**Национальный исследовательский университет**

**«МЭИ»**

**Институт радиотехники и электроники**

**Кафедра радиотехнических систем**

Курсовой проект

по дисциплине

Аппаратура потребителей спутниковых радионавигационных систем

ФИО студента: Кагин И.И.

Группа: ЭР-15-15

Вариант №:11

Дата:­ ­

Подпись:­ ­

ФИО преподавателя: Корогодин И.В.

Оценка: ­ ­

**Москва, 2020**

Содержание

[Введение 3](#_Toc42006037)

[1 Использование сторонних средств 3](#_Toc42006038)

[2 Получение SkyView с помощью Trimble GNSS Planning 5](#_Toc42006039)

[3 График угла места 9](#_Toc42006040)

[4 Моделирование 9](#_Toc42006041)

[5 Реализация 16](#_Toc42006042)

[Заключение 18](#_Toc42006043)

[6 Список использованных источников 19](#_Toc42006044)

[ПРИЛОЖЕНИЕ А 20](#_Toc42006045)

[ПРИЛОЖЕНИЕ Б 25](#_Toc42006046)

# Введение

Название проекта: Разработка модуля расчёта координат спутника ГЛОНАСС. Техническая цель - 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

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.

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

Получим эфемериды спутника по данным RTKNAVI из состава RTKLIB.Программа RTKNAVI позволяет вывести таблицу текущих и предыдущих эфемерид (Рисунок 1).

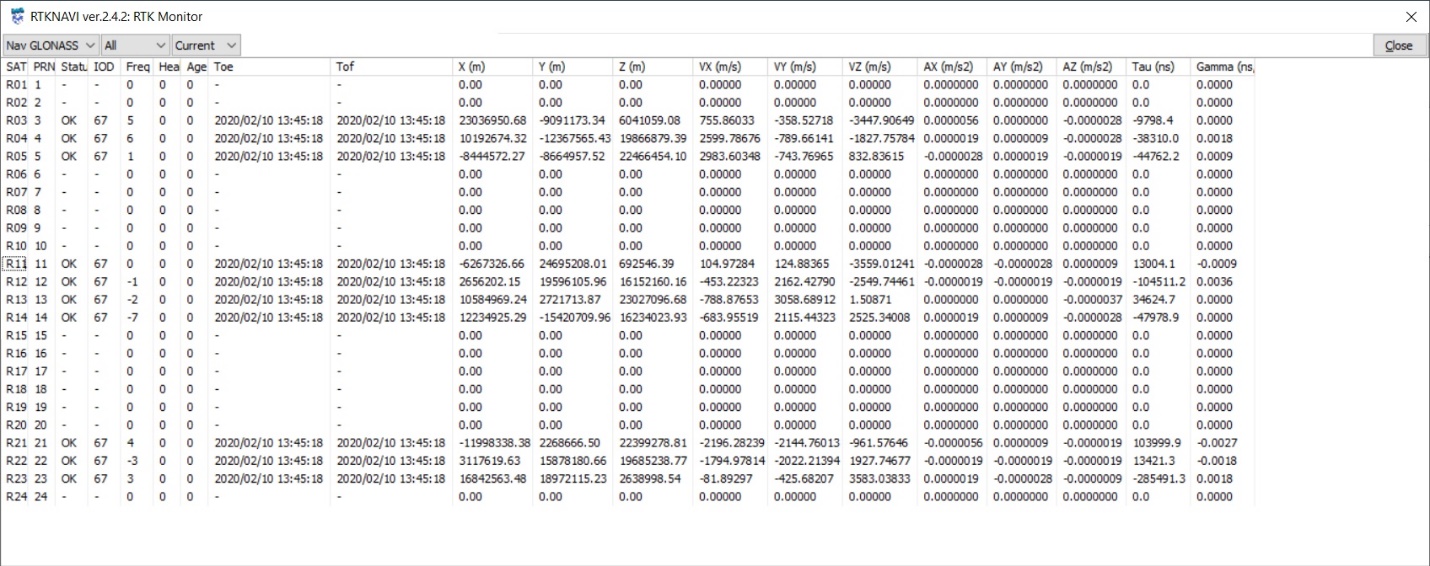


Рисунок 1 – Таблица эфемерид программы RTKNAVI

Далее конвертируем бинарный файл BINR.bin в текстовый формат NVS BINR. Для этого используем программу RTKCONV (Рисунок 2). Программа RTKCONV позволяет конвертировать бинарный файл в текстовый формат RINEX, в частности получить текстовый gnav-файл с эфемеридами ГЛОНАСС.

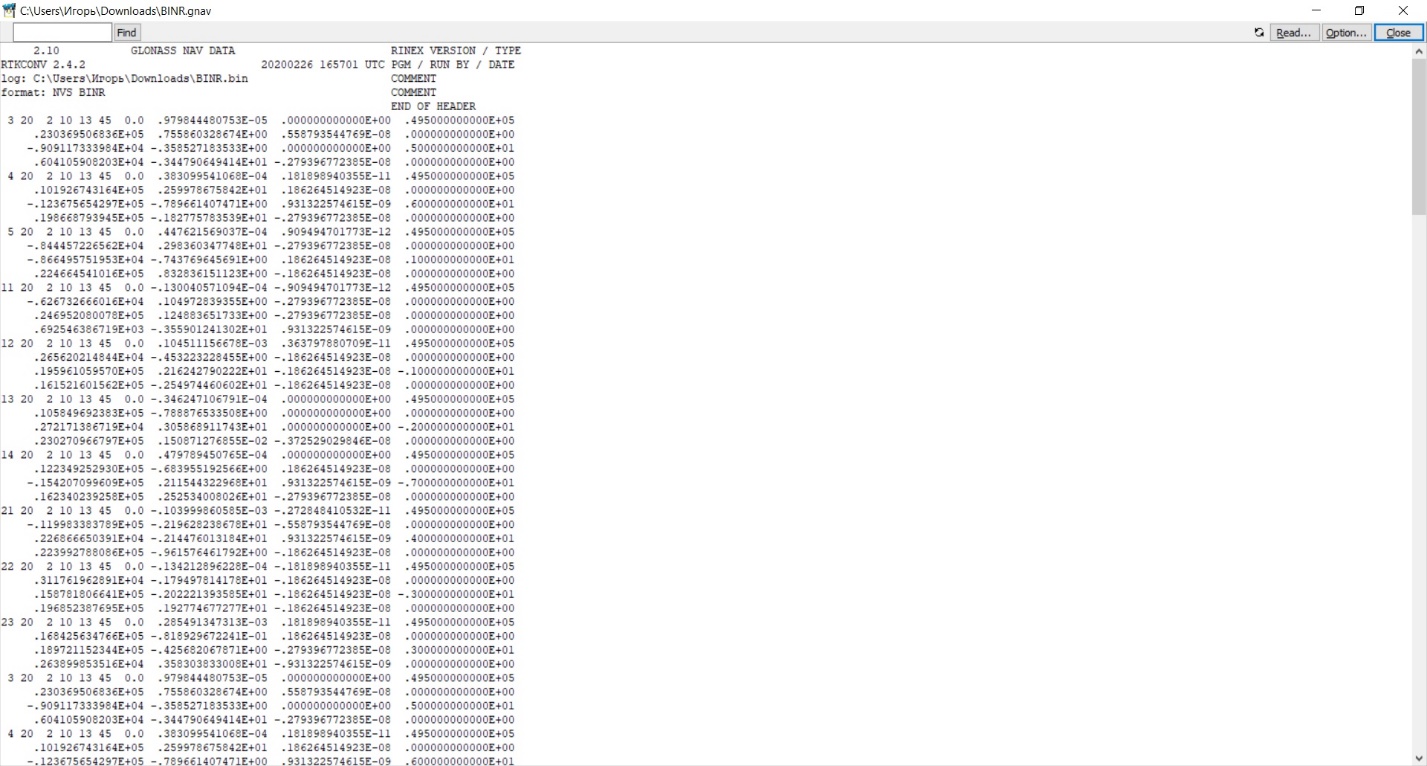


Рисунок 2 – Текстовый gnav-файл с эфемеридами ГЛОНАСС

# 2 Получение SkyView с помощью Trimble GNSS Planning

Построим график угла места собственного спутника от времени по данным Trimble GNSS Planning Online (<https://www.gnssplanning.com>). Для начала указываем координаты места наблюдения (Рисунок 3). Также во вкладке библиотеки спутников (Satellite Library) отключаем отображение всех спутников, кроме заданного.

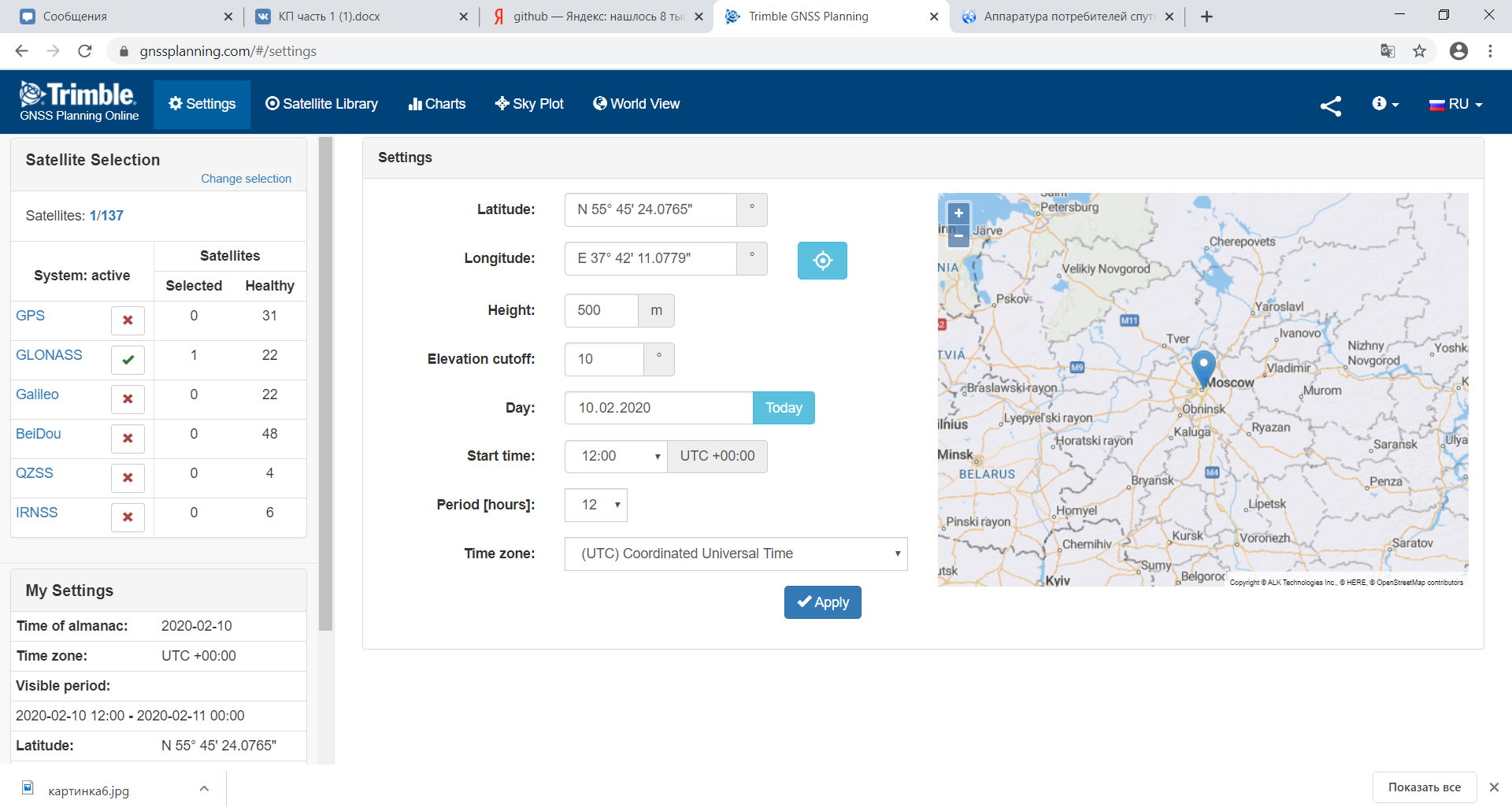


Рисунок 3 – Вкладка настроек Trimble GNSS Planning Online

Перейдя во вкладку «Sky Plot»,получаем карту небосвода. На заданном интервале времени спутник был виден 2 раза. Приведу полученные результаты на рисунках 4 и 5.

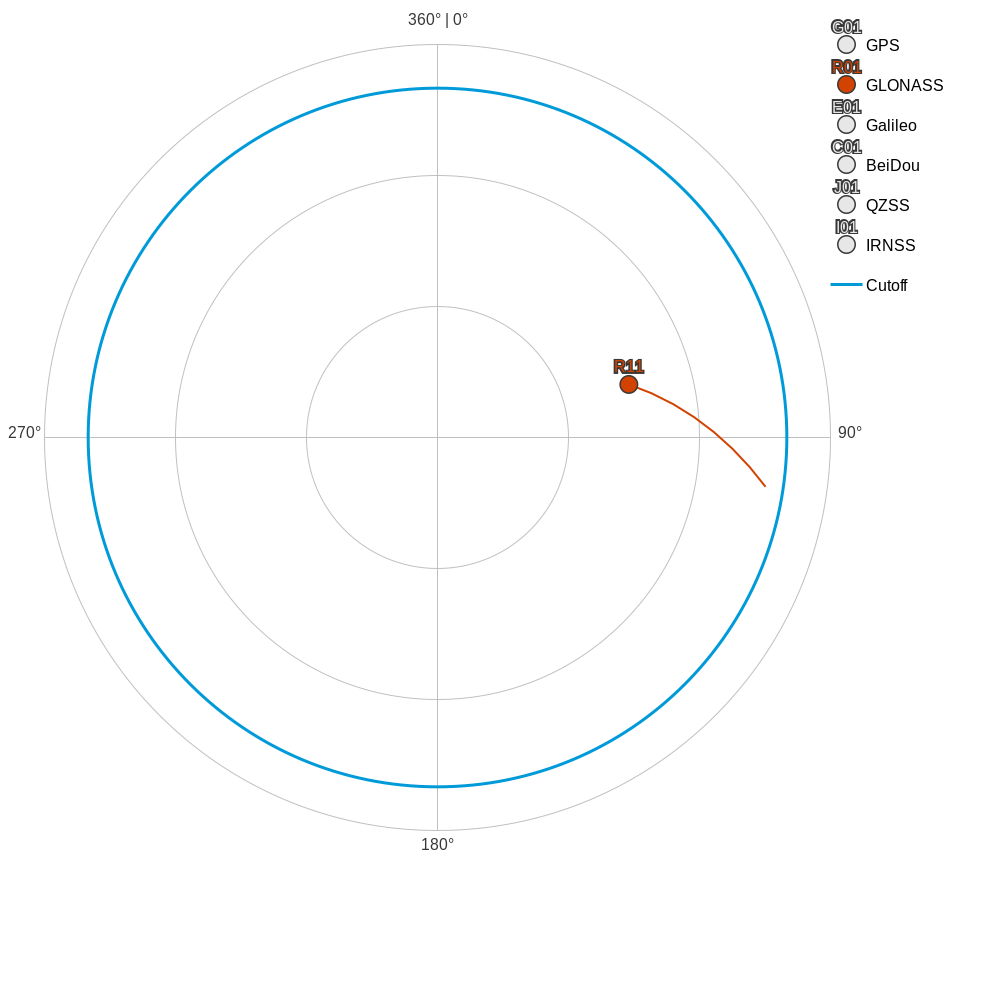


Рисунок 4 – SkyView спутника ГЛОНАСС №11: первый пролёт

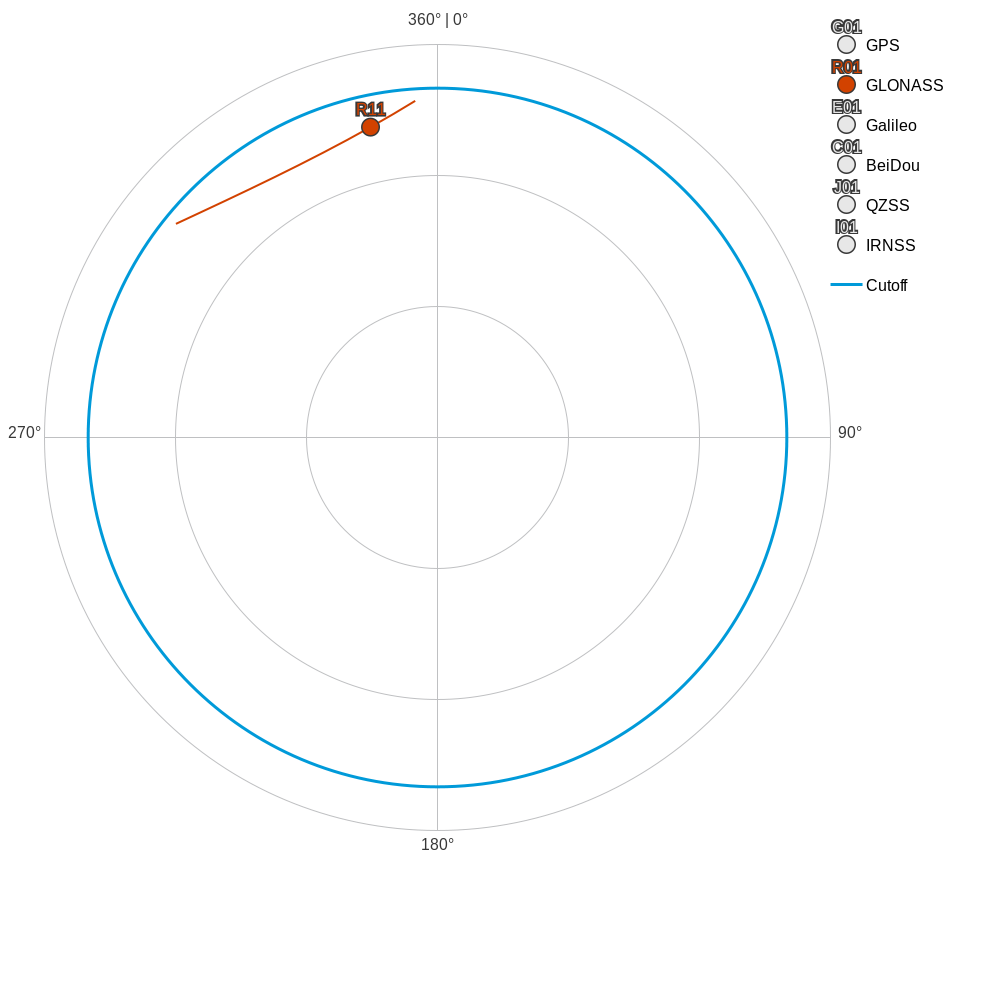


Рисунок 5 – SkyView спутника ГЛОНАСС №11: второй пролёт

# 3 График угла места

Далее во вкладке Charts получаем график угла места (Рисунок 6). Полученные данные подтверждают, что спутник был виден 2 раза. Первое появление с 12:00 до 13:10, второе с 21:50 до 23:30.

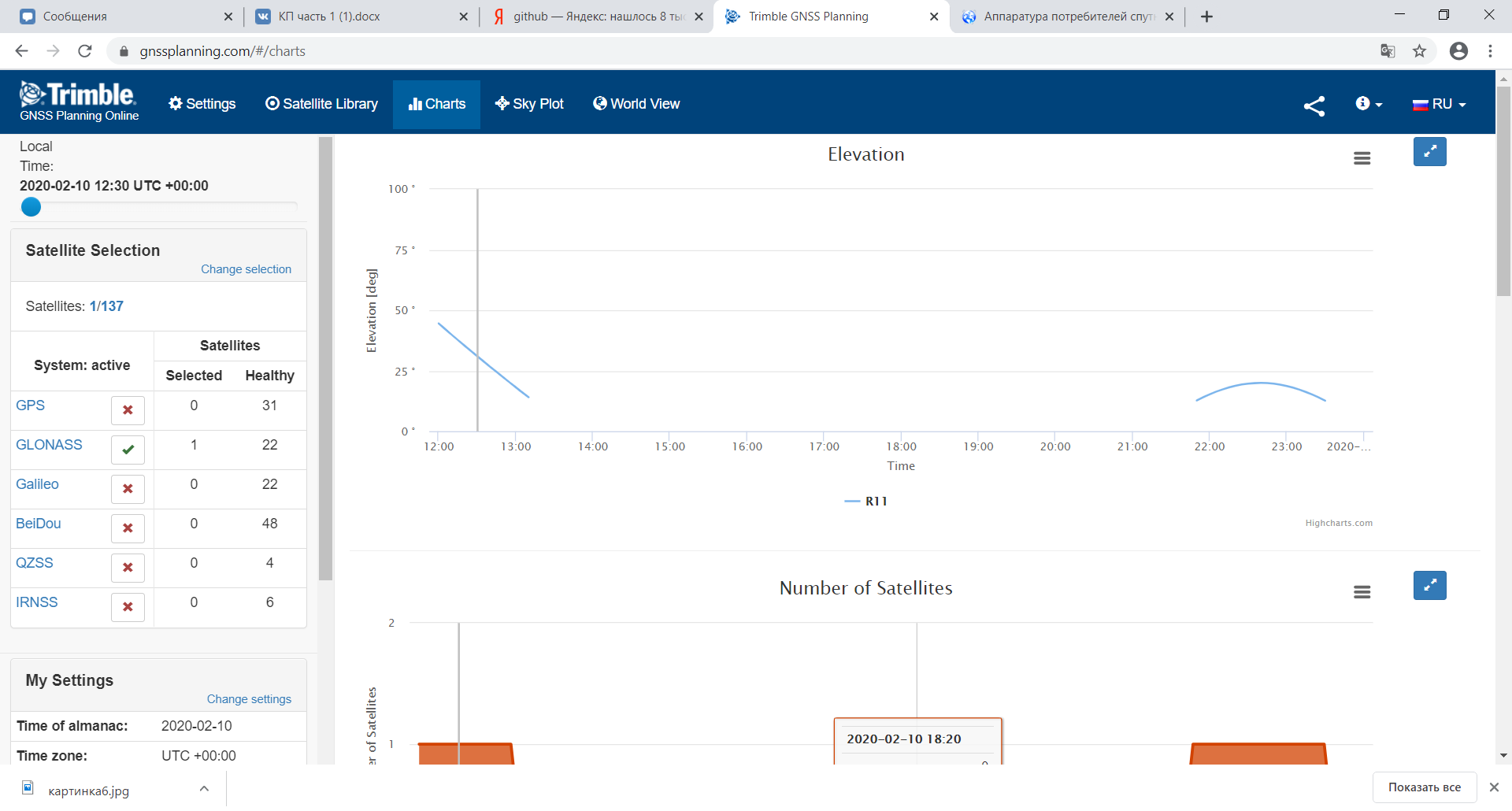


Рисунок 6 – График угла места спутника ГЛОНАСС №11

# 4 Моделирование

Для расчёта положения спутника ГЛОНАСС по эфемеридным данным приводят численное интегрирование дифференциального уравнения. Реализую на языке Matlab функцию расчёта положения спутника ГЛОНАСС на заданный момент времени по шкале UTC. В качестве эфемерид использованы данные, полученные на предыдущем этапе. Предъявляю код программы, реализованный в Matlab в [приложении](file:///C:\Users\Polina\Downloads\Отчёт%20этап%202.docx#приложении).

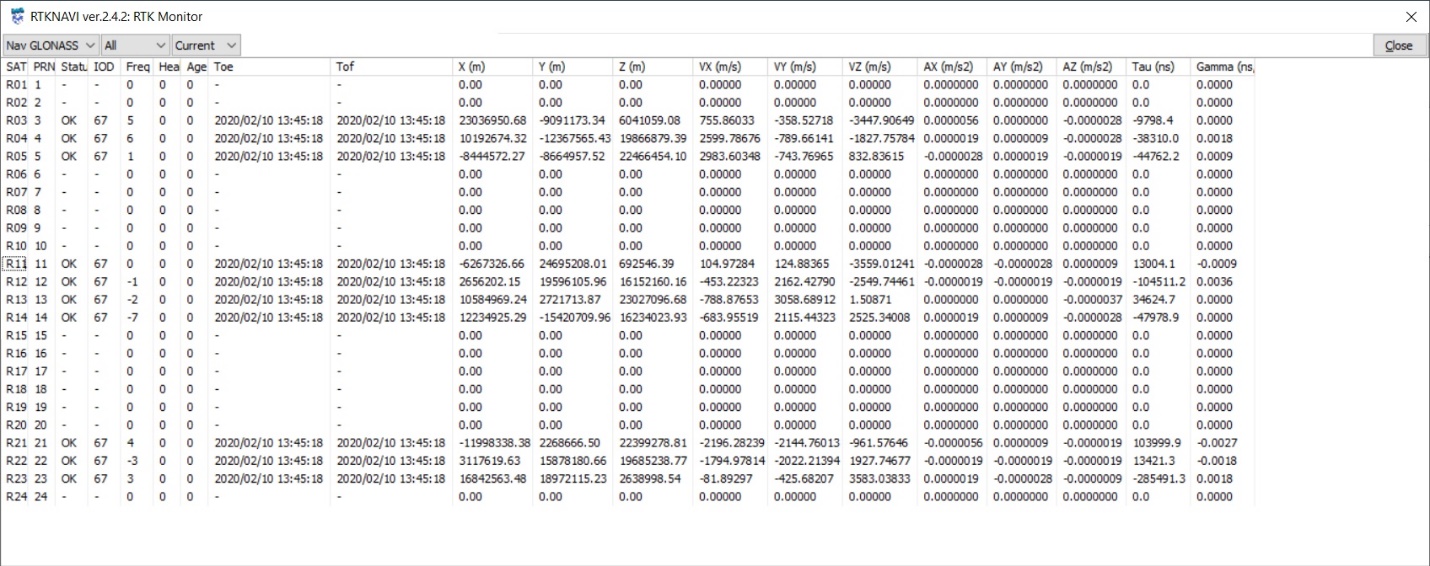


Рисунок 7 – Таблица эфемерид программы RTKNAVI

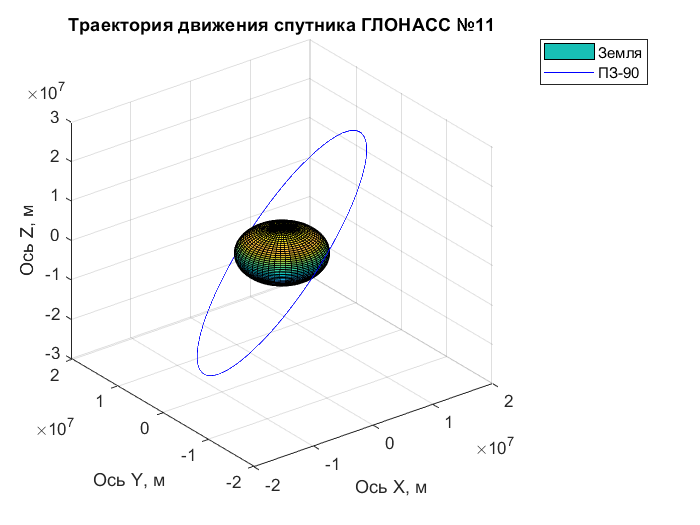


Рисунок 8 – Трехмерный график траектории движения спутника №11

Для расчёта положения спутника ГЛОНАСС по эфемеридным данным проводят численное интегрирование дифференциального уравнения. Приведу полученные результаты на рисунках 9 и 10.

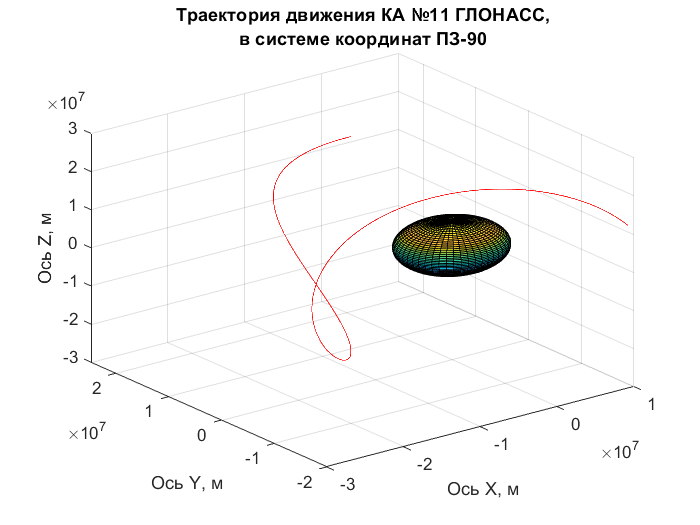


Рисунок 9 – Трехмерный график положений спутника в ECEF

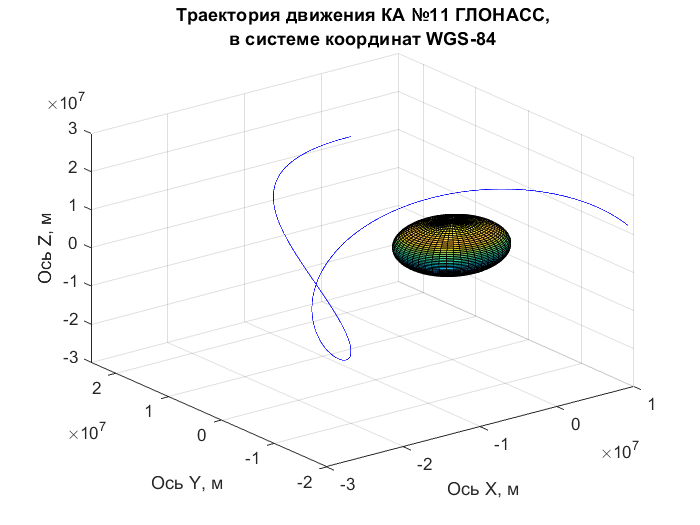


Рисунок 10 – Трехмерный график положений спутника в ECI

Использую Trimble GNSS Planning Online (<https://www.gnssplanning.com>) для получения SkyView. На заданном интервале времени спутник был виден 2 раза. Приведу полученные результаты на рисунках 11 и 12.

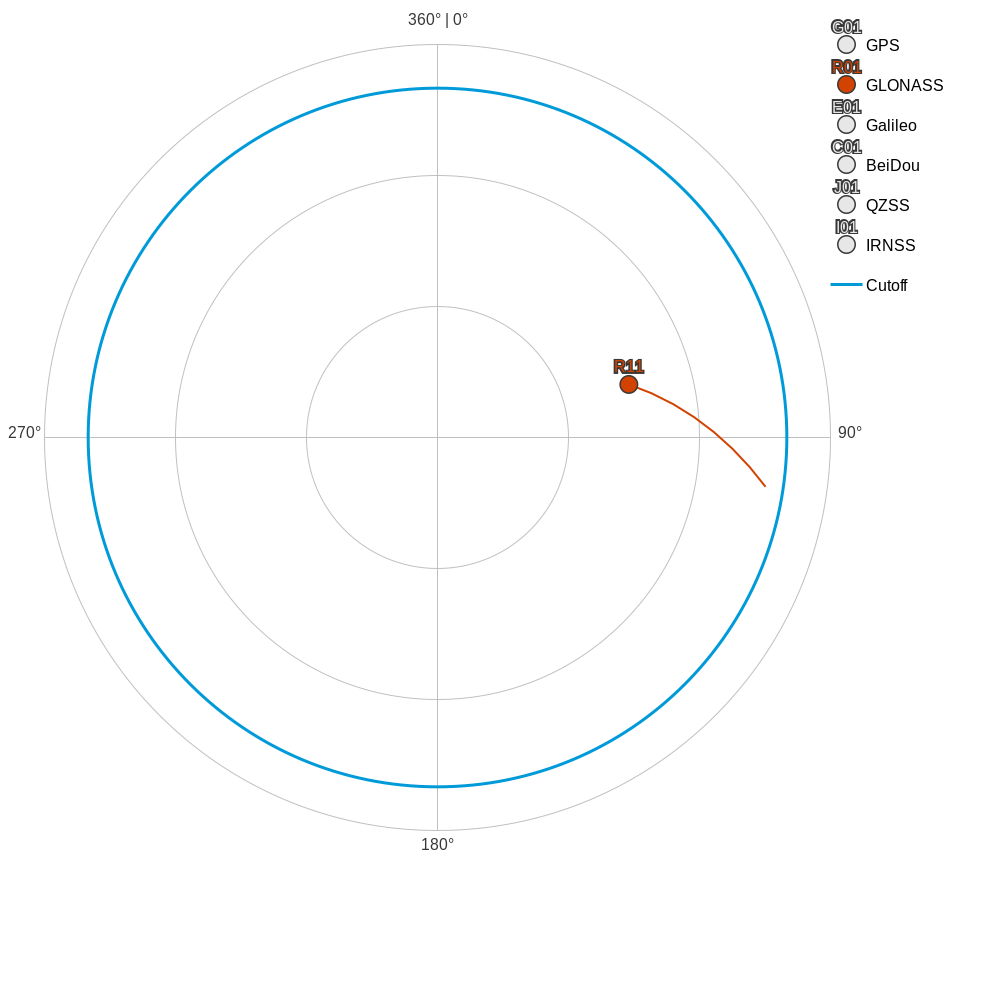


Рисунок 11 – SkyView спутника ГЛОНАСС №11: первый пролёт

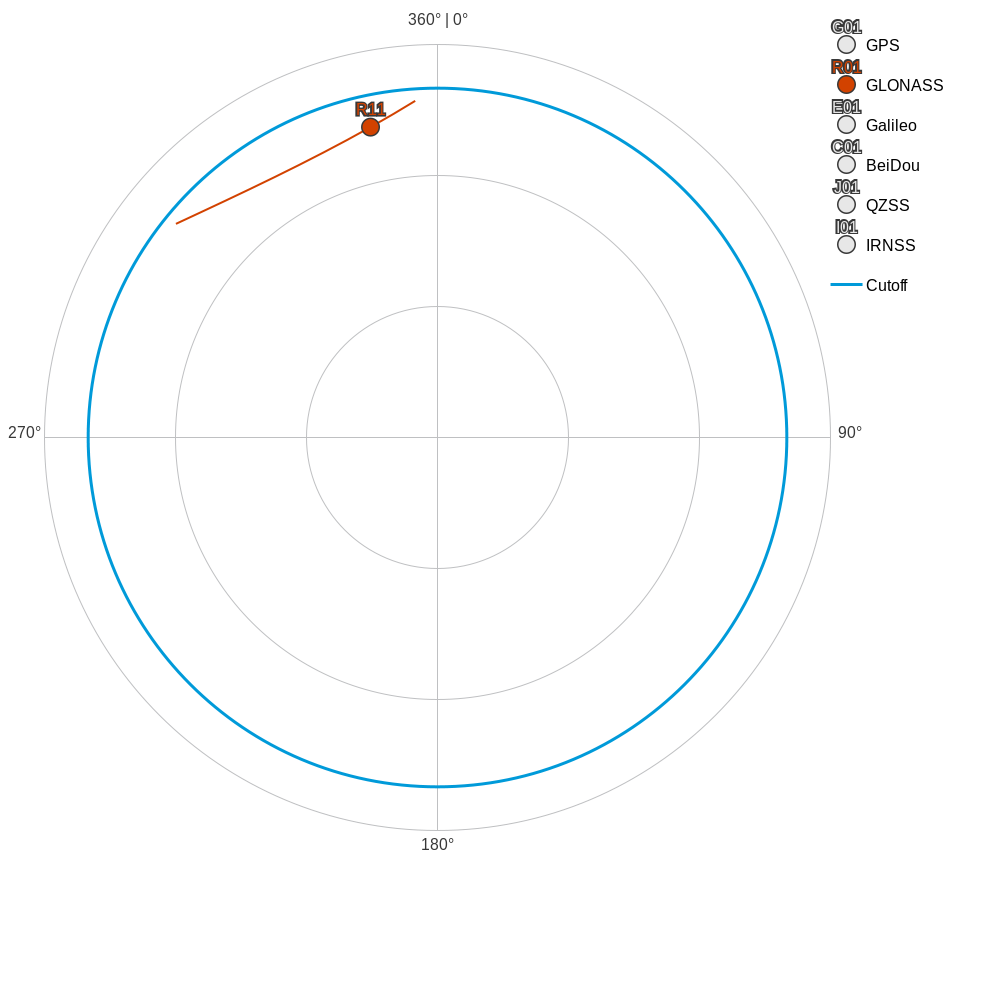


Рисунок 12 – SkyView спутника ГЛОНАСС №11: второй пролёт

Приведу расчётный SkyView спутника ГЛОНАСС №11 на рисунке 8 и расчётный график угла места на рисунке 8.

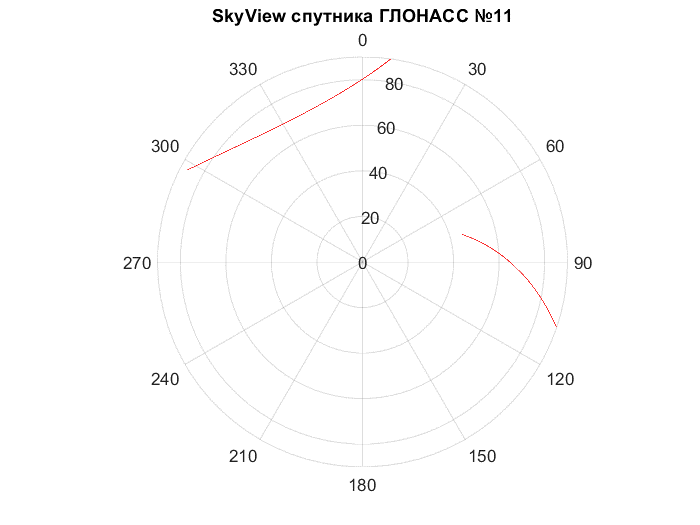


Рисунок 13 – Расчётный SkyView спутника ГЛОНАСС №11

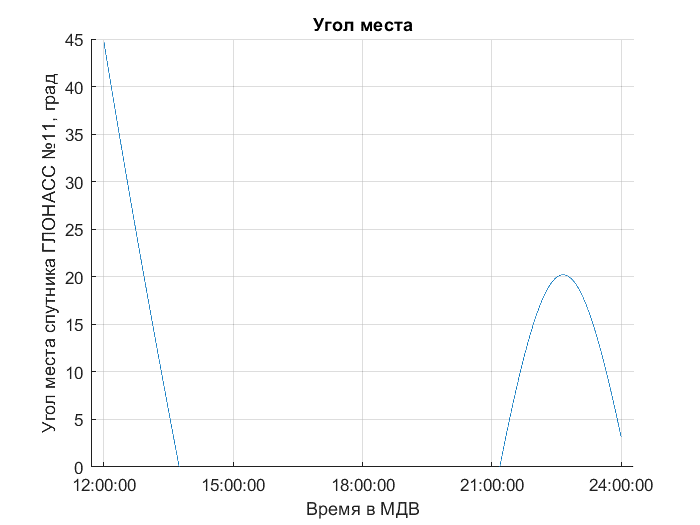


Рисунок 14 – Расчётный угол места спутника ГЛОНАСС №11

# 5 Реализация

Требуется разработать на языке С/С++ функцию расчета положения спутника ГЛОНАСС 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. Допускается использовать одни и те же эфемериды на весь рассматриваемый интервал.

Программный модуль должен сопровождаться unit-тестами:

* Тесты функции реализации метода Рунге-Кутты
* Тест расчетного положения спутника в сравнении с Matlab с шагом 0.1 секунды.

Во время второго теста должно вычисляться и выводиться средняя длительность исполнения функции.

Требуется провести проверку на утечки памяти.

Функция расчета положения спутника в Matlab 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 в прошивку приемника дословно, как правило, не получается. В рассматриваемом примере потребуется, как минимум, выполнить свою реализацию решения дифура методом Рунге-Кутты.

Начнём с реализации программы расчёта положения спутника ГЛОНАСС в Visual Studio, используя тот же алгоритм. Поскольку в среде отсутствуют готовые функции решения дифференциального уравнения методом Рунге-Кутты, необходимо выполнить данный метод самостоятельно. Метод представлен на рисунке 15.

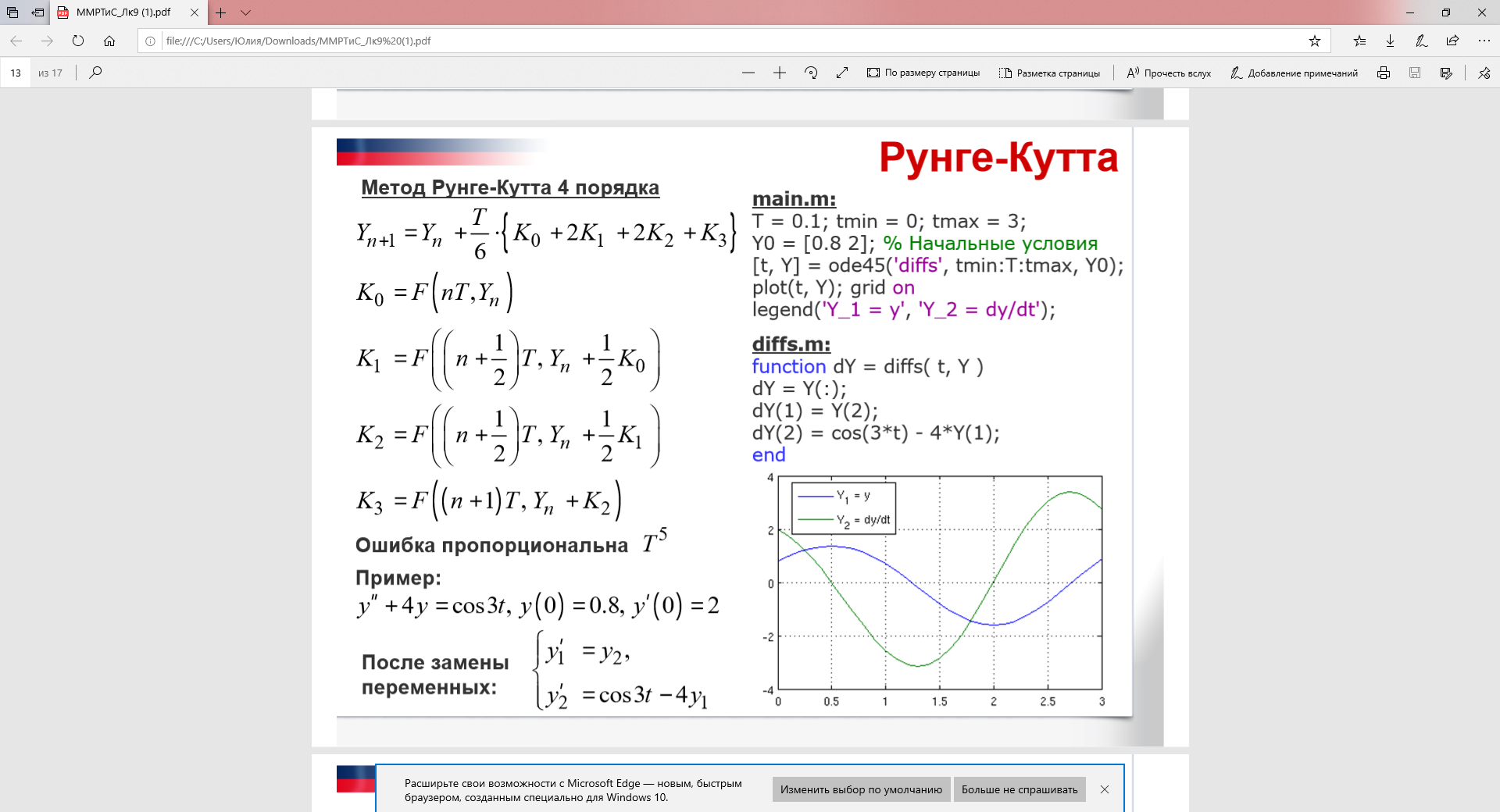


Рисунок 15 – Метод Рунге-Кутта 4-ого порядка

Листинг программы приведён в приложении Б.

# Заключение

В ходе выполнения курсового проекта были получены следующие результаты:

1)Обработаны данные от приёмника ГНСС с помощью приложений из пакета RTKLIB и преобразованы в таблицу эфемериды спутников ГЛОНАСС;

2)Эфемериды собственного спутника в конвертированном gnav-файле RINEX с помощью RTKCONV из пакета RTKLIB;

3)График угла места собственного спутника на заданный интервал времени, а также установлено количество появлений спутника в заданный промежуток времени;

4)SkyView по данным Trimble GNSS Planning Online на заданный интервал времени.

5)Реализация в Matlab расчёта положения спутника ГЛОНАСС по эфемеридным данным;

6)Трёхмерные графики положений спутника в ECEF и ECI;

7)Соответствие расчётного SkyView полученному в Trimble GNSS Planning Online;

8)Расчётный график угла места.

6 Список использованных источников

1. ИКД ГЛОНАСС. Общее описание системы с кодовым разделением.
2. GLONASS Satellite Coordinates Computation – Navipedia.
3. Материалы лекций по курсам: «Математическое моделирование РТУ и С», и «Аппаратура потребителей спутниковых радионавигационных систем».

# ПРИЛОЖЕНИЕ А

%%Данные из эфемерид

xte=-6267326.66;

Y=24695208.01;

zte=692546.39;

vxte=104.97284;

vyte=124.88365;

vzte=-3559.01241;

axte=-0.0000028;

ayte=-0.0000028;

azte=0.0000009;

Year=2020;

Month=2;

Day=25;

Hour=13;

minutes=45;

seconds=18;

gamma=-0.0009;%SV относительное смещение частоты ,ns

tau=13004.1;% SV временное смещение, ns

%Расчёт времени формата ГЛОНАСС

N4=floor((Year-1996)/4)+1;%Номер текущего четырёхлетия

NT=365\*(Year-1996-4\*(N4-1))+31+Day+1;%Номер текущих суток

tb=Hour\*60\*60+minutes\*60+seconds+10800;%момент по шкале МДВ, к которому привязаны эфемериды ГЛОНАСС, в сек

%Расчет среднего звездного времени по Гринвичу

GMST=GMST\_calc(N4,NT);

omega\_e=0.7292115e-4;%earth's rotation rate

thetaGe=GMST+omega\_e\*(tb-3\*60\*60);% the sidereal time at epoch , to which are referred the initial conditions, in Greenwich meridian

%%

%%Пересчёт в инерциальиную систему координат

Xate=xte\*cos(thetaGe)-Y\*sin(thetaGe);

Yate=xte\*sin(thetaGe)+Y\*cos(thetaGe);

Zate=zte;

Vxate=vxte\*cos(thetaGe)-vyte\*sin(thetaGe)-omega\_e\*Yate;

Vyate=vxte\*sin(thetaGe)+vyte\*cos(thetaGe)+omega\_e\*Xate;

Vzate=vzte;

Axte=axte\*cos(thetaGe)-ayte\*sin(thetaGe);

Ayte=axte\*sin(thetaGe)+ayte\*cos(thetaGe);

Azte=azte;

%Метод Рунге-Кутты

toe=(12+3)\*60\*60;

tend=(24+3)\*60\*60;

tp=1;

ti=toe:tp:tend;

Y0=[Xate Yate Zate Vxate Vyate Vzate];

[t, Y] = ode45('diffs',tb:-tp:ti(1) ,Y0);

Y1=Y(end:-1:2,:);

t1=t(end:-1:2,:);

[t,Y]=ode45('diffs',tb:tp:ti(end) ,Y0);

Y1=[Y1;Y];

t1=[t1;t];

% Учет ускорений

tau1 = t1 - tb;

AXTE = Axte\*(tau1.^2)/2;

AYTE = Ayte\*(tau1.^2)/2;

AZTE = Azte\*(tau1.^2)/2;

delta\_Vxate = Axte\*tau1;

delta\_Vyate = Ayte\*tau1;

delta\_Vzate = Azte\*tau1;

delta\_A = [AXTE AYTE AZTE delta\_Vxate delta\_Vyate delta\_Vzate];

Y1 = Y1 + delta\_A;

% Пересчет координат центра масс НКА в систему координат ПЗ-90

thetaGe=GMST+omega\_e\*(t1-3\*60\*60);

PZ90(:,1) = Y1(:,1).\*cos(thetaGe) + Y1(:,2).\*sin(thetaGe);

PZ90(:,2) = -Y1(:,1).\*sin(thetaGe) + Y1(:,2).\*cos(thetaGe);

PZ90(:,3) = Y1(:,3);

%Kooрдинаты корпуса E

%широта 55° 45' 24.0765" ,переводя получим 55.756687916667

N = 55.756687916667\*pi/180 ;% широта [рад]

%долгота 37° 42' 11.0779" переводя в десятичные доли градуса получаем 37.703077194444

E=37.703077194444\*pi/180;% долгота [рад]

H = 500; % высота [м]

geogr\_coorE = [N E H];

%Skyplot

for i = 1:length(PZ90(:,1))

[xte(i) Y2(i) zte(i)] = ecef2enu(PZ90(i,1),PZ90(i,2),PZ90(i,3),N,E,H,wgs84Ellipsoid,'radians');

if zte(i) > 0

r(i) = sqrt(xte(i)^2 + Y2(i)^2 + zte(i)^2);

teta(i) = acos(zte(i)/r(i));%

if xte(i) > 0

phi(i) = -atan(Y2(i)/xte(i))+pi/2;

elseif (xte(i)<0)&&(Y2(i)>0)

phi(i) = -atan(Y2(i)/xte(i))+3\*pi/2;

elseif (xte(i)<0)&&(Y2(i)<0)

phi(i) = -atan(Y2(i)/xte(i))-pi/2;

end

else teta(i) = NaN;

r(i) = NaN;

phi(i) = NaN;

end

end

%% Пересчет в систему координат WGS-84

ppb = 1e-9;

mas = 1e-3/206264.8; % [рад]

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

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

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

WGS84 = PZ90.'; % Переход к вектору-столбцу

for i = 1:length(WGS84(1,:))

WGS84(:,i) = WGS84(:,i) + MATRIX\_WGS84 \* WGS84(:,i) + [0.07; -0; -0.77];

end

WGS84 = WGS84.'; % Переход к вектору-строки

R\_WGS84 = sqrt(WGS84(:,1).^2 + WGS84(:,2).^2 + WGS84(:,3).^2);

%%

%Построение графиков

figure(1)

[X,Y,Z]=sphere(50);

Rz=6371000;%радиус Земли

surf(Rz\*X,Rz\*Y,Rz\*Z)

hold on

grid on

plot3(Y1(:,1), Y1(:,2), Y1(:,3), 'b')

title('Траектория движения спутника ГЛОНАСС №11')

xlabel('Ось Х, м')

ylabel('Ось Y, м')

zlabel('Ось Z, м')

hold off

legend('Земля','ПЗ-90', 'Инерциальная СК');

figure(2)

surf(Rz\*X,Rz\*Y,Rz\*Z)

hold on

grid on

plot3(PZ90(:,1),PZ90(:,2),PZ90(:,3),'r')

title({'Траектория движения КА №11 ГЛОНАСС,' ; 'в системе координат ПЗ-90'})

xlabel('Ось Х, м')

ylabel('Ось Y, м')

zlabel('Ось Z, м')

hold off

%SkyPlot

figure (3)

pax = polaraxes;

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

pax.ThetaDir = 'clockwise';

pax.ThetaZeroLocation = 'top';

title('SkyView спутника ГЛОНАСС №11')

th = hours(t1./3600-3);

figure(4);

grid on

hold on

plot(th,(-teta\*180/pi+90),'DurationTickFormat','hh:mm:ss')

title('Угол места')

xlabel('Время в МДВ')

ylabel('Угол места спутника ГЛОНАСС №11, град')

figure(5)

surf(Rz\*X,Rz\*Y,Rz\*Z)

grid on

hold on

plot3(WGS84(:,1),WGS84(:,2),WGS84(:,3), 'b')

title({'Траектория движения КА №11 ГЛОНАСС,' ; 'в системе координат WGS-84'})

xlabel('Ось Х, м')

ylabel('Ось Y, м')

zlabel('Ось Z, м')

hold off

# ПРИЛОЖЕНИЕ Б

#include <iostream>

#include <vector>

#include <array>

#include <math.h>

using namespace std;

struct coordinate

{

double xte, yte, zte, vxte, vyte, vzte;

};

coordinate F(struct coordinate cor);

void RungKUTT(coordinate res[], int t, int dt);

int main()

{

coordinate after[5];

int leng\_after = 5;

after[0] = { -6267326.66,24695208.01,692546.39,104.97284,124.88365,-3559.01241 };

int dt = 1;

RungKUTT(after, leng\_after, dt);

std::cout << after[1].xte << endl;

}

#include<iostream>

#include <math.h>

#include <stdio.h>

using namespace std;

struct coordinate

{

double xte, yte, zte, vxte, vyte, vzte;

};

coordinate F(struct coordinate cor)

{

double C20 = -1082.63e-6;

double nu = 398600441.8e6;

double ae = 6378136;

double xte = cor.xte;

double yte = cor.yte;

double zte = cor.zte;

double vxte = cor.vxte;

double vyte = cor.vyte;

double vzte = cor.vzte;

double r = sqrt(xte \* xte + yte \* yte + zte \* zte);

double nu\_Strih = nu / r \* r;

double ro = ae / r;

double Xate = xte / r;

double Yate = yte / r;

double Zate = zte / r;

double vxate = -nu\_Strih \* Xate + 1.5 \* C20 \* nu\_Strih \* Xate \* (ro \* ro) \* (1 - 5 \* Zate \* Zate);

double vyate = -nu\_Strih \* Yate + 1.5 \* C20 \* nu\_Strih \* Yate \* (ro \* ro) \* (1 - 5 \* Zate \* Zate);

double vzate = -nu\_Strih \* Zate + 1.5 \* C20 \* nu\_Strih \* Zate \* (ro \* ro) \* (1 - 5 \* Zate \* Zate);

cor.xte = vxte;

cor.yte = vyte;

cor.zte = vzte;

cor.vxte = vxate;

cor.vyte = vyate;

cor.vzte = vzate;

return cor;

}

#include <math.h>

#include <iostream>

#include <iostream>

#include <vector>

#include <array>

#include <math.h>

struct coordinate

{

double xte, yte, zte, vxte, vyte, vzte;

};

coordinate F(struct coordinate cor);

void RungKUTT(coordinate res[], int t, int dt)

{

coordinate K0;

coordinate K1;

coordinate K2;

coordinate K3;

coordinate resa;

coordinate ink1, ink2, ink3, ink;

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

{

resa = res[i];

K0 = F(resa);

ink1 = { resa.xte + 0.5 \* dt \* K0.xte,resa.yte + 0.5 \* dt \* K0.yte,resa.zte + 0.5 \* dt \* K0.zte,resa.vxte + 0.5 \* dt \* K0.vxte,resa.vyte + 0.5 \* dt \* K0.vyte,resa.vzte + 0.5 \* dt \* K0.vzte };

K1 = F(ink1);

ink2 = { resa.xte + 0.5 \* dt \* K1.xte,resa.yte + 0.5 \* dt \* K1.yte,resa.zte + 0.5 \* dt \* K1.zte,resa.vxte + 0.5 \* dt \* K1.vxte,resa.vyte + 0.5 \* dt \* K1.vyte,resa.vzte + 0.5 \* dt \* K1.vzte };

K2 = F(ink2);

ink3 = { (resa.xte + dt \* K2.xte),(resa.yte + dt \* K2.yte),(resa.zte + dt \* K2.zte),(resa.vxte + dt \* K2.vxte),(resa.vyte + dt \* K2.vyte),(resa.vzte + dt \* K2.vzte) };

K3 = F(ink3);

ink = { resa.xte + (dt / 6) \* (K0.xte + 2 \* K1.xte + 2 \* K2.xte + K3.xte),resa.yte + (dt / 6) \* (K0.yte + 2 \* K1.yte + 2 \* K2.yte + K3.yte),resa.zte + (dt / 6) \* (K0.zte + 2 \* K1.zte + 2 \* K2.zte + K3.zte),resa.vxte + (dt / 6) \* (K0.vxte + 2 \* K1.vxte + 2 \* K2.vxte + K3.vxte),resa.vyte + (dt / 6) \* (K0.vyte + 2 \* K1.vyte + 2 \* K2.vyte + K3.vyte),resa.vzte + (dt / 6) \* (K0.vzte + 2 \* K1.vzte + 2 \* K2.vzte + K3.vzte) };

res[i + 1] = ink;

}

}