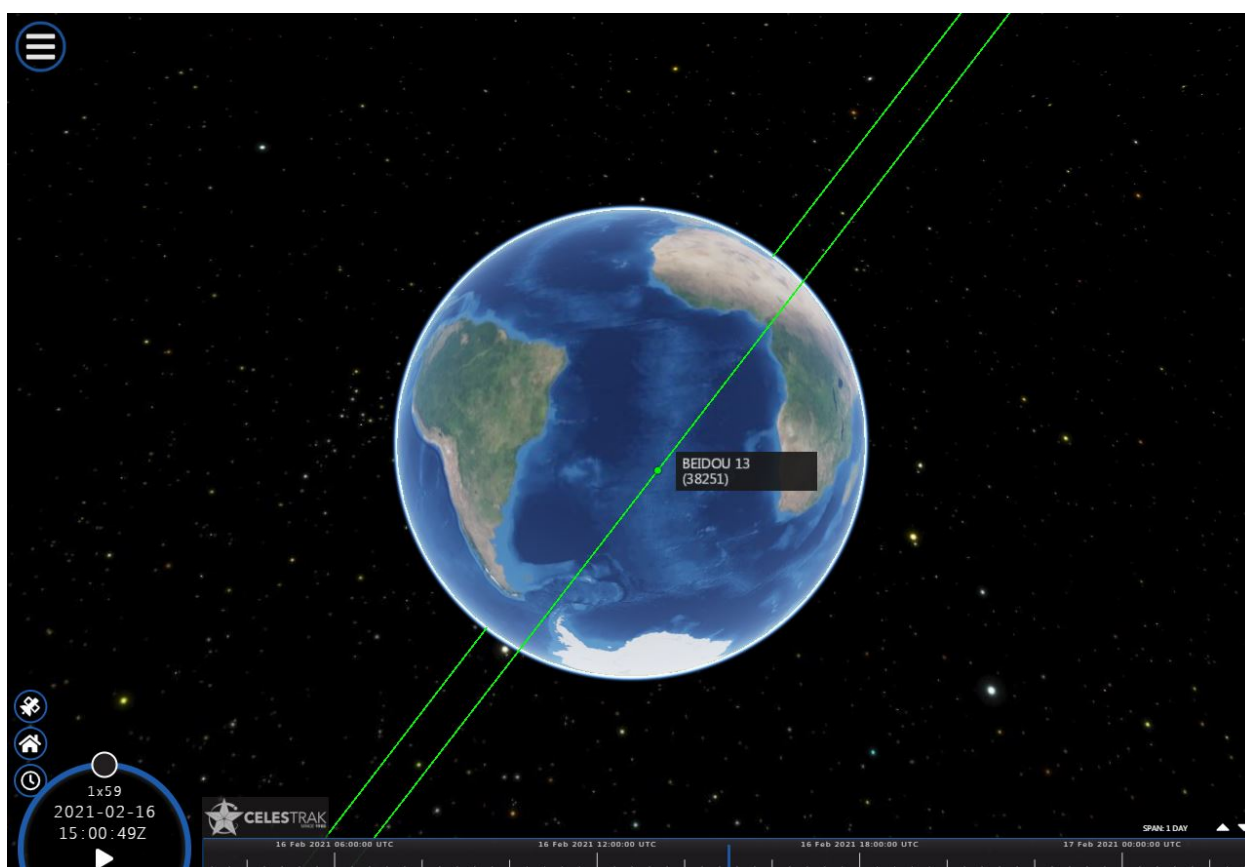


Рисунок 3 — Положение спутника на орбите

## **Расчет графика угла места собственного спутника от времени по данным Trimble GNSS Planning Online.**

Рассчитаем график угла места собственного спутника от времени по данным Trimble GNSS Planning Online на интервал времени с 18:00 МСК 16 февраля до 06:00 МСК 17 февраля 2021 года.

Satellite Selection

Change selection

Satellites: 49/131

System: active	Satellites	
	Selected	Healthy
GPS	<input checked="" type="checkbox"/>	0 32
GLONASS	<input checked="" type="checkbox"/>	0 23
Galileo	<input checked="" type="checkbox"/>	0 22
BeiDou	<input checked="" type="checkbox"/>	49 49
QZSS	<input checked="" type="checkbox"/>	0 4

My Settings

Time of almanac: 2021-02-16  
Time zone: UTC +03:00  
Visible period:  
2021-02-16 18:00 - 2021-02-17 18:00  
Latitude: N 55° 45' 23.5244"  
Longitude: E 37° 42' 12.0821"  
Height: 500 m  
Elevation cutoff: 10 °

Settings

Latitude: N 55° 45' 23.7744"  
Longitude: E 37° 42' 12.1739"  
Height: 500 m  
Elevation cutoff: 10 °  
Day: 16.02.2021 Today  
Start time: 18:00 UTC +03:00  
Period [hours]: 12  
Time zone: (UTC+03:00) Moscow, St. Petersburg  
Apply

Рисунок 4 — Настройки Trimble GNSS Planning Online

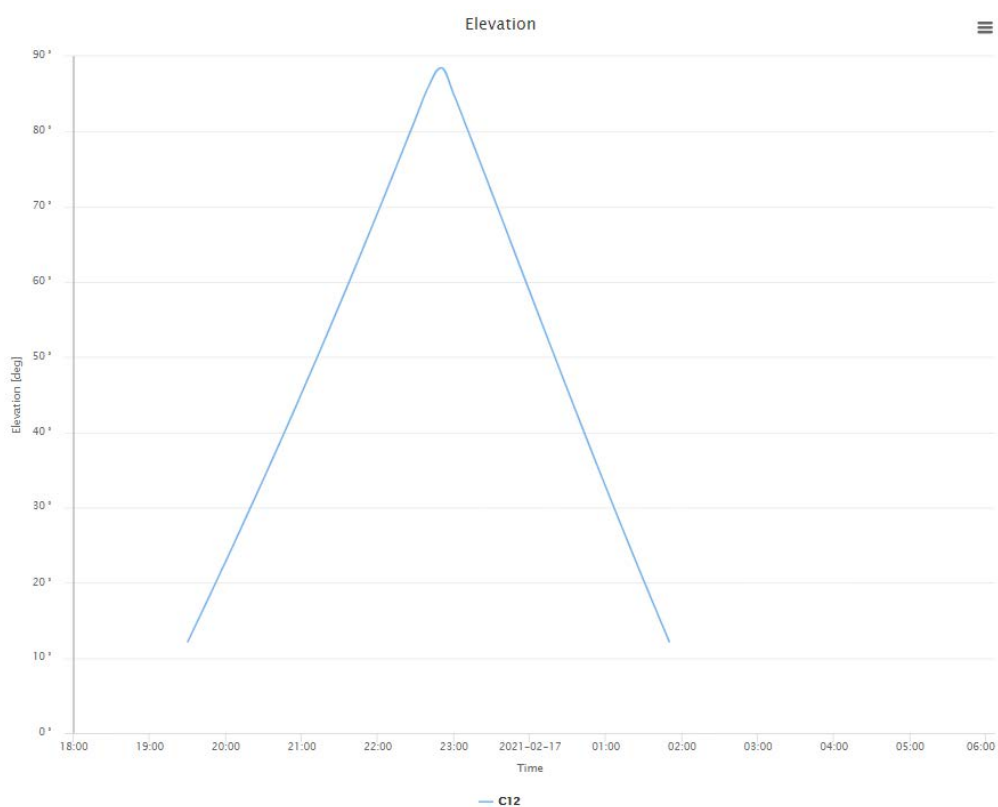


Рисунок 5 — График угла места спутника C12 от времени

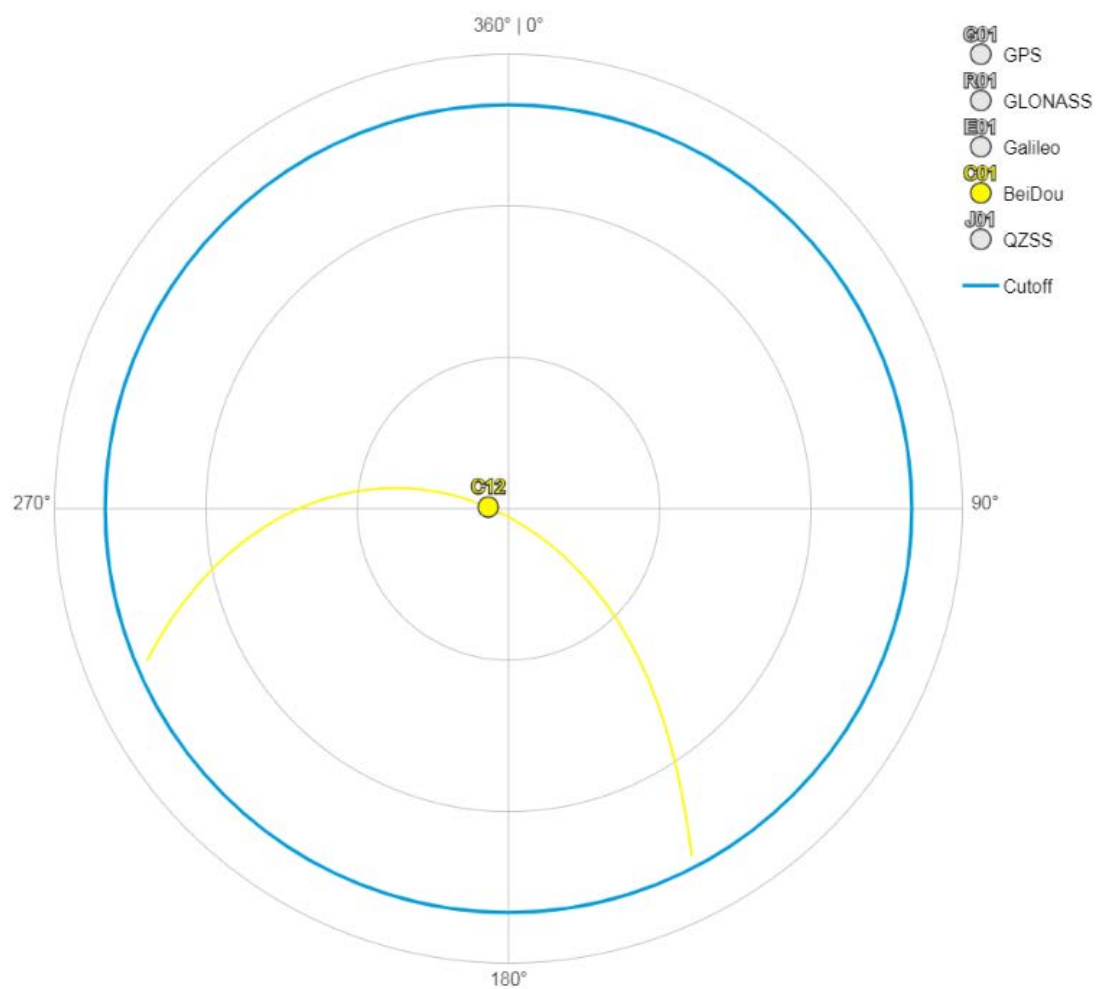


Рисунок 6 – SkyView спутника Beidou C12

## Формирование списка и описание параметров, входящих в состав эфемерид

Таблица 1 – Описание параметров, входящих в состав эфемерид

Параметры	Определение
$t_{oe}$	Отсчет времени эфемерид
$\sqrt{A}$	Квадратный корень из большой полуоси орбиты
$e$	Эксцентриситет
$\omega$	Аргумент перигея
$\Delta n$	Среднее отклонение движения от расчетного значения
$M_0$	Средняя аномалия в исходное время
$\Omega_0$	Долгота восходящего узла орбитальной плоскости, вычисленная по опорному времени
$\dot{\Omega}$	Скорость прямого восхождения
$i_0$	Угол наклона в исходное время
$IDOT$	Скорость угла наклона
$C_{uc}$	Амплитуда косинусной поправки к аргументу широты
$C_{us}$	Амплитуда синусной поправки к аргументу широты
$C_{rc}$	Амплитуда косинусной поправки к радиусу орбиты
$C_{rs}$	Амплитуда синусной поправки к радиусу орбиты
$C_{ic}$	Амплитуда косинусной поправки к углу наклона



$C_{is}$	Амплитуда синусной поправки к углу наклона
----------	--

### Формирование таблицы эфемерид собственного спутника

Данные спутника берутся из текстового файла, полученного из дампа бинарного потока данных от приемника в формате NVS BINR.

Таблица 2 – Значения эфемерид спутника C12

Параметры	Значение	Размерность
SatNum	12	-
toe, $t_{oe}$	241200000.000	мс
Crs, $C_{rs}$	3.9687500000000000e+00	рад
Dn, $\Delta n$	3.06691347665144498e-12	рад/мс
M0, $M_0$	-3.07094212214015183e+00	рад
Cuc, $C_{uc}$	1.85333192348480225e-07	рад
e	1.23757799156010151e-03	-
Cus, $C_{us}$	1.09374523162841797e-05	рад
sqrtA, $\sqrt{A}$	5.28261268997192383e+03	м <sup>1/2</sup>
Cic, $C_{ic}$	2.60770320892333984e-08	рад
Omega0, $\Omega_0$	-2.36115369616889925e+00	рад
Cis, $C_{is}$	2.37487256526947021e-08	рад

$i_0, \dot{i}_0$	9.89305735914883022e-01	рад
$C_{rc}, C_{rs}$	1.5196875000000000e+02	рад
$\omega, \omega$	-1.81447090177903547e+00	рад
$\Omega, \dot{\Omega}$	-6.44633994450513345e-12	рад/мс
$\dot{I}, IDOT$	-8.67893294096485793e-14	рад/с
$T_{gd}, T_{GD}$	3.1000000000000000e+04	мс
$t_{oc}, t_{oc}$	2.4120000000000000e+08	мс
$a_{f2}, a_{f2}$	1.89735386041466636e-22	мс/мс <sup>2</sup>
$a_{f1}, a_{f1}$	2.65760746742671472e-11	мс/мс
$a_{f0}, a_{f0}$	8.02835941314697266e-01	мс
URA	0	-
IODE	3598	-
IODC	13	-
codeL2	0	-
L2P	0	-
WN	789	-

## 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 проводится в программе Matlab. Код программы приведен в приложении 1.

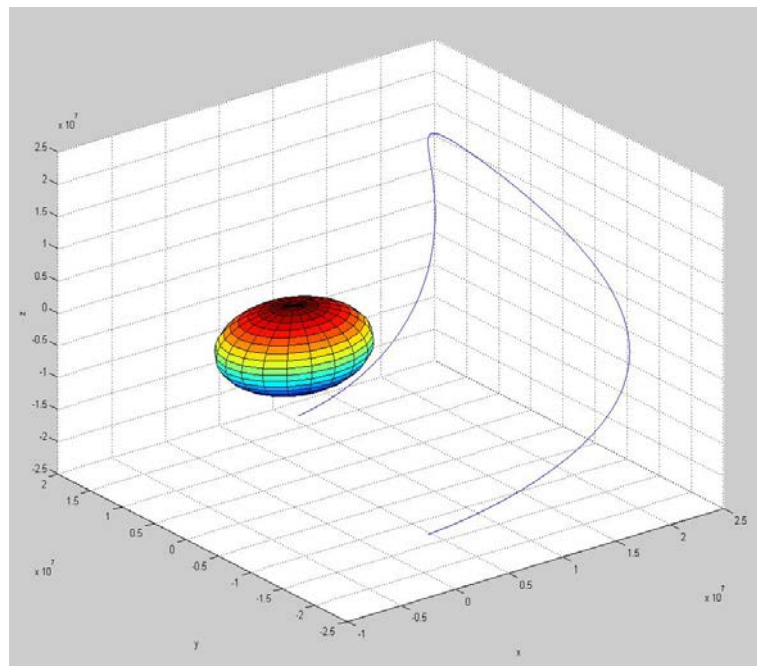


Рисунок 7 - Траектория движения спутника Beidou в системе ECEF

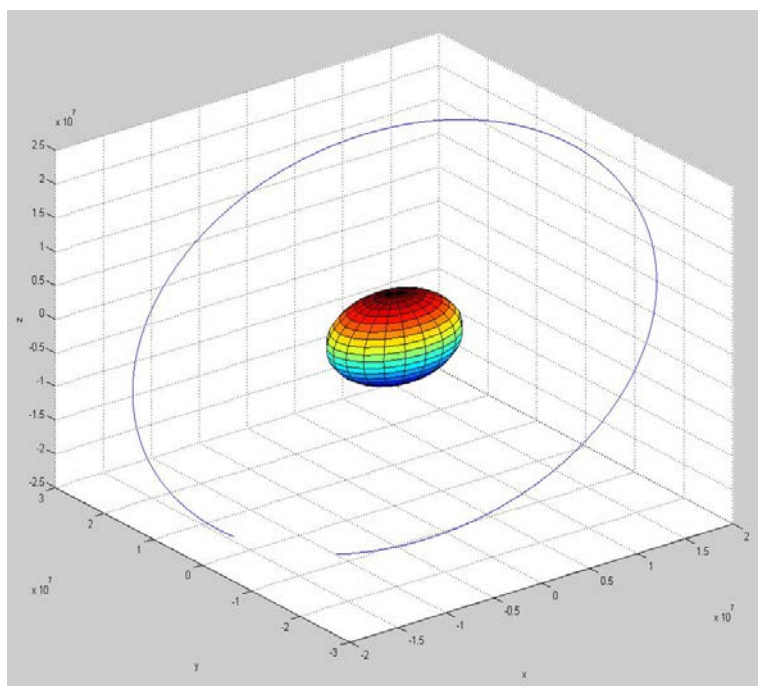


Рисунок 8 - Траектория движения спутника Beidou в системе ECI

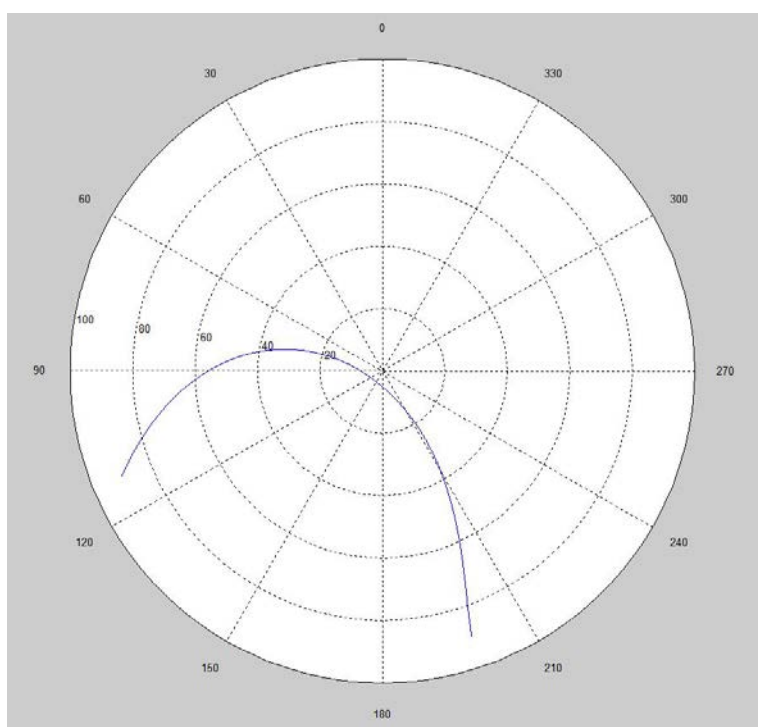


Рисунок 9 - SkyView спутника Beidou

Сравнивая графики SkyView на рисунках 6 и 9, можно сказать, что результаты совпадают с небольшими отличиями, которые могут быть связаны из - за использования одних эфемерид при моделировании.

### ЭТАП 3. РЕАЛИЗАЦИЯ

Требуется разработать на языке C/C++ функцию расчета положения спутника Beidou на заданное время по шкале UTC, минимизируя время её исполнения и количество затрачиваемой оперативной памяти. Вызов функции не должен приводить к выбросу исключений или утечкам памяти при любом наборе входных данных.

Функция расчета положения спутника в Matlab/Python относительно проста, т.к. доступны библиотеки линейной алгебры и решения уравнений. Но при разработке встраиваемого ПО приходится сохранять лицензионную частоту, минимизировать вычислительную нагрузку и затраты памяти. Поэтому отобразить модель из Matlab/Python в прошивку приемника дословно, как правило, не получается. В рассматриваемом примере потребуется, как минимум, выполнить свою реализацию решения трансцендентного уравнения.

#### Код реализации

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    setlocale(LC_ALL, "rus");
    std::cout << "Начало расчета!\n";
    time_t t_start, t_stop;
    time(&t_start);
    double SatNum = 12;
    double toe = 241200;
    double Crs = 3.9687500000000000e+00;
    double Dn = 3.06691347665144498e-12;
    double M0 = -3.07094212214015183e+00;
    double Cuc = 1.85333192348480225e-07;
```

```

double e = 1.23757799156010151e-03;
double Cus = 1.09374523162841797e-05;
double sqrtA = 5.28261268997192383e+03;
double Cic = 2.60770320892333984e-08;
double Omega0 = -2.36115369616889925e+00;
double Cis = 2.37487256526947021e-08;
double i0 = 9.89305735914883022e-01;
double Crc = 1.51968750000000000e+02;
double omega = -1.81447090177903547e+00;
double OmegaDot = -6.44633994450513345e-12;
double iDot = -8.67893294096485793e-14;
double Tgd = 3.1000000000000000e+04;
double toc = 2.4120000000000000e+08;
double af2 = 1.89735386041466636e-22;
double af1 = 2.65760746742671472e-11;
double af0 = 8.02835941314697266e-01;
double URA = 0;
double IODE = 3598;
double IODC = 13;
double codeL2 = 0;
double L2P = 0;
double WN = 789;
// Значения констант
double mu = 3.986004418e14; // гравитационная постоянная
double omega_e = 7.2921151467e-5; // скорость вращения
// Временной промежуток
int begin_time = (24 * 2 + 18 - 3) * 60 * 60; // время начала 8:00 по МСК 16 февраля
int end_time = (24 * 3 + 6 - 3) * 60 * 60; // время окончания 6:00 по МСК 17 февраля
//Длина временного промежутка
int t_pr = end_time - begin_time;
//Большая полуось
double A = pow(sqrtA,2);
// Среднее движение

```

```

double n0 = sqrt(mu /pow(A,3));
double n = n0 + Dn;
// Координаты
double** coord = new double*[3];
for (int i = 0; i < 3; i++) {
coord[i] = new double[t_pr;
}
// Координаты из матлаба
double** coordMat = new double*[3];
for (int i = 0; i < 3; i++) {
coordMat[i] = new double[t_pr];
}
for (int k = 0; k < t_pr; k++) {
// Время
double t = begin_time + k - (int)toe;
if (t > 302400)
t = t - 604800;
if (t < -302400)
t = t + 604800;
// Средняя аномалия
double M = M0 + n * t;
// Решение уравнения Кеплера
double E = M;
double E_old = M + 1;
double epsilon = 1e-6;
while (abs(E - E_old) > epsilon) {
E_old = E;
E = M + e * sin(E);
}
// Истинная аномалия
double nu = atan2(sqrt(1 - e*e) * sin(E), cos(E) - e);
// Коэффициент коррекции
double cos_correction = cos(2 * (omega + nu));

```

```

double sin_correction = sin(2 * (omega + nu));
// Аргумент широты
double u = omega + nu + Cuc * cos_correction + Cus * sin_correction;
// Радиус
double r = A * (1 - e * cos(E)) + Crc * cos_correction + Crs * sin_correction;
// Наклон
double i = i0 + iDot * t + Cic * cos_correction + Cis * sin_correction;
// Долгота восходящего угла
double lambda = Omega0 + (OmegaDot - omega_e) * t - omega_e * toe;
// Положение на орбите
double x = r * cos(u);
double y = r * sin(u);
// Координаты
double Xk = x * cos(lambda) - y * cos(i) * sin(lambda);
double Yk = x * sin(lambda) + y * cos(i) * cos(lambda);
double Zk = y * sin(i);
coord[0][k] = Xk;
coord[1][k] = Yk;
coord[2][k] = Zk;
}
// Чтение координат из матлаба
ifstream file("data_matlab.txt");
if (!file.is_open())
cout << "Файл не может быть открыт!" << endl;
else {
for (int k = 0; k < t_pr; k++) {
file >> coordMat[0][k] >> coordMat[1][k] >> coordMat[2][k];
}
file.close();
}
// Сравнение с матлабом
double max_del = 0;
for (int i = 0; i < 3; i++) {

```



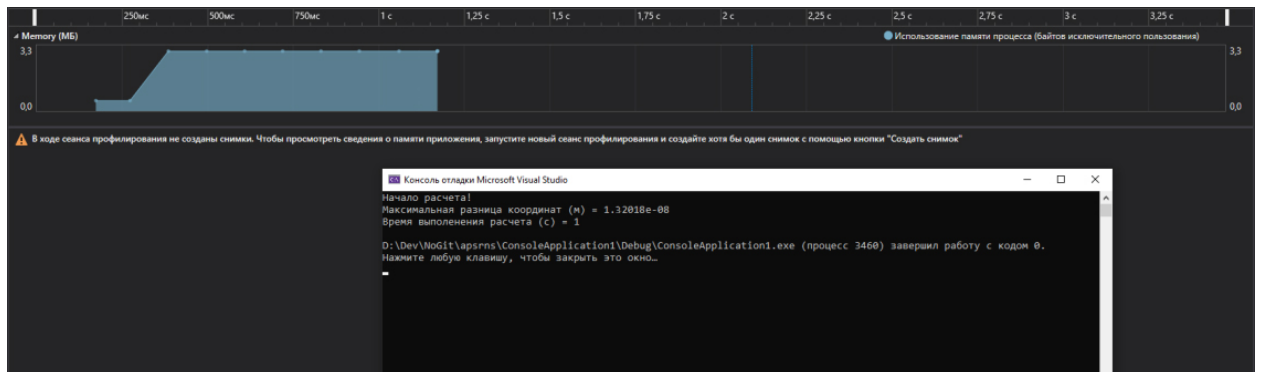
```

for (int k = 0; k < t_pr; k++) {
    if (abs(coord[0][k] - coordMat[0][k]) > max_del) {
        max_del = abs(coord[0][k] - coordMat[0][k]);
    }
}

// Очищение памяти
delete[] *coord;
delete[] coord;
delete[] *coordMat;
delete[] coordMat;
time(&t_stop);
double del_time = difftime(t_stop, t_start);
cout << "Максимальная разница координат (м) = " << max_del << endl;
cout << "Время выполнения расчета (с) = " << del_time << endl;
}

```

## Анализ программы



Время выполнения программы увеличилось и составило примерно 1,15 секунды.

На этом этапе работы была реализована на языке C/C++ функция расчета положения спутника Veidou на заданное время по шкале UTC. Погрешность вычисления координат функцией и моделью составляет порядка  $1,32e-8$  м, при использовании вещественного типа данных двойной точности `double`. Время выполнение функции составляет одну секунду. Утечек памяти не обнаружено.

## **ЗАКЛЮЧЕНИЕ**

В рамках данного курсового проекта был произведен анализ ИКД Beidou, который состоит из трех этапов.

На первой этапе была проведена ознакомительная работа сторонних средств с помощью которых была построена орбита заданного спутника Beidou, также получена траектория его движения на заданный промежуток времени.

На втором этапе был реализован расчет координат для заданного спутника, были получены сравнимые значения и графические отображения траекторий основных характеристик.

На третьем, заключительном, этапе была разработка функции расчета местоположения спутника Beidou на языке C++. В ходе выполнения данного этапа было произведено сравнение результатов с программой из Matlab.

## ПРИЛОЖЕНИЕ

### Приложение 1

```
clc
close all
%% Параметры
toe = 241200;
Crs = 3.9687500000000000e+00;
Dn = 3.06691347665144498e-12;
M0 = -3.07094212214015183e+00;
Cuc = 1.85333192348480225e-07;
e = 1.23757799156010151e-03;
Cus = 1.09374523162841797e-05;
A = 5.28261268997192383e+03^2;
Cic = 2.60770320892333984e-08;
Omega_0 = -2.36115369616889925e+00;
Cis = 2.37487256526947021e-08;
i0 = 9.89305735914883022e-01;
Crc = 1.5196875000000000e+02;
omega = -1.81447090177903547e+00;
OmegaDot = -6.44633994450513345e-12;
iDot = -8.67893294096485793e-14;
mu = 3.986004418e+14;
Omega_e = 7.2921151467e-5;
n0 = sqrt(mu/A^3);
n = n0 + Dn;
%% Расчет положения спутника Beidou
for i = 1:43200
    t = 313200 + i;
    tk = t - toe;
    Mk = M0 + n*tk;
    E = zeros(1,4);
    E(1,1) = Mk;
    for k = 2:4
        E(1,k) = Mk + e*sin(E(1,k - 1));
        E(1,k-1) = E(1,k);
    end
    Ek = E(1,4);
    vk = atan2(sqrt(1 - e^2)*sin(Ek),cos(Ek) - e);
    Fk = vk + omega;
    del_uk = Cus*sin(2*Fk) + Cuc*cos(2*Fk);
    del_rk = Crs*sin(2*Fk) + Crc*cos(2*Fk);
    del_ik = Cis*sin(2*Fk) + Cic*cos(2*Fk);
    uk = Fk + del_uk;
    rk = A*(1 - e*cos(Ek)) + del_rk;
    ik = i0 + del_ik + iDot*tk;
    xk = rk*cos(uk);
    yk = rk*sin(uk);
    Omega_k = Omega_0 + (OmegaDot - Omega_e)*tk - Omega_e*toe;
    Xk = xk*cos(Omega_k) - yk*cos(ik)*sin(Omega_k);
    Yk = xk*sin(Omega_k) + yk*cos(ik)*cos(Omega_k);
```

```

Zk = yk*sin(ik);
Kx(1,i) = Xk;
Ky(1,i) = Yk;
Kz(1,i) = Zk;
Theta = Omega_e*tk;
Xk1 = Xk*cos(Theta) - Yk*sin(Theta);
Yk1 = Xk*sin(Theta) + Yk*cos(Theta);
Zk1 = Zk;
Kx1(1,i) = Xk1;
Ky1(1,i) = Yk1;
Kz1(1,i) = Zk1;
H = 500;
a = 6378137;
b = 55.45237744;
[East, North, Up] = ecef2enu(Xk, Yk, Zk, b, a, H,
wgs84Ellipsoid);
R = sqrt(East^2 + North^2 + Up^2);
Thetal(i) = rad2deg(-asin(Up/R))+90;
phil(i) = atan2(East, North);
t=t+1;
end
%% Графики
figure (1)
[x_sphere, y_sphere, z_sphere] = sphere(20);
x_Earth=a*x_sphere;
y_Earth=a*y_sphere;
z_Earth=a*z_sphere;
surf(x_Earth,y_Earth,z_Earth); hold on;
plot3(Kx(1,:), Ky(1,:), Kz(1,:));
xlabel('x');
ylabel('y');
zlabel('z');
figure (2)
surf(x_Earth,y_Earth,z_Earth); hold on;
plot3(Kx1(1,:), Ky1(1,:), Kz1(1,:));
xlabel('x');
ylabel('y');
zlabel('z');
m = 1;
figure(3)
for y = 1:length(Thetal);
if Thetal(y) <= 90;
Theta2(m) = Thetal(y);
phi2(m) = phil(y);
m = m+1;
end
end
polar(- phi2, Theta2);
camroll(90);
grid on;

```

## СПИСОК ЛИТЕРАТУРЫ

1. Электронный ресурс: [www.glonass-iac.ru](http://www.glonass-iac.ru) - информационно-аналитический центр координатно - временного и навигационного обеспечения.
2. Сайт: <https://www.celestrak.com> - определение формы орбиты и положения спутника на ней.
3. Интернет-сервис: <https://www.gnssplanningonline.com> – программа, моделирующая положение спутников и их характеристики над заданной точкой земной поверхности.
4. BeiDou Navigation Satellite System. System In Space. Interface Control Document.

## ОГЛАВЛЕНИЕ

Постановка задачи.....	2
Этап 1. Использование сторонних средств.....	4
Этап 2. Моделирование.....	11
Этап 3. Реализация.....	13
Заключение.....	18
Приложение.....	19
Список литературы.....	21