**Национальный исследовательский университет**

**«МЭИ»**

**Институт радиотехники и электроники**

**Кафедра радиотехнических систем**

Курсовой проект

по дисциплине

Аппаратура потребителей спутниковых радионавигационных систем

«Разработка модуля расчёта координат спутника ГЛОНАСС»

ФИО студента: Юрьев Д.С.

Группа: ЭР-15-15

Вариант №:22

Дата:­ ­

Подпись:­ ­

ФИО преподавателя: Корогодин И.В.

Оценка: ­ ­

**Москва, 2020**

Содержание

[Введение 3](#_Toc43300160)

[1 Использование сторонних средств 3](#_Toc43300161)

[1.1 Описание процесса использования RTKLIB 3](#_Toc43300162)

[1.2 Получение графика угла места и SkyView с помощью Trimble GNSS Planning 6](#_Toc43300163)

[2. Моделирование 8](#_Toc43300164)

[2.1 Результаты моделирования в Matlab 9](#_Toc43300165)

[2.2 Построение SkyView в Matlab 10](#_Toc43300166)

[Заключение 11](#_Toc43300167)

# **Введение**

Техническая цель проекта - добавление в программное обеспечение приемника функции расчета положения спутника ГЛОНАСС на заданное время по данным его эфемерид.

Конечная цель проекта - получить библиотечные функции на С++, позволяющие рассчитывать положение спутника ГЛОНАСС по эфемеридам.

Для достижения цели выполняется ряд задач:

* обработка данных от приемника ГНСС в RTKLIB для проверки входных данных и формирования проверочных значений;
* обработка данных и моделирование в Matlab/Python для эскизного проектирования модуля;
* реализация программного модуля на С/С++, включая юнит-тестирование в Check.

Требования:

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

Курсовой проект разбит на три этапа, отличающиеся осваиваемыми инструментами.

# **1 Использование сторонних средств**

# **1.1 Описание процесса использования RTKLIB**

На крыше корпуса Е МЭИ установлена трехдиапазонная антенна Harxon HX-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.

Приемники осуществляют первичную обработку сигналов, выдавая по интерфейсам соответствующие потоки данных - наблюдения псевдодальностей и эфемериды спутников.

Необходимо обрабатывать данные от приемника Clonicus, представленные в бинарном виде в формате NVS BINR. Для этого воспользуемся пакетом RTKLIB, в состав которого входит парсер формата NVS BINR и удобные средства отображения данных.

При запуске программы RTKLIB получаем следующее окно (Рисунок 1):



Рисунок 1 – Окно программы RTKLIB v.2.4.2

В окне программы RTKLIB выбираем RTKCONV (Рисунок 2), чтобы конвертировать бинарный файл BINR.bin в текстовый формат NVS BINR.

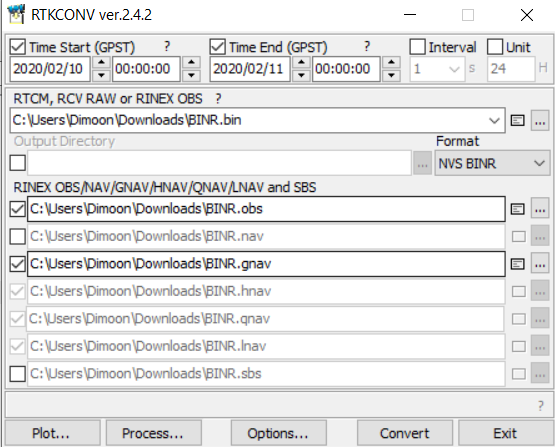


Рисунок 2 – Окно программы RTKCONV ver.2.4.2

В открывшемся окне выбираем Time Start (GPST), Time End (GPST), и ставим время интервала наблюдений с 00:00 10.02.20 до 00:00 11.02.20. В меню «Options» (Рисунок 3) выбираем спутниковую систему ГЛОНАСС и указываем в поле «Excluded Satellite» следующее: R3, R4, R5, R11, R12, R13, R14, R21, R23, тем самым исключая данные спутники из обработки. В первой строке RTKCONV указываем путь на файл бинарного потока .bin, указываем формат NVS BINR, и ставим галочки для конвертации файлов в форматы .obs и .gnav.

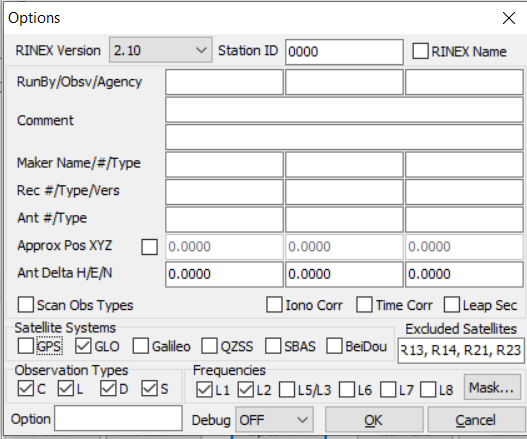


Рисунок 3 – Окно настроек программы RTKCONV ver.2.4.2

Затем нажимаем на кнопку «Convert» и получаем необходимые файлы. Для того, чтобы посмотреть содержимое открываем файл с расширением «gnav» и получаем эфемериды собственного спутника в gnav-файле RINEX (Рисунок 4).

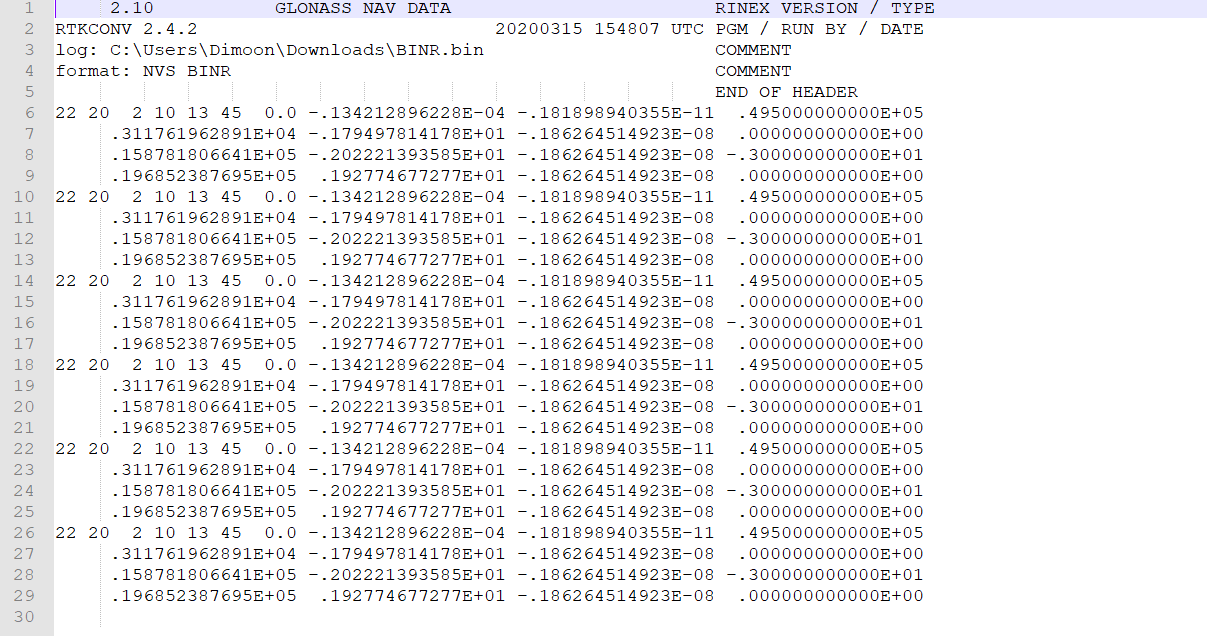


Рисунок 4 – Эфемериды спутника ГЛОНАСС №22 в .gnav файле

Для того, чтобы получить эфемериды собственного спутника по данным RTKNAVI из состава RTKLIB, в окне программы RTKLIB выбираем программу RTKNAVI, которая позволяет вывести таблицу текущих и предыдущих эфемерид (Рисунок 5).

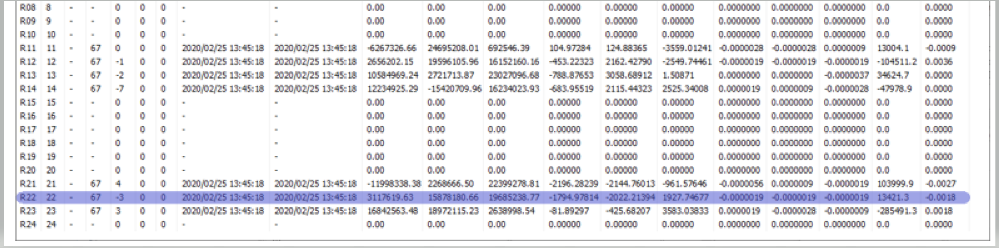


Рисунок 5 – Окно программы RTKNAVI ver.2.4.2

# **1.2 Получение графика угла места и SkyView с помощью Trimble GNSS Planning**

Для построения графика угла места собственного спутника от времени по данным Trimble GNSS Planning Online на заданный интервал времени, во вкладке настроек (Settings) указываем координаты корпуса Е МЭИ и время, когда производились замеры (Рисунок 6). Во вкладке библиотеки спутников (Satellite Library) отключаем отображение всех спутников, кроме заданного (Рисунок 7).

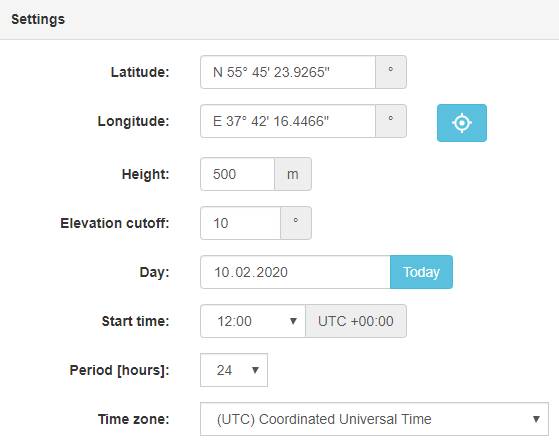


Рисунок 6 – Вкладка настроек (Settings) Trimble GNSS Planning

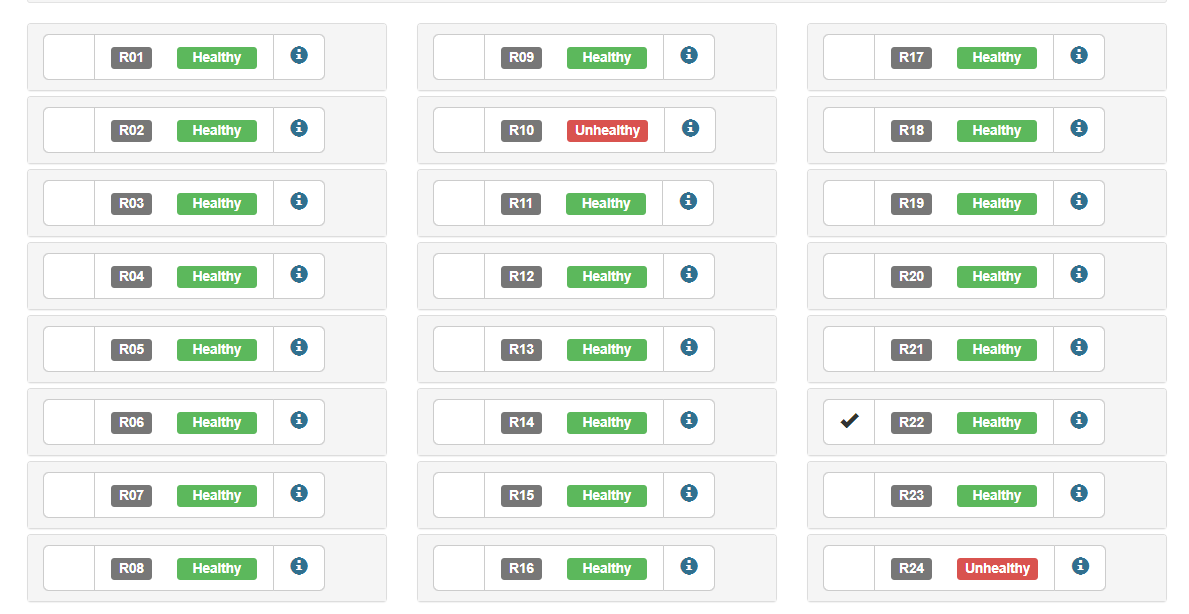


Рисунок 7 – Вкладка библиотеки спутников (Satellite Library) Trimble GNSS Planning

Далее переходим во вкладку «Charts» и получаем график угла места спутника (Рисунок 8). Полученные данные говорят нам о том, что спутник было видно 1 раз с 12:00 до 15:40.

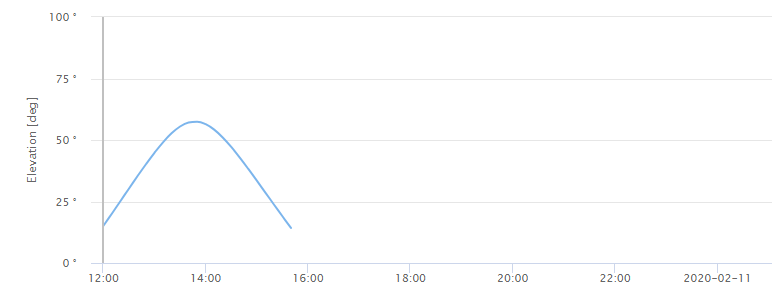


Рисунок 8 – График угла места спутника ГЛОНАСС №22

Перейдя во вкладку «Sky Plot», получаем карту небосвода (SkyView). Траектория движения спутника (Рисунок 9) соответствует его появлению.

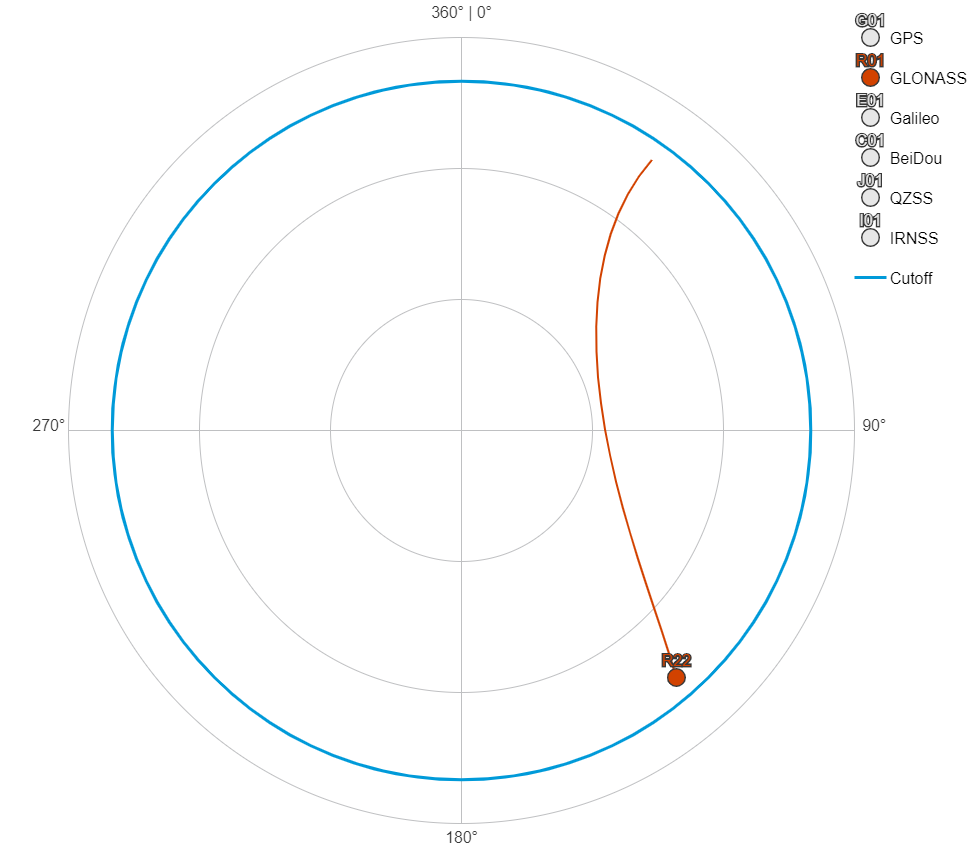


Рисунок 9 – SkyView спутника ГЛОНАСС №22

# **2. Моделирование**

На предыдущем этапе получено решение навигационной задачи с помощью программы вторичной обработки измерений – RTKLIB. В процессе работы она рассчитывает положение спутников на соответствующий момент сигнального времени. При этом используются эфемериды - параметры некоторой модели движения спутника. В разных ГНСС эти модели разные, а значит отличается и формат эфемерид, и алгоритмы расчета положения спутника.

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

Для расчета положения спутника ГЛОНАСС по эфемеридным данным системы проводят численное интегрирование дифференциального уравнения.

Эфемериды, полученные на предыдущем этапе, будем брать из рисунка 5, так как они записаны в удобном варианте.

Необходимо построить трехмерные графики множества положений спутника №22 ГЛОНАСС. Графики в двух вариантах: в СК ECEF ПЗ-90.11 и соответствующей ей инерциальной СК. Положения должны соответствовать временному интервалу с 12:00 10.02.20 до 00:00 11.02.20. Допускается использовать одни и те же эфемериды на весь рассматриваемый интервал. Листинг программы моделирования приведен Приложении 1.

# **2.1 Результаты моделирования в Matlab**

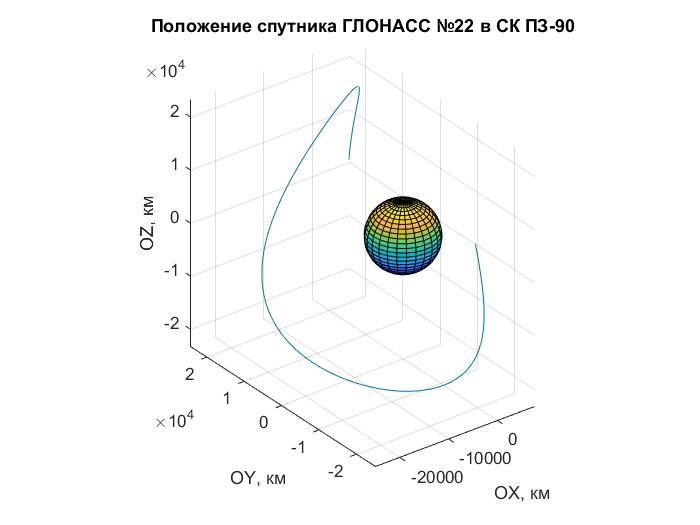


Рисунок 10 – Траектория движения спутника ГЛОНАСС №22 в системе координат ПЗ-90

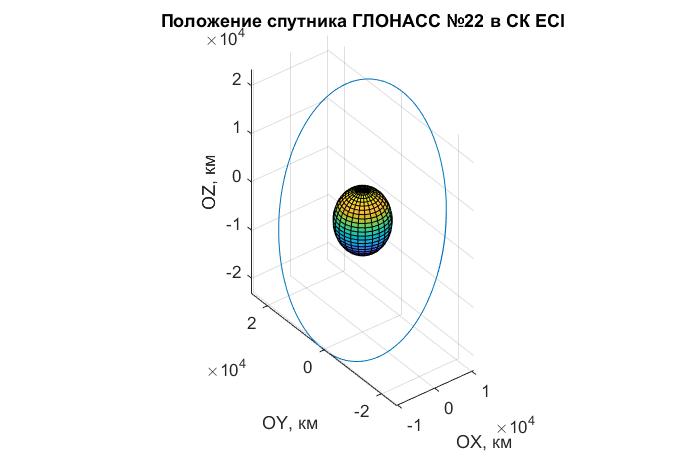


Рисунок 11 – Траектория движения спутника ГЛОНАСС №22 в инерциальной системе координат

# **2.2 Построение SkyView в Matlab**

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

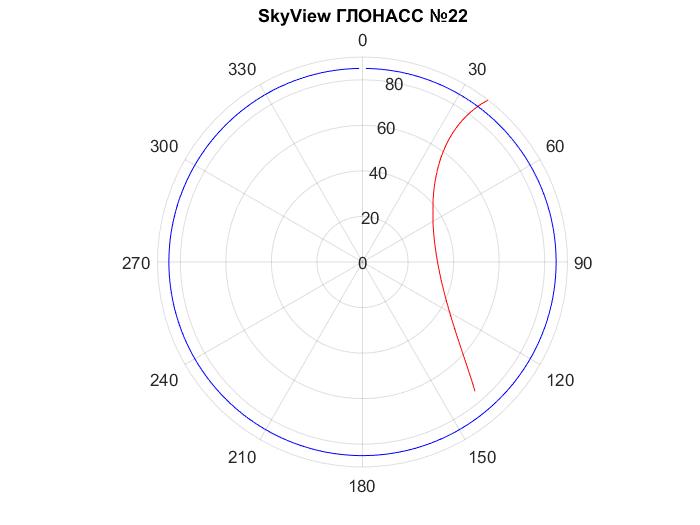


Рисунок 12 – SkyView спутника ГЛОНАСС №22

# **3. Реализация**

Требуется разработать на языке С/С++ функцию расчета положения спутника ГЛОНАСС №22 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.

# **3.1 Результат расчетов**

На рисунке 13 приведены результаты расчетов положения спутника ГЛОНАСС №22. Среднее время выполнения составляет 41.667 [мс]. Максимальная разница координат примерно 202 [км]. Также был получен номер отсчета с максимальной разницей координат. Код программы указан п Приложении 2.

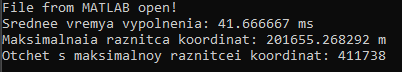


Рисунок 13 – Результат расчетов

# **Заключение**

С помощью RTKLIB были получены эфемериды спутника ГЛОНАСС №22 на интервале наблюдений с 12:00 10.02.20 до 00:00 11.02.20. Эфемериды получены в двух форматах: в виде таблицы текущих и предыдущих эфемерид, в которую входит 11 значений и текстового RINEX файла, в котором эти значения перечислены через пробел. Вторым удобно воспользоваться в модели на следующем этапе, первым - для проверки вручную.

Получены графики SkyPlot с помощью Trimble GNSS Planning Online. За время наблюдения спутник ГЛОНАСС №22 появляется один раз. Пролет наблюдается на интервале времени с 12:00 до 15:40. В момент времени 13:50 угол возвышения максимальный и составляет 57.46º. С помощью этих данных можно будет проверить работу алгоритма расчета положения НС.

В ходе второго этапа с использованием программы MATLAB R2017a, был реализован алгоритм расчета положения спутника ГЛОНАСС №22 на интервале времени с 12:00 10.02.2020 до 00:00 11.02.2020. В качестве исходных данных для алгоритма использовались полученные на первом этапе эфемериды. Получены графики траектории движения спутника ГЛОНАСС №22 в системе координат ПЗ-90 (рисунок 10) и инерциальной системе координат (рисунок 11), а также изображение в SkyView (рисунок 12). Полученное в ходе моделирования изображение спутника ГЛОНАСС №22 в SkyView, совпадает с изображением, полученным при помощи сервиса Trimble GNSS Planning Online.

В ходе третьего этапа была разработана функция расчета положения спутника ГЛОАНАСС №22. Были проведены все необходимые тесты и получены результаты расчета, а именно: среднее время выполнения, максимальная разница и номер отсчета с максимальной разницей координат. Полученные результаты расчета можно объяснить особенностью реализации метода Рунге-Кутты в среде С++.

В ходе выполнения курсового проекта были получены эфемериды спутника ГЛОНАСС №22 на заданный промежуток времени и его изображение в SkyView для последующей проверки результатов моделирования. Получен навык работы с пакетом RTKLIB, предназначенный для получения необходимых эфемерид. По полученным эфемеридам была реализована модель положения спутника в нескольких системах координат и изображение SkyView. Был улучшен навык по работе с программной средой Matlab. Заключительном этапом было написание в среде С++ функции для получения результатов расчета положения спутника.

# **Использованные ресурсы**

1. www.gnssplanningonline.com

2. Пакет RTKLIB.

3. ИКД ГЛОНАСС.

**ПРИЛОЖЕНИЕ 1**

Листинг программы для этапа 2

clear all;

close all;

tic;

% Константы

dt = 1.0E-01; % шаг изменения времени, с

w\_e = 7.292115E-05; % угловая скорость вращения Земли

mu = 3.986004418E+14; % константа гравитационного поля Земли

R\_z = 6371; % радиус Земли, км

% Эфемериды

T\_eph = 49500 + 18 + 3\*3600; % время задания эфемерид

% Координаты

X = .311761962891E+07;

Y = .158781806641E+08;

Z = .196852387695E+08;

% Составляющие скорости

Vx = -.179497814178E+04;

Vy = -.202221393585E+04;

Vz = .192774677277E+04;

% Составляющие ускорения от прочих небесных тел (Луна и Солнце)

Ax = -.186264514923E-05;

Ay = -.186264514923E-05;

Az = -.186264514923E-05;

coordpotr = [55.756657, 37.703288 190]; % координаты потребителя 55°45'24.0"N 37°42'11.8"E

n = fix(12\*3600/dt); % Количество отсчетов за 12 часовой интервал расчета

n\_eph = fix((T\_eph-(12+3)\*3600)/dt); % номер отсчета с эфемеридными данными

t\_12h = (12+3)\*60\*60 + dt.\*(1:1:n); % Вектор отсчетов времени

coordPZ90 = zeros(n,3); % Вектора координат с нулевым заполнением

coordECI = zeros(n,3);

t\_G0 = 9\*3600+18\*60+10.5009; %9:18:10.5009; истинное звездное время в гринвичскую полночь даты задания tэ

t\_G = t\_G0 + w\_e\*(T\_eph - 3\*3600);

% Пересчет координат из ПЗ-90 в ECI

Xa = X\*cos(t\_G) - Y\*sin(t\_G);

Ya = X\*sin(t\_G) + Y\*cos(t\_G);

Za = Z;

Vxa = Vx\*cos(t\_G) - Vy\*sin(t\_G) - w\_e\*Ya;

Vya = Vx\*sin(t\_G) + Vy\*cos(t\_G) + w\_e\*Xa;

Vza = Vz;

Yn = [Xa Ya Za Vxa Vya Vza];

% Интегрирование методом Рунге-Кутты

% Расчет координат от начала диапазона до момента времени эфемерид

[t, Yn1] = ode45('difury', T\_eph:-dt:t\_12h(1), Yn);

coordECI(1:n\_eph,:) = Yn1(end:-1:1,1:3);

% Расчет координат от момента времени эфемерид до конца исследуемого

% диапазона

[t, Yn1] = ode45('difury', T\_eph:dt:t\_12h(n), Yn);

coordECI(n\_eph:end,:) = Yn1(1:end,1:3);

% Пересчет полученных координат из ECI в ПЗ-90

for i = 1:n

t\_G = t\_G0 + w\_e\*(t\_12h(i)- 3\*3600);

cos\_t\_G = cos(t\_G);

sin\_t\_G = sin(t\_G);

coordPZ90(i,1) = coordECI(i,1)\*cos\_t\_G + coordECI(i,2)\*sin\_t\_G;

coordPZ90(i,2) = -coordECI(i,1)\*sin\_t\_G + coordECI(i,2)\*cos\_t\_G;

coordPZ90(i,3) = coordECI(i,3);

end

% Пересчет координат из ПЗ-90 в WGS84 (для получения SkyView)

ppb = 1e-9;

mas = 1e-3/206264.8; % [рад]

M\_WGS84 = [-3\*ppb -353\*mas -4\*mas;

353\*mas -3\*ppb 19\*mas;

4\*mas -19\*mas -3\*ppb];

coordWGS84 = coordPZ90.'; % Переход к вектору-столбцу

for i = 1:length(coordWGS84(1,:))

coordWGS84(:,i) = coordWGS84(:,i) + M\_WGS84 \* coordWGS84(:,i) + [0.07; -0; -0.77];

end

coordWGS84 = coordWGS84.'; % Переход к вектору-строке

% Пересчет координат из WGS84 в SkyView

X = zeros(n,1);

Y = zeros(n,1);

Z = zeros(n,1);

r = zeros(n,1);

teta = zeros(n,1);

phi = zeros(n,1);

for i = 1:length(coordWGS84(:,1))

[X(i), Y(i), Z(i)] = ecef2enu(coordWGS84(i,1),coordWGS84(i,2),coordWGS84(i,3),coordpotr(1),coordpotr(2),coordpotr(3),wgs84Ellipsoid);

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

% Построение графиков

% Расчет сферы, изображения земли

[X\_sf, Y\_sf, Z\_sf] = sphere(25);

% Окружность, для изображения угла отсечки

cutoff\_grads = pi/180.\*(1:359)';

cutoff\_angle = 85.\*ones(359,1);

% График координат в ПЗ-90

plot3(coordPZ90(:,1)/1000, coordPZ90(:,2)/1000, coordPZ90(:,3)/1000);

hold on

grid on

title('Положение спутника в СК ПЗ-90');

xlabel('OX, км');

ylabel('OY, км');

zlabel('OZ, км');

axis('square');

axis('equal');

surf(X\_sf\*R\_z, Y\_sf\*R\_z, Z\_sf\*R\_z);

hold off

% График координат в ECI

figure;

plot3(coordECI(:,1)/1000, coordECI(:,2)/1000, coordECI(:,3)/1000);

hold on

grid on

title('Положение спутника в СК ECI');

xlabel('OX, км');

ylabel('OY, км');

zlabel('OZ, км');

axis('square');

axis('equal');

surf(X\_sf\*R\_z, Y\_sf\*R\_z, Z\_sf\*R\_z);

hold off

% SkyView спутника (угол-место) относительно корпуса Е

figure;

axes = polaraxes;

hold on

polarplot(axes,phi,teta\*180/pi,'r')

polarplot(axes,cutoff\_grads,cutoff\_angle,'b')

hold off

axes.ThetaDir = 'clockwise';

axes.ThetaZeroLocation = 'top';

title('SkyView ГЛОНАСС №22')

toc;

Листинг подпрограммы difury:

function dy = difury(t, y)

mu = 3.986004418E+14; % конствнтва гравитационного поля Земли

Rz = 6378136; % экваториальный радиус Земли

w\_e = 7.292115E-05; % угловая скорость вращения Земли

C20 = -1082.62575E-06; % коэффициент при второй зональной гармонике разложения геопотенциала в ряд по сферическим функциям

t\_G0 = (9\*3600+18\*60+10.5009+3\*3600);

t\_G = t\_G0 + w\_e\*(t- 3\*3600);

% Ускорения, принимаем постоянными на всем интервале расчета

Ax = -.186264514923E-05;

Ay = -.186264514923E-05;

Az = -.186264514923E-05;

JsumX = Ax\*cos(t\_G)-Ay\*sin(t\_G);

JsumY = Ax\*sin(t\_G)+Ay\*cos(t\_G);

JsumZ = Az;

dy = y(:);

dy(1) = y(4);

dy(2) = y(5);

dy(3) = y(6);

r = sqrt(y(1)\*y(1)+y(2)\*y(2)+y(3)\*y(3));

mu\_ = mu/(r\*r);

x\_ = y(1)/r;

y\_ = y(2)/r;

z\_ = y(3)/r;

ro = Rz/r;

dy(4) = -mu\_\*x\_ + 3/2\*C20\*mu\_\*x\_\*ro^2\*(1-5\*z\_^2) + JsumX;

dy(5) = -mu\_\*y\_ + 3/2\*C20\*mu\_\*y\_\*ro^2\*(1-5\*z\_^2) + JsumY;

dy(6) = -mu\_\*z\_ + 3/2\*C20\*mu\_\*z\_\*ro^2\*(3-5\*z\_^2) + JsumZ;

end

**ПРИЛОЖЕНИЕ 2**

Листинг программы для 3 этапа

Исполняемый файл main.cpp

#include <iostream>

#include <fstream>

#include <include\libglnsvpos\glnsvpos.h>

#include <include\libglnsvpos\rungekutta.h>

#include <ctime>

using namespace std;

// 22 20 2 10 13 45 0.0 -.134212896228E-04 -.181898940355E-11 .495000000000E+05

// .311761962891E+04 -.179497814178E+01 -.186264514923E-08 .000000000000E+00

// .158781806641E+05 -.202221393585E+01 -.186264514923E-08 -.300000000000E+01

// .196852387695E+05 .192774677277E+01 -.186264514923E-08 .000000000000E+00

int **main**()

{

time\_t start, end;

double del\_t = 1E-01;

double max\_del = 0;

int i\_max = 0;

int n = (int) 12\*3600/del\_t;

double \*\*koord\_raschet = new double \* [n];

double \*koord\_file = new double[3];

ofstream out;

out.open("D:\\res\_cpp.txt");

time(&start);

ifstream in("D:\\res\_mat.txt");

if (!in)

{

cout << "ERORR: File from MATLAB not open!" << endl;

} else {

cout << "File from MATLAB open!" << endl;

}

for (int i = 0; i < n; i++)

{

koord\_raschet[i] = new double [6];

}

koordinate\_GLONASS(koord\_raschet);

for (int i = 0; i < n; i++)

{

in >> koord\_file[0] >> koord\_file[1] >> koord\_file[2];

string koord\_str1 = to\_string(koord\_raschet[i][0]);

string koord\_str2 = to\_string(koord\_raschet[i][1]);

string koord\_str3 = to\_string(koord\_raschet[i][2]);

out << koord\_str1 << "\t" << koord\_str2 << "\t" << koord\_str3 << endl;

for (int j = 0; j < 3; j++)

{

if (abs(koord\_raschet[i][j] - koord\_file[j]) > max\_del)

{

max\_del = abs(koord\_raschet[i][j] - koord\_file[j]);

i\_max = i;

}

}

delete [] koord\_raschet[i];

koord\_raschet[i] = nullptr;

}

time(&end);

in.close();

out.close();

delete[] koord\_raschet;

koord\_raschet = nullptr;

delete[] koord\_file;

koord\_file = nullptr;

double seconds = difftime(end, start);

string seconds1 = to\_string(seconds\*1000000/n);

cout << "Srednee vremya vypolnenia: " << seconds1 << " ms" << endl;

string max\_del1 = to\_string(max\_del);

cout << "Maksimalnaia raznitca koordinat: " << max\_del1 << " m" << endl;

string imax = to\_string(i\_max);

cout << "Otchet s maksimalnoy raznitcei koordinat: " <<imax << endl;

}

Файл заголовок

#ifndef GLNSVPOS\_H

#define GLNSVPOS\_H

void **koordinate\_GLONASS**(double \*\*koord\_n);

#endif /\* #ifndef GLNSVPOS\_H \*/

Исполняемый файл

#include <include\libglnsvpos\glnsvpos.h>

#include <include\libglnsvpos\rungekutta.h>

#include <iostream>

#include <cmath>

#include <ostream>

using namespace std;

// 22 20 2 10 13 45 0.0 -.134212896228E-04 -.181898940355E-11 .495000000000E+05

// .311761962891E+04 -.179497814178E+01 -.186264514923E-08 .000000000000E+00

// .158781806641E+05 -.202221393585E+01 -.186264514923E-08 -.300000000000E+01

// .196852387695E+05 .192774677277E+01 -.186264514923E-08 .000000000000E+00

void **koordinate\_GLONASS**(double \*\*koord\_n)

{

double t\_0 = 49500 + 18 + 3\*3600;

double t\_nachalo = (12 + 3)\*3600;

double t\_konez = t\_nachalo + 12\*3600;

double del\_t = 1E-01; //шаг по времени

double omega\_e = 7.292115E-05; // угловая скорость вращения Земли

int num = (int) abs((t\_konez - t\_nachalo)/del\_t);

double \*t = new double [num];

double \*koord\_prom = new double [6];

for (int i = 0; i < num; i++)

{

t[i] = t\_nachalo + i\*del\_t;

}

int num\_eph = (int) (t\_0 - t\_nachalo)/del\_t;

// начальные условия

koord\_prom[0] = .311761962891E+07;

koord\_prom[1] = .158781806641E+08;

koord\_prom[2] = .196852387695E+08;

koord\_prom[3] = -.179497814178E+04;

koord\_prom[4] = -.202221393585E+04;

koord\_prom[5] = .192774677277E+04;

double t\_G0 = (9\*3600+18\*60+10.5009); //9:18:10.5009; Истинное звездное время на гринвичевскую полночь текущей даты

double t\_G = t\_G0 + omega\_e\*(t[num\_eph-1]- 3\*3600);

double cos\_t = cos(t\_G);

double sin\_t = sin(t\_G);

// пересчет координат из ПЗ-90 в ECI

koord\_n[num\_eph-1][0] = koord\_prom[0]\*cos\_t - koord\_prom[1]\*sin\_t;

koord\_n[num\_eph-1][1] = koord\_prom[0]\*sin\_t + koord\_prom[1]\*cos\_t;

koord\_n[num\_eph-1][2] = koord\_prom[2];

koord\_n[num\_eph-1][3] = koord\_prom[3]\*cos\_t - koord\_prom[4]\*sin\_t - omega\_e\*koord\_n[num\_eph-1][1];

koord\_n[num\_eph-1][4] = koord\_prom[3]\*sin\_t + koord\_prom[4]\*cos\_t + omega\_e\*koord\_n[num\_eph-1][0];

koord\_n[num\_eph-1][5] = koord\_prom[5];

delete[] koord\_prom;

koord\_prom = nullptr;

for (int i = num\_eph-1; i > 0; i--)

{

RungeKutta(t[i], t[i-1], koord\_n[i], koord\_n[i-1]);

}

for (int i = num\_eph-1; i < num-1 ; i++)

{

RungeKutta(t[i], t[i+1], koord\_n[i], koord\_n[i+1]);

}

delete[] t;

t = nullptr;

delete[] koord\_prom;

koord\_prom = nullptr;

}

Файл заголовок (РК)

#ifndef RUNGEKUTTA\_H

#define RUNGEKUTTA\_H

void **RungeKutta**(double t0, double tn, double \*koord0, double \*koord);

#endif /\* #ifndef RUNGEKUTTA\_H \*/

Исполняемый файл (РК)

#include <include\libglnsvpos\rungekutta.h>

#include <cmath>

#include <include\libglnsvpos\diffs.h>

#include <iostream>

#include <fstream>

using namespace std;

void **RungeKutta**(double t0, double tn, double \*koord0, double \*koord){

double del\_t = tn-t0;

double \*proizv = new double [6];

double \*K\_1 = new double [6];

double \*K\_2 = new double [6];

double \*K\_3 = new double [6];

double \*K\_4 = new double [6];

double \*vremen = new double [6];

// Rschet K1

diffs(t0, koord0, proizv);

for (int j = 0; j<=5; j++){

K\_1[j] = del\_t\*proizv[j];

vremen[j] = koord0[j]+K\_1[j]/2.0;

}

// Rschet K2

diffs(t0+del\_t/2.0, vremen, proizv);

for (int j = 0; j<=5; j++){

K\_2[j] = del\_t\*proizv[j];

vremen[j] = koord0[j]+K\_2[j]/2.0;

}

// Rschet K3

diffs(t0+del\_t/2.0, vremen, proizv);

for (int j = 0; j<=5; j++){

K\_3[j] = del\_t\*proizv[j];

vremen[j] = koord0[j]+K\_3[j];

}

// Rschet K4

diffs(tn, vremen, proizv);

for (int j = 0; j<=5; j++){

K\_4[j] = del\_t\*proizv[j];

koord[j] = koord0[j] + (K\_1[j]+2\*K\_2[j]+2\*K\_3[j]+K\_4[j])/6.0;

}

delete[] K\_1;

K\_1 = nullptr;

delete[] K\_2;

K\_2 = nullptr;

delete[] K\_3;

K\_3 = nullptr;

delete[] K\_4;

K\_4 = nullptr;

delete[] proizv;

proizv = nullptr;

delete[] vremen;

vremen = nullptr;

}

Исполняемый файл (диф)

#include <cmath>

#include <include\libglnsvpos\diffs.h>

// 22 20 2 10 13 45 0.0 -.134212896228E-04 -.181898940355E-11 .495000000000E+05

// .311761962891E+04 -.179497814178E+01 -.186264514923E-08 .000000000000E+00

// .158781806641E+05 -.202221393585E+01 -.186264514923E-08 -.300000000000E+01

// .196852387695E+05 .192774677277E+01 -.186264514923E-08 .000000000000E+00

void **diffs**(double t, double \*koord, double \*dif)

{

double mu = 3.986004418E+14; // конствнтва гравитационного поля Земли

double R\_z = 6378136; // экваториальный радиус Земли

double omega\_e = 7.292115E-05; // угловая скорость вращения Земли

double C\_20 = -1082.62575E-06;

double t\_g0 = (9\*3600 + 18\*60 + 10.5009);

double t\_g = t\_g0 + omega\_e\*(t - 3\*3600);

// Ускорения, принимаем постоянными на всем интервале расчета

double Ax = -.186264514923E-05;

double Ay = -.186264514923E-05;

double Az = -.186264514923E-05;

double Jsum\_x = Ax\*cos(t\_g) - Ay\*sin(t\_g);

double Jsum\_y = Ax\*sin(t\_g) + Ay\*cos(t\_g);

double Jsum\_z = Az;

dif[0] = koord[3];

dif[1] = koord[4];

dif[2] = koord[5];

double r = sqrt(koord[0]\*koord[0] + koord[1]\*koord[1] + koord[2]\*koord[2]);

double mu\_ = mu/(r\*r);

double x\_ = koord[0]/r;

double y\_ = koord[1]/r;

double z\_ = koord[2]/r;

double ro = R\_z/r;

dif[3] = -mu\_\*x\_ + 3/2\*C\_20\*mu\_\*x\_\*ro\*ro\*(1-5\*z\_\*z\_) + Jsum\_x;

dif[4] = -mu\_\*y\_ + 3/2\*C\_20\*mu\_\*y\_\*ro\*ro\*(1-5\*z\_\*z\_) + Jsum\_y;

dif[5] = -mu\_\*z\_ + 3/2\*C\_20\*mu\_\*z\_\*ro\*ro\*(3-5\*z\_\*z\_) + Jsum\_z;

}

Файл заголовок (диф)

#ifndef DIFFS\_H

#define DIFFS\_H

void **diffs**(double t, double \*koord, double \*proizv);

#endif // DIFFS\_H