Национальный исследовательский университет

«МЭИ»

Институт радиотехники и электроники  
Кафедра РТС

Курсовой проект

по дисциплине

Аппаратура потребителей спутниковых радионавигационных систем

ФИО студента: Мокшин А.С.

Группа: ЭР-15-14

Вариант №8

Дата: 25.06.2019

Подпись \_\_\_\_\_\_\_\_\_\_\_\_\_

ФИО преподавателя: Корогодин И.В.

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_

Москва, 2019

**Содержание**

Этап 1.

Использование сторонних средств........................................................................3

1.1 Постановка задачи вторичной обработки данных ........................................3

1.2 Описание процесса использования RTKLIB..................................................3

1.3 Использование интернет-рисурса Trimble .....................................................5

Этап 2. Моделирование........................................................................................................7

2.1 Постановка задачи моделирования .................................................................7

2.2 Листинг программы..........................................................................................7

2.3 Результат моделирования в среде MatLab .....................................................9

2.4 Вывод по этапу................................................................................................10

Этап 3.

Реализация..............................................................................................................11

3.1 Постановка задачи программной реализации...............................................11

3.2 Листинг программы ........................................................................................11

3.3 Сравнение результатов ...................................................................................13

3.4 Тесты.................................................................................................................14

Вывод .................................................................................................................... 16

### Этап 1. Использование сторонних средств

### 1.1 Постановка задачи вторичной обработки данных

На крыше корпуса Е МЭИ установлена трехдиапазонная антенна [Harxon HX-CSX601A](https://en.harxon.com/u_file/product/18_08_08/Harxon%20HX-CSX601A%20Brochure.pdf). Она через 50-метровый кабель, [сплиттер, bias-tee и усилитель](https://srns.ru/wiki/Blog:Mikhaylova/25.06.2018_%D0%90%D0%BD%D1%82%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BF%D0%BE%D1%81%D1%82_%D0%BD%D0%B0_8_%D0%BF%D1%80%D0%B8%D0%B5%D0%BC%D0%BD%D0%B8%D0%BA%D0%BE%D0%B2" \o "Blog:Mikhaylova/25.06.2018 Антенный пост на 8 приемников) подключена к трем навигационным приемникам:

* Javad Lexon LGDD,
* SwiftNavigation Piksi Multi,
* FPGA-based приемник на основе нашего ядра CoreZh.

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

**1.2 Описание процесса использования RTKLIB**

В первом пункте были получены RINEX файлы наблюдений .obs и RINEX

файлы навигационных сообщений .nav. Конвертировали из бинарного файла

в текстовый файл с помощью программы RTCCONV.

Далее бинарный файл эфемерид вывели в табличном виде с помощью

программы RTKNAVI. В качестве входных данных использовали бинарные

файлы BINR\_morning.bin и BINR\_evening.bin. Результат представлен ниже

на Рис.1.1 и Рис.1.2.

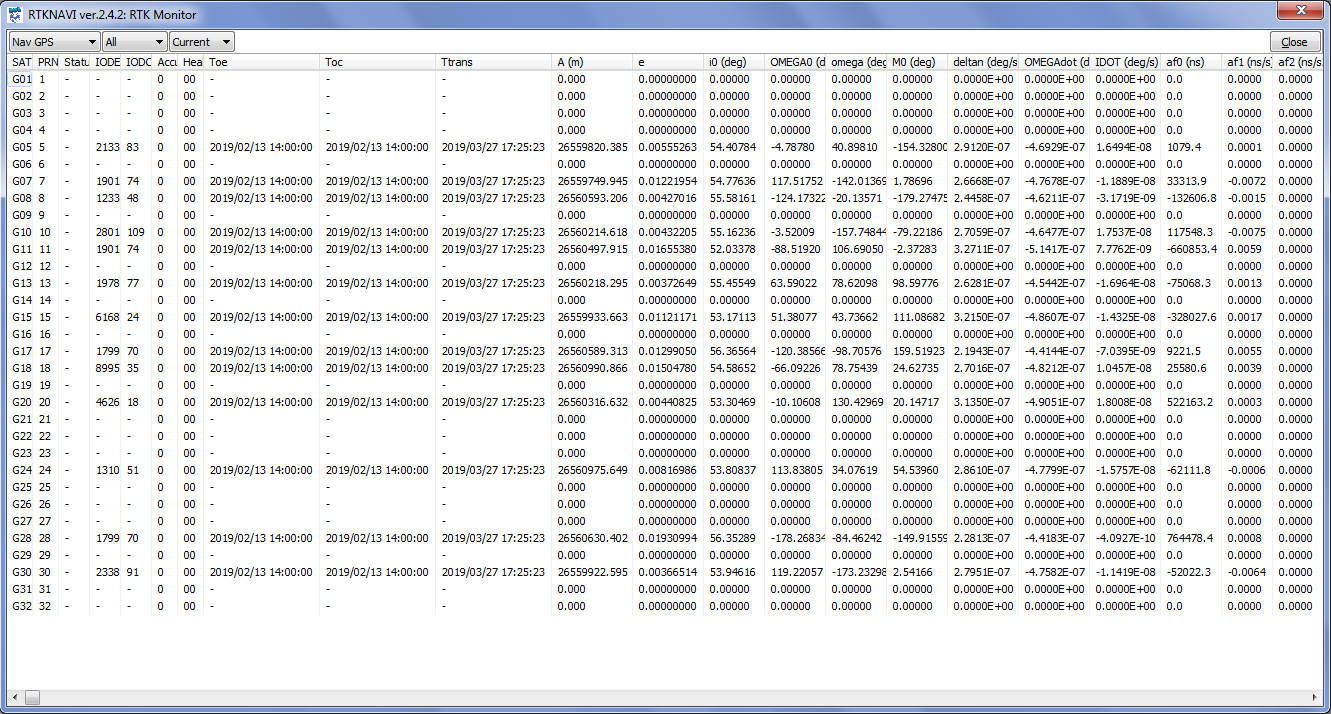


Рис.1.1 Эфемериды, полученные из файла BINR\_evening.bin

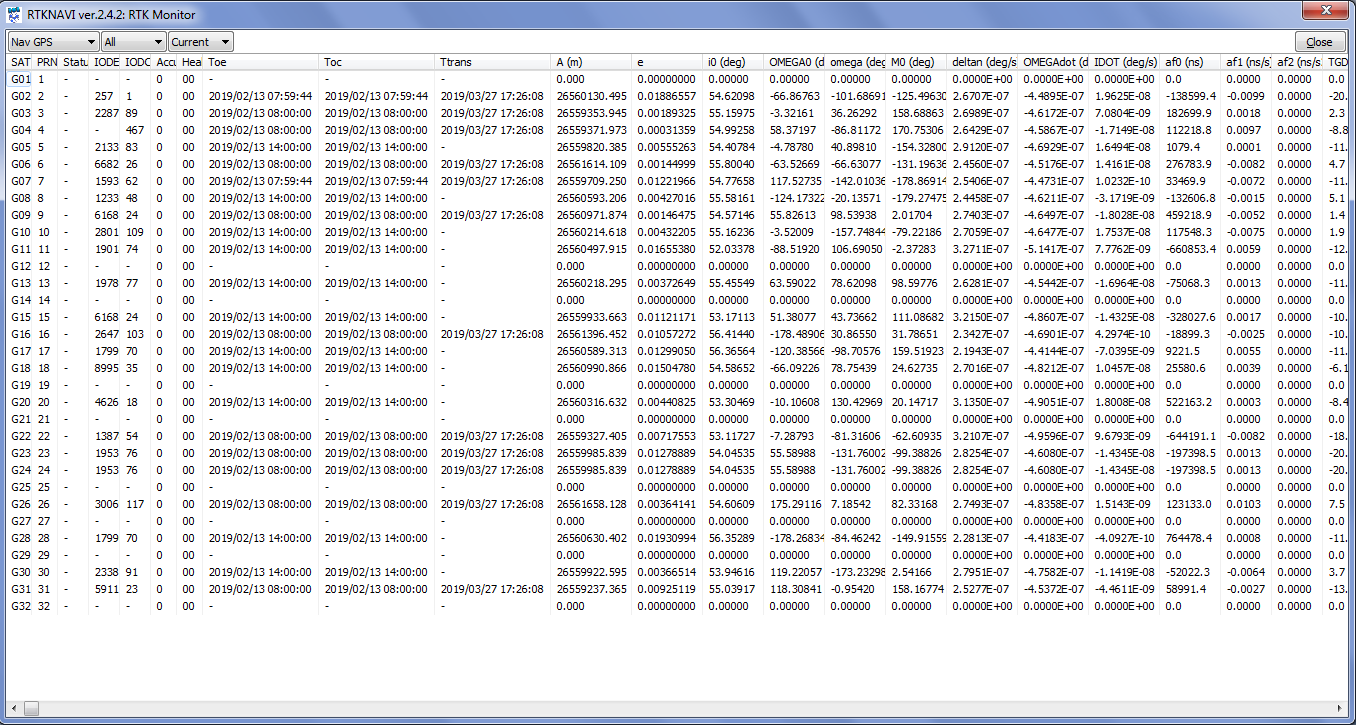


Рис.1.2 Эфемериды, полученные из файла BINR\_morning.bin

Эфемериды собственного спутника в nav-файле RINEX представлены на рис.1.3.

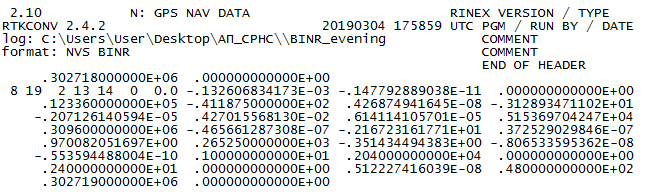
****

Рис.1.3 Эфемериды собственного спутника.

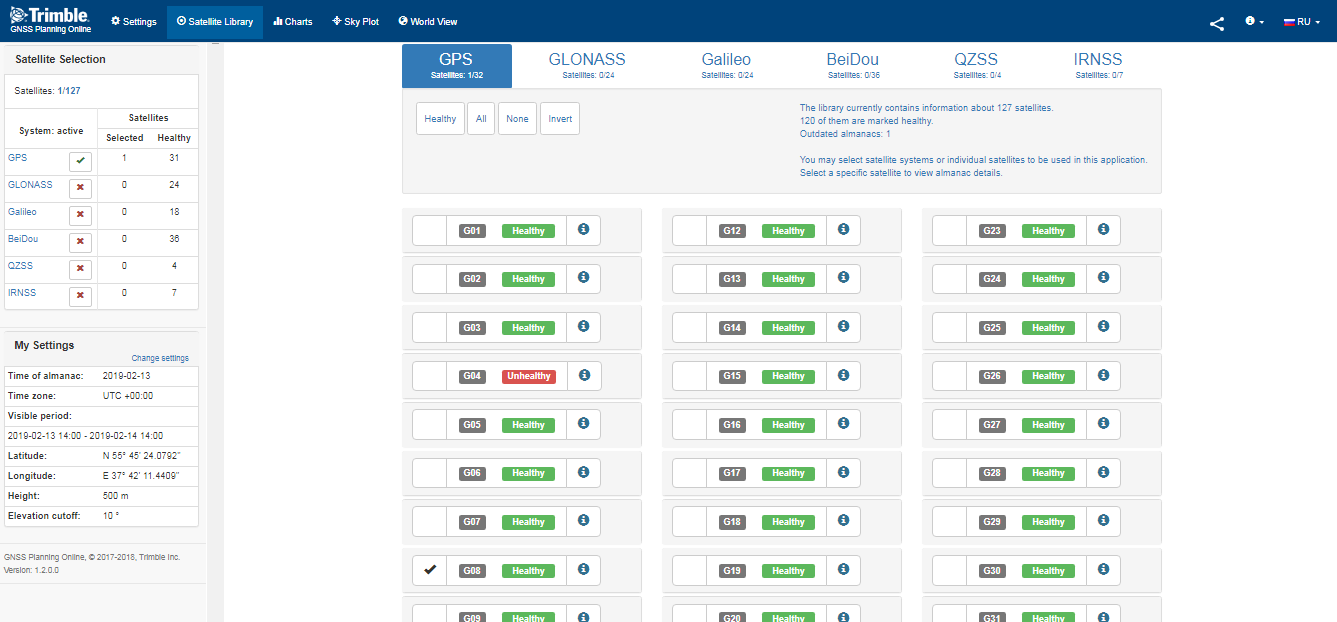
**1.3 Использование интернет-рисурса Trimble **

Рис.1.4 Выбор спутника согласно варианту курсового проекта (спутник №8)

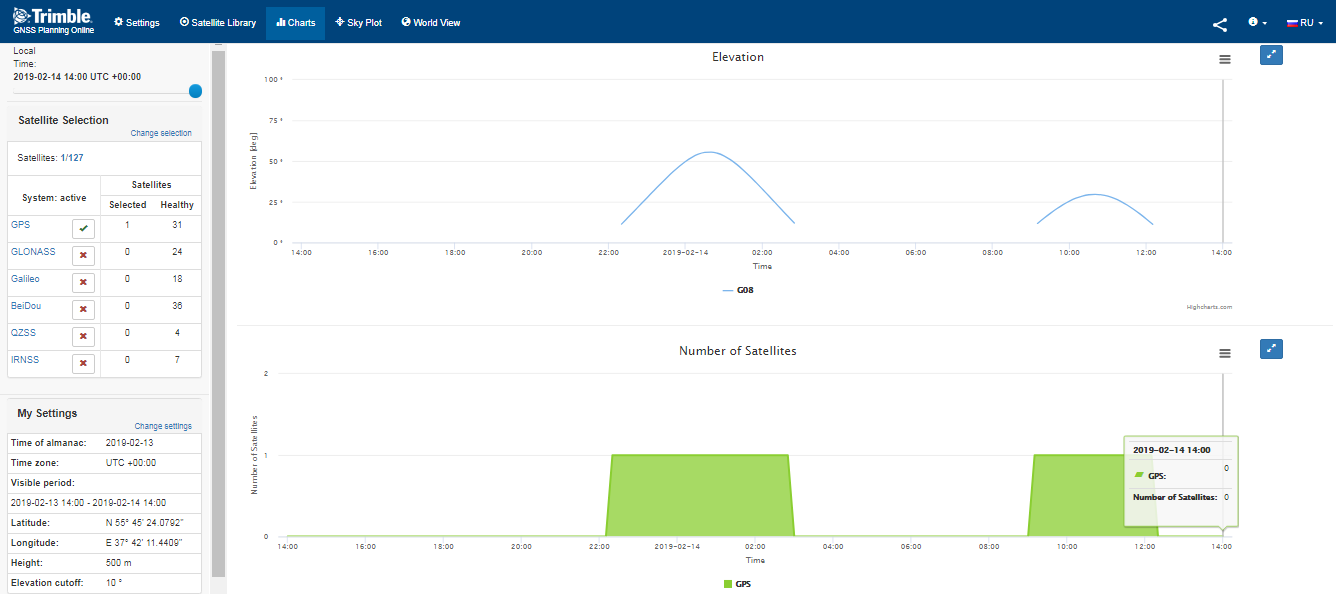


Рис.1.5 Угол места спутника №8 на заданный интервал

|  |  |
| --- | --- |
| Рис.1.6а SkyView полученное в GNSS Planing Online | Рис.1.6б SkyView полученное в GNSS Planing Online |

**Этап 2. Моделирование**

**2.1 Постановка задачи моделирования**

Требуется реализовать на языке Matlab или Python функцию расчета положения спутника GPS на заданный момент по шкале GPST.

**2.2 Листинг программы**

clear all, close all

for i = 1:86400

mu = 3.986005\*10^14; %гравитационная постоянная земли

dOmega\_e\_dt = 7.2921151467\*10^(-5); %скорость вращения земли

t\_oe = 309600 + 18; % эфемеридное время

Az = 26560593.206; % квадратный корень большой полуоси

e = 0.00427016; % эксцентричность

i\_0 = 55.581618; %угол наклона в эталонное время

i\_0 = degtorad(i\_0);

OMEGA\_0 = -124.17322; %долгота

OMEGA\_0 = degtorad(OMEGA\_0);

omega = -20.13571; %Аргумент Перигея

omega = degtorad(omega);

M\_0 = -179.27475; %средняя аномалия

M\_0 = degtorad(M\_0);

delta\_n = 2.4458E-07;

delta\_n = degtorad(delta\_n);

OMEGAdot = -4.6211E-07; %скорость правильного вознесения

OMEGAdot = degtorad(OMEGAdot);

IDOT = -3.1719E-09; %скорость угла наклона

IDOT = degtorad(IDOT);

C\_uc = -2.0713E-06;

C\_us = 6.1411E-06;

C\_rc = 2.6525E+02;

C\_rs = -4.1188E+01;

C\_ic = -4.6566E-08;

C\_is = 3.7253E-08;

n\_0 = sqrt(mu/Az^3);

t = 396000 + 18 + i; % время передачи + високосные секунды

t\_k = t - t\_oe; % время из эфемерид эталонной эпохи

n = n\_0 + delta\_n;

M\_k = M\_0+n\*t\_k;

m = 1;

x(1) = 0;

for m = 1:200

x(m+1) = M\_k + e\*sin(x(m));

if abs(x(m+1) - x(m))<10^(-8)

break

end

m = m + 1;

end

E\_k = x(m+1);

v\_k = atan2( (sqrt(1-e^2)\*sin(E\_k))/(1 - e\*cos(E\_k)),(cos(E\_k) - e)/(1 - e\*cos(E\_k)));

Phi\_k = v\_k + omega;

delta\_u\_k = C\_us\*sin(2\*Phi\_k) + C\_uc\*cos(2\*Phi\_k); %аргумент коррекции широты

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

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

u\_k = Phi\_k+delta\_u\_k;

r\_k = Az\*(1 - e\*cos(E\_k)) + delta\_r\_k;

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

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

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

Omega\_k = OMEGA\_0 + (OMEGAdot - dOmega\_e\_dt)\*t\_k - dOmega\_e\_dt\*t\_oe;

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

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

z\_ecef = y\_k\_orb\*sin(i\_k);

X(1,i) = x\_ecef;

Y(1,i) = y\_ecef;

Z(1,i) = z\_ecef;

%Перевод из ECEF в ECI

theta = dOmega\_e\_dt \* t\_k

x\_eci = x\_ecef\*cos(theta) - y\_ecef\*sin(theta);

y\_eci = x\_ecef\*sin(theta) + y\_ecef\*cos(theta);

z\_eci = z\_ecef;

X\_eci(1,i) = x\_eci;

Y\_eci(1,i) = y\_eci;

Z\_eci(1,i) = z\_eci;

%Нахождение азимута и угла места

lat = 55.75;

lon = 37.62;

[East,North,Up] = ecef2enu(x\_ecef,y\_ecef,z\_ecef, lat, lon, 150, wgs84Ellipsoid );

p = sqrt(East^2 + North^2 + Up^2);

El = asin(Up/p);

Az = atan2(East,North);

E\_i(1,i) = -El\*180/pi + 90;

A\_i(1,i) = Az;

end

figure;

plot(E\_i(1,:));

grid on;

figure;

polar (A\_i(1,:),E\_i(1,:));

camroll(90)

N = 20;

thetavec = linspace(0,pi,N);

phivec = linspace(0,2\*pi,2\*N);

[th, ph] = meshgrid(thetavec,phivec);

R = 6.371\*10^6;

x = R.\*sin(th).\*cos(ph);

y = R.\*sin(th).\*sin(ph);

z = R.\*cos(th);

lat = degtorad(55);

lon = degtorad(37);

x\_msk = R\*cos(lat)\*cos(lon);

y\_msk = R\*cos(lat)\*sin(lon);

z\_msk = R\*sin(lat);

figure;

surf(x,y,z);

axis equal

hold on

plot3(X(1,:), Y(1,:),Z(1,:));

axis vis3d

grid on

title('Earth-Centered, Earth-Fixed (ECEF) coordinate system');

xlabel('X, m');

ylabel('Y, m');

zlabel('Z, m');

figure ;

surf(x,y,z);

axis equal

hold on

plot3(X\_eci(1,:), Y\_eci(1,:),Z\_eci(1,:),x\_msk,y\_msk,z\_msk, 'k.','MarkerSize',20 );

axis vis3d

grid on

title('Earth-centered inertial (ECI) coordinate system');

xlabel('X, m');

ylabel('Y, m');

zlabel('Z, m');

**2.3 Результат моделирования в среде MatLab**

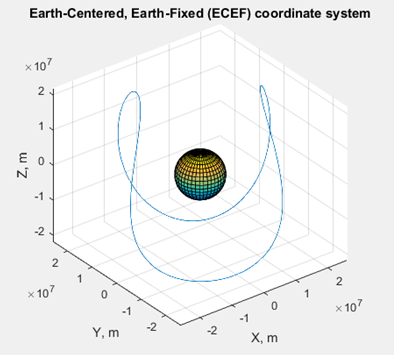


Рис.2.1 Траектории спутника в системе координат ECEF

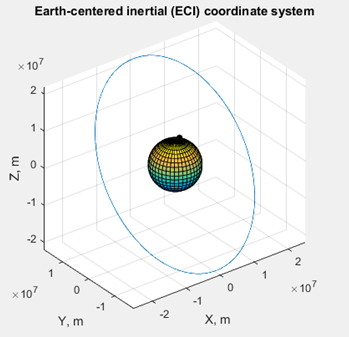


Рис.2.2 Траектории спутника в системе координат ECI

|  |  |
| --- | --- |
| Рис.2.3а SkyView полученное при моделировании в MatLab | Рис.2.3б SkyView полученное при моделировании в MatLab |
| Рис.2.4а SkyView полученное в GNSS Planing Online | Рис.2.4б SkyView полученное в GNSS Planing Online |

**2.4 Вывод по этапу**

В ходе предыдущего этапа данного курсового проекта были получены данные эфемерид спутников. На данном этапе требовалось, используя полученные эфемериды, промоделировать положение спутника в разные моменты времени и построить траекторию его движения. Рассчитанные в Matlab и полученное положение спутников в GNSS Planing Online совпадают.

**Этап 3. Реализация**

**3.1 Постановка задачи программной реализации**

Требуется разработать на языке С/С++ функцию расчета положения спутника GPS на заданное время по шкале GPST, 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.

**3.2 Листинг программы**

#include <iostream>

#include <fstream>

#include <gpssvpos.h>

#include <kepler.h>

#include <ctime>

using namespace std;

int **main**()

{

time\_t start, end;

double t = 396018;

double delt = 0.1;

double \*koord = new double[3];

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

double max\_del = 0;

int i\_max = 0;

std::ofstream out;

out.open("Е:\\cpp.txt");

std::ifstream in("Е:\\mat.txt");

if (!in)

{

std::cout << "File not open!" << std::endl;

} else {

std::cout << "File open!" << std::endl;

}

time(&start);

for (int i = 0; i < (12\*3600/delt); i++)

{

gps\_coord(t,koord);

t += delt;

std::string koord\_str1 = std::to\_string(koord[0]);

std::string koord\_str2 = std::to\_string(koord[1]);

std::string koord\_str3 = std::to\_string(koord[2]);

out << koord\_str1 << " " << koord\_str2 << " " << koord\_str3 << std::endl;

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

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

{

if (abs(koord[j]-koord\_matlab[j]) > max\_del)

{

max\_del = abs(koord[j]-koord\_matlab[j]);

i\_max = i;

}

}

}

time(&end);

in.close();

delete[] koord;

koord = nullptr;

delete[] koord\_matlab;

koord\_matlab = nullptr;

double seconds = difftime(end, start);

std::string seconds1 = std::to\_string(seconds\*1000000/(12\*3600/delt));

cout << "Srednee vremia raschota, mcs: " << seconds1 << std::endl;

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

cout << "Maximalnaia raznost koordinat: " << max\_del1 << std::endl;

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

cout << "Nomer otcheta s max raznostiy: " << imax << std::endl;

out.close();

in.close();

}

#include <gpssvpos.h>

#include <kepler.h>

#include <iostream>

#include <cmath>

#include <ostream>

using namespace std;

void **gps\_coord**(double t, double \*koord)

{

double mu = 3.986005E+14;

double we = 7.292115E-05;

double toe = .309618000000E+06;

double a\_sqr = sqrt(.26560593206E+06);

double e = .427016000000E-02;

double M0 = -.17927475E+03\*M\_PI/180;

double omega = -.2013571E+02\*M\_PI/180;

double i0 = .55581618E+02\*M\_PI/180;

double omega0 = -.12417322E+03\*M\_PI/180;

double del\_n = .24458E-08\*M\_PI/180;

double i\_dot = -.31719E-08\*M\_PI/180;

double omega\_dot = -.46211E-06\*M\_PI/180;

double cuc = -.20713E-05;

double cus = .61411E-05;

double crc = .26525E+03;

double crs = -.41188E+02;

double cic = .46566E-07;

double cis = .37253E-07;

double tk = t - toe;

double Mk = M0 + (sqrt(mu)/pow(a\_sqr,3) + del\_n)\*tk;

double Ek = kepler(Mk,e);

double Vk = atan2(sqrt(1-pow(e,2))\*sin(Ek),cos(Ek)-e);

double Uk = omega + Vk + cuc\*cos(2\*(omega + Vk)) + cus\*sin(2\*(omega + Vk));

double rk = a\_sqr\*a\_sqr\*(1-e\*cos(Ek)) + crc\*cos(2\*(omega+Vk))+crs\*sin(2\*(omega+Vk));

double ik = i0 + i\_dot\*tk + cic\*cos(2\*(omega + Vk)) + cis\*sin(2\*(omega + Vk));

double lambk = omega0 + (omega\_dot - we)\*tk - we\*toe;

koord[0] = (cos(-lambk)\*cos(-Uk)-sin(-lambk)\*cos(-ik)\*sin(-Uk))\*rk;

koord[1] = (-sin(-lambk)\*cos(-Uk)-cos(-lambk)\*cos(-ik)\*sin(-Uk))\*rk;

koord[2] = (-sin(-ik)\*sin(-Uk))\*rk;

}

#include <kepler.h>

#include <cmath>

double **kepler**(double Mk, double e){

double Ek = Mk;

double Ek1 = Ek;

do{

Ek1 = Ek;

Ek = Mk + e\*sin(Ek);

}while(fabs(Ek1-Ek)/fabs(Ek) > 0.0001);

return Ek;

}

**3.3 Сравнение результатов**

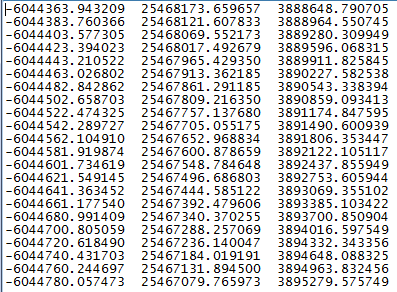


Рис. 3.1 Результаты полученные на С++

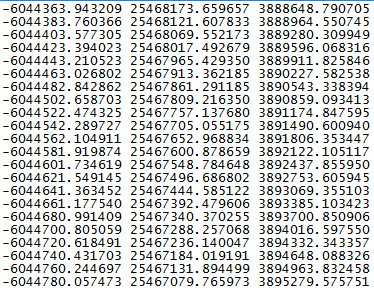


Рис. 3.2 Результаты полученные в программе Matlab

Максимальная разница координат представлена на рис. 3.3

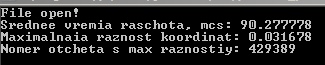


Рис. 3.3 Максимальная разница координат

Видно из результатов, максимальное различие координат, полученных с помощью C++ и Matlab составляет 0.031678 м.

**3.4 Тесты**

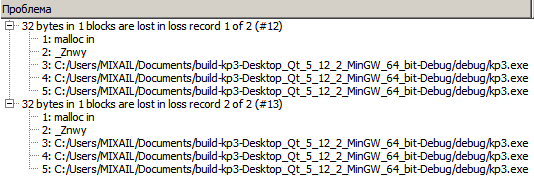


Рис. 3.4 Результаты теста при наличии ошибок памяти

Наблюдаются утечки памяти. Они вызваны из-за выделения памяти под хранение координат и не очисткой этой памяти после использования.

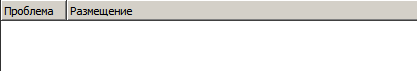


Рис. 3.5 Результаты теста без ошибок

Утечек памяти не обнаружено.

Замерять время производилась с помощью времени выполнения цикла и делением на количество итераций (оценка приближенная).



Рис. 3.6 Результаты теста на время выполнения

**Вывод**

В ходе выполнения курсового проекта по дисциплине «Аппаратура потребителей спутниковых радионавигационных систем» были получены практические знания об обработке сигналов СРНС на основании измерений навигационных приемников. Изучили методы простейшего моделирования движения спутника с помощью онлайн-сервиса Trimble GNSS Planning Online. Во втором этапе курсового проекта произвели моделирование траектории движения спутника по полученным эфемеридам. В третьем этапе курсового проекта была произведена вторичная обработка движения спутника. Так же произвели анализ утечек.