

Федеральное государственное бюджетное образовательное учреждение
высшего образования "Национальный исследовательский университет "МЭИ"

ИРЭ им. В.А. Котельникова

Кафедра радиотехнических систем

КУРСОВОЙ ПРОЕКТ

по дисциплине

Аппаратура потребителей спутниковых радионавигационных систем

ФИО студента: Климова А.П.

Группа: ЭР-15-14

Вариант №24

Дата: 20.06.2019

Подпись _____

ФИО преподавателя: Корогодин И.В.

Оценка _____

Москва, 2019

Содержание

Этап 1. Использование	3
1.1 Постановка задачи вторичной обработки данных	3
1.2 Использование программы RTKPOST	3
1.3 Использование интернет-ресурса Trimble	6
Этап 2. Моделирование	9
2.1 Постановка задачи моделирования	9
2.2 Расчет текущего времени по шкале GPS.....	9
2.3 Анализ данных .nav файла эфемерид	9
2.4 Результат моделирования в среде MatLab	11
2.5 Код программы моделирования в среде MatLab	13
Этап 3. Реализация.....	16
3.1 Постановка задачи программной реализации.....	16
3.2 Реализация на языке C++	16
3.3 Контроль утечек	20
Вывод	23

Этап 1. Использование

1.1 Постановка задачи вторичной обработки данных

На крыше корпуса Е МЭИ установлена трехдиапазонная антенна Naigon HX-CSX601A. Она через 50-метровый кабель, сплиттер, bias-tee и усилитель 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. Необходимо провести обработку данных от приемника CoreZh, представленных в бинарном виде в формате NVS BINR. Для проведения вторичной обработки имеющихся наблюдений используется подпрограмма RTKPOST программы RTKlib версии 2.4.3 (Рисунок 1).



Рисунок 1 – Окно запуска RTKlib v.2.4.2

1.2 Использование программы RTKPOST

Для нахождения навигационного решения в открывшемся окне RTKPOST (Рисунок 2) в верхней строке необходимо указать путь к одному из файлов наблюдений с расширением .obs , а в третьей строке – путь к файлу эфемерид с расширением .nav. Для всех трех приемников используется один файл эфемерид на утро - BINR_morning.nav и на вечер BINR_evening.nav.

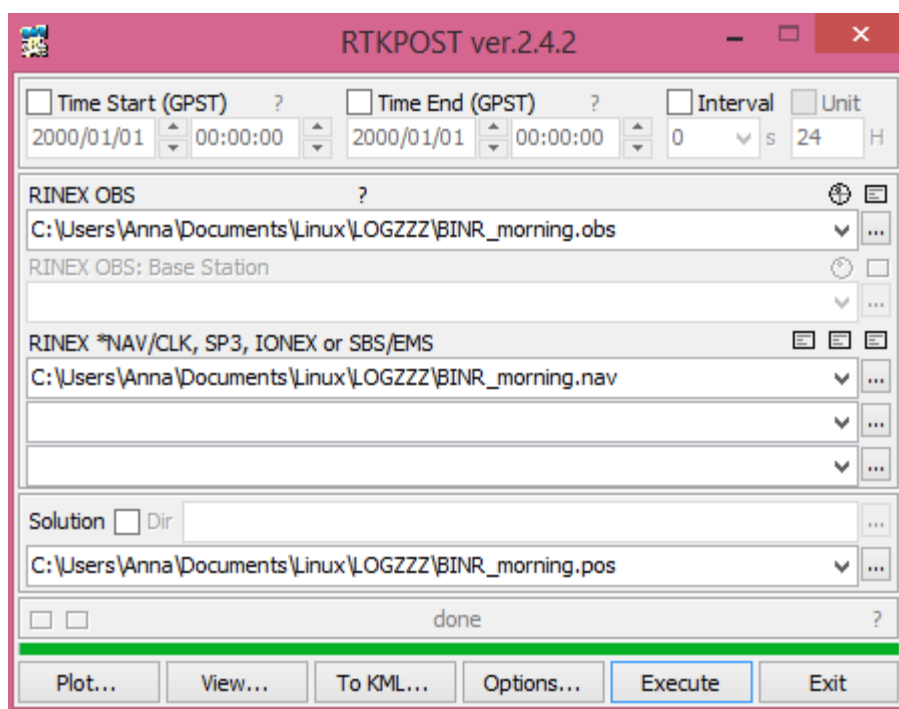


Рисунок 2 – Окно RTKPOST

После нажатия кнопки «Execute» программа производит вторичную обработку, результаты которой записываются в файл с расширением .pos по адресу, указанном в последней строке окна на рис. 2. Нажатие кнопки «view» открывает полученный файл с навигационным решением в текстовом виде (Рисунок 3):

```
% program : RTKPOST ver.2.4.3 b29
% inp file : C:\Users\Александр\Desktop\X\Курсовой по АП_CPHC\Receivers_logs_2018\piksi.obs
% inp file : C:\Users\Александр\Desktop\X\Курсовой по АП_CPHC\Receivers_logs_2018\corezh.nav
% obs start : 2018/02/02 22:18:26.0 GPST (week1986 512306.0s)
% obs end : 2018/02/03 21:25:52.0 GPST (week1986 595552.0s)
% pos mode : single
% elev mask : 15.0 deg
% ionos opt : broadcast
% tropo opt : saastamoinen
% ephemeris : broadcast
%
% (lat/lon/height=WGS84/ellipsoidal,Q=1:fix,2:float,3:sbas,4:dgps,5:single,6:ppp,ns=# of satellites)
% GPST latitude(deg) longitude(deg) height(m) Q ns sdn(m) sde(m) sdu(m) sdsne(m) sdsde(m) sdsdu(m) age(s) ratio
2018/02/02 22:18:26.000 55.756363846 37.708054261 194.3335 5 5 21.9602 2.7540 51.6579 6.1622 9.5154 33.4693 0.00 0.0
2018/02/02 22:18:27.000 55.756350732 37.708049491 191.2321 5 5 22.0053 2.7590 51.7654 6.1825 9.5466 33.5394 0.00 0.0
2018/02/02 22:18:28.000 55.756330390 37.708045623 186.1683 5 5 22.0505 2.7640 51.8733 6.2029 9.5780 33.6097 0.00 0.0
2018/02/02 22:18:29.000 55.756316735 37.708040296 182.3651 5 5 22.0959 2.7691 51.9816 6.2234 9.6094 33.6802 0.00 0.0
2018/02/02 22:18:30.000 55.756318063 37.708039554 182.7434 5 5 22.1414 2.7742 52.0902 6.2439 9.6410 33.7509 0.00 0.0
2018/02/02 22:18:31.000 55.756300379 37.708035945 178.2230 5 5 22.1871 2.7793 52.1991 6.2645 9.6726 33.8219 0.00 0.0
2018/02/02 22:18:32.000 55.756294429 37.708034042 176.3551 5 5 22.2329 2.7845 52.3085 6.2851 9.7044 33.8931 0.00 0.0
2018/02/02 22:18:33.000 55.756312852 37.708037284 181.3868 5 5 22.2789 2.7897 52.4182 6.3058 9.7363 33.9646 0.00 0.0
2018/02/02 22:18:34.000 55.756323233 37.708040889 183.8051 5 5 22.3250 2.7950 52.5283 6.3266 9.7682 34.0363 0.00 0.0
2018/02/02 22:18:35.000 55.756326697 37.708042190 184.3380 5 5 22.3713 2.8002 52.6387 6.3474 9.8003 34.1082 0.00 0.0
2018/02/02 22:18:36.000 55.756347747 37.708049993 190.1953 5 5 22.4177 2.8055 52.7495 6.3684 9.8325 34.1803 0.00 0.0
2018/02/02 22:18:37.000 55.756368317 37.708056326 196.2063 5 5 22.4643 2.8109 52.8607 6.3893 9.8647 34.2527 0.00 0.0
2018/02/02 22:18:38.000 55.756390522 37.708058571 202.4140 5 5 22.5110 2.8162 52.9723 6.4104 9.8971 34.3254 0.00 0.0
2018/02/02 22:18:39.000 55.756375529 37.708058562 198.3641 5 5 22.5580 2.8217 53.0843 6.4315 9.9296 34.3983 0.00 0.0
2018/02/02 22:18:40.000 55.756358656 37.708055621 192.3379 5 5 22.6051 2.8271 53.1966 6.4527 9.9622 34.4715 0.00 0.0
2018/02/02 22:18:41.000 55.756343873 37.708053349 187.3424 5 5 22.6523 2.8326 53.3093 6.4740 9.9949 34.5449 0.00 0.0
2018/02/02 22:18:42.000 55.756325193 37.708049597 182.6356 5 5 22.6998 2.8381 53.4224 6.4953 10.0277 34.6186 0.00 0.0
2018/02/02 22:18:43.000 55.756317816 37.708046462 180.9061 5 5 22.7473 2.8436 53.5359 6.5167 10.0606 34.6925 0.00 0.0
```

Find Read... Option... Close

```

% program : RTKPOST ver.2.4.2
% inp file : C:\Users\Anna\Documents\Linux\LOGZZZ\BINR_morning.obs
% inp file : C:\Users\Anna\Documents\Linux\LOGZZZ\BINR_morning.nav
% obs start : 2019/02/13 06:34:52.0 GPST (week2040 282892.0s)
% obs end : 2019/02/13 06:43:03.0 GPST (week2040 283383.0s)
% pos mode : single
% elev mask : 15.0 deg
% ionos opt : broadcast
% tropo opt : saastamoinen
% ephemeris : broadcast
%
% (lat/lon/height=WGS84/ellipsoidal,Q=1:fix,2:float,3:sbas,4:dgps,5:single,6:ppp,ns=# of satellites)
% GPST latitude(deg) longitude(deg) height(m) Q ns sdn(m) sde(m) sdu(m) sdne(m) sdeu(m) sduu(m) age(s) ratio
2019/02/13 06:35:20.000 55.756727964 37.703259108 189.4054 5 6 2.4396 2.0847 4.7521 0.9638 -0.9841 -1.3904 0.00 0.0
2019/02/13 06:35:21.000 55.756735124 37.703262345 190.3729 5 6 2.4396 2.0848 4.7523 0.9637 -0.9840 -1.3907 0.00 0.0
2019/02/13 06:35:22.000 55.756744615 37.703259245 188.4035 5 6 2.4397 2.0848 4.7526 0.9636 -0.9839 -1.3911 0.00 0.0
2019/02/13 06:35:23.000 55.756756044 37.703243530 187.9403 5 6 2.4397 2.0848 4.7529 0.9634 -0.9837 -1.3914 0.00 0.0
2019/02/13 06:35:24.000 55.756771472 37.703237213 185.2231 5 6 2.4397 2.0848 4.7532 0.9633 -0.9836 -1.3917 0.00 0.0
2019/02/13 06:35:25.000 55.756772103 37.703237676 186.6111 5 6 2.4397 2.0849 4.7534 0.9632 -0.9835 -1.3921 0.00 0.0
2019/02/13 06:35:26.000 55.756760828 37.703234388 186.4243 5 6 2.4397 2.0849 4.7537 0.9631 -0.9833 -1.3924 0.00 0.0
2019/02/13 06:35:27.000 55.756747726 37.703233194 187.9784 5 6 2.4397 2.0849 4.7540 0.9630 -0.9832 -1.3927 0.00 0.0
2019/02/13 06:35:28.000 55.756747475 37.703226867 188.1822 5 6 2.4398 2.0849 4.7542 0.9629 -0.9830 -1.3930 0.00 0.0
2019/02/13 06:35:29.000 55.756738774 37.703251321 188.3195 5 6 2.4398 2.0850 4.7545 0.9628 -0.9829 -1.3934 0.00 0.0
2019/02/13 06:35:30.000 55.756737176 37.703242994 190.9801 5 6 2.4398 2.0850 4.7548 0.9626 -0.9828 -1.3937 0.00 0.0
2019/02/13 06:35:31.000 55.756739802 37.703228373 188.6256 5 6 2.4398 2.0850 4.7550 0.9625 -0.9826 -1.3940 0.00 0.0
2019/02/13 06:35:32.000 55.756726219 37.703237255 188.7990 5 6 2.4398 2.0851 4.7553 0.9624 -0.9825 -1.3943 0.00 0.0
2019/02/13 06:35:33.000 55.756733220 37.703250232 190.0518 5 6 2.4399 2.0851 4.7556 0.9623 -0.9824 -1.3947 0.00 0.0
2019/02/13 06:35:34.000 55.756735832 37.703269425 188.8554 5 6 2.4399 2.0851 4.7559 0.9622 -0.9822 -1.3950 0.00 0.0
2019/02/13 06:35:35.000 55.756723252 37.703267812 190.5771 5 6 2.4399 2.0851 4.7561 0.9621 -0.9821 -1.3953 0.00 0.0
2019/02/13 06:35:36.000 55.756725073 37.703264636 191.0034 5 6 2.4399 2.0852 4.7564 0.9620 -0.9819 -1.3957 0.00 0.0
2019/02/13 06:35:37.000 55.756738264 37.703270647 190.3633 5 6 2.4399 2.0852 4.7567 0.9618 -0.9818 -1.3960 0.00 0.0

```

Рисунок 3 – Вид .pos файла для утра

Эфемериды собственного спутника №24 получены из .nav файла.

BINR_morning.nav — Блокнот

Файл	Правка	Формат	Вид	Справка
-.113090000000E+05	-.125937500000E+02	.461269217267E-08	.298020318584E+01	
-.692903995514E-06	.313590513542E-03	.883825123310E-05	.515357855988E+04	
.288000000000E+00	-.540167093277E-07	.101878308024E-01	.111758708954E-07	
.959801664072E+00	.207375000000E+03	-.151515041066E+01	-.800533345427E-08	
-.299298181256E-09	.100000000000E+01	.204000000000E+04	.000000000000E+00	
.240000000000E+01	.000000000000E+00	-.884756445885E-08	.467000000000E+03	
.282919000000E+06	.000000000000E+00			
7 19 2 13 7 59 44.0	.334698706865E-04	-.716227077646E-11	.000000000000E+00	
.159340000000E+05	.592812500000E+02	.443411323284E-08	-.312185539735E+01	
.299699604511E-05	.122196576558E-01	.932998955250E-05	.515361128235E+04	
.287984000000E+06	-.149011611938E-07	.205123923947E+01	-.134110450745E-06	
.956031671517E+00	.196718750000E+03	-.247854833991E+01	-.780711091179E-08	
.178578867098E-11	.100000000000E+01	.204000000000E+04	.000000000000E+00	
.240000000000E+01	.000000000000E+00	-.111758708954E-07	.620000000000E+02	
.282919000000E+06	.000000000000E+00			
22 19 2 13 8 0 0.0	-.644191145897E-03	-.818545231596E-11	.000000000000E+00	
.138780000000E+05	-.746875000000E+01	.560380518030E-08	-.109273929892E+01	
-.361353158951E-06	.717552879360E-02	.708922743797E-05	.515357423592E+04	
.288000000000E+06	-.182539224625E-06	-.127198374620E+00	.204890966415E-07	
.927071221495E+00	.224062500000E+03	-.141923298427E+01	-.865607484598E-08	
.168935608275E-09	.100000000000E+01	.204000000000E+04	.000000000000E+00	
.240000000000E+01	.000000000000E+00	-.181607902050E-07	.540000000000E+02	
.282919000000E+06	.000000000000E+00			
23 19 2 13 8 0 0.0	-.197398483753E-03	.125055521494E-11	.000000000000E+00	
.195320000000E+05	-.137812500000E+02	.493127674112E-08	-.173465232856E+01	
-.832602381706E-06	.127888879506E-01	.821799039841E-05	.515363811684E+04	
.288000000000E+06	.193715095520E-06	.970226437323E+00	-.175088644028E-06	
.943269317559E+00	.207812500000E+03	-.229964623594E+01	-.804247785863E-08	
-.25036751671E-09	.100000000000E+01	.204000000000E+04	.000000000000E+00	

Рисунок 4 – Выдержка из полученного .nav файла для утренних измерений

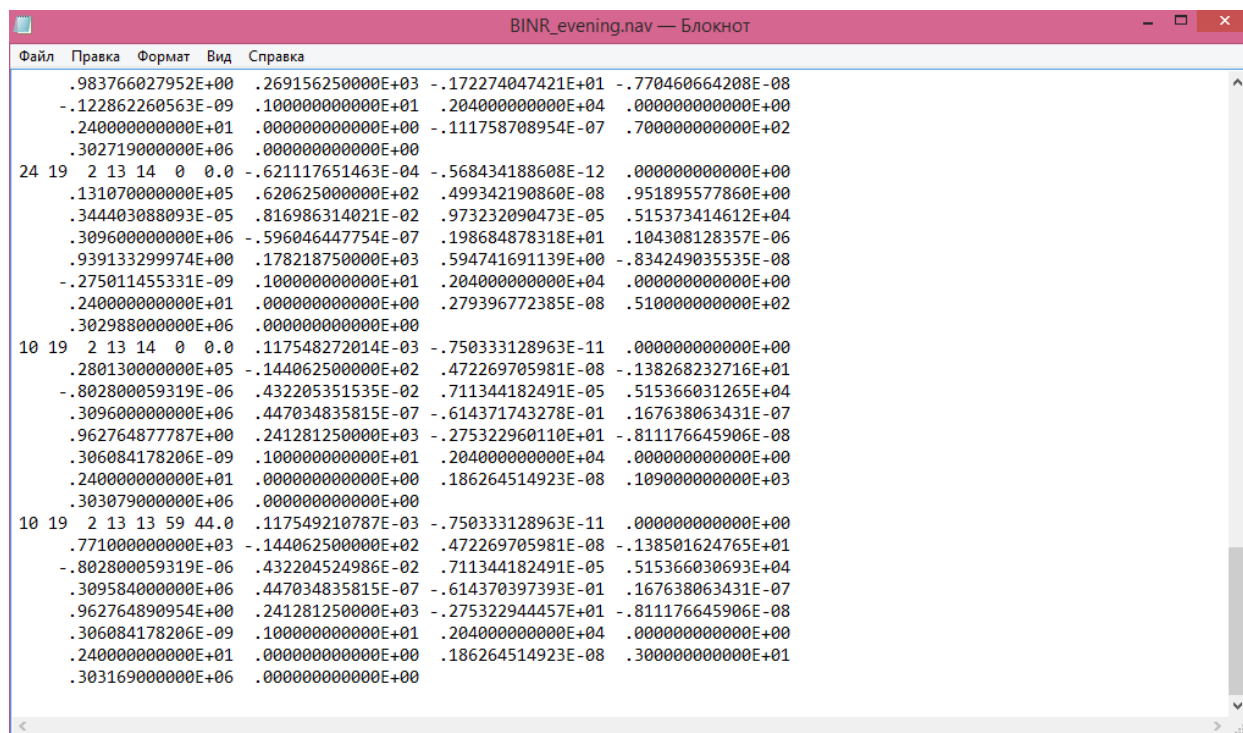


Рисунок 5 – Выдержка из полученного .nav файла для вечерних измерений

```

24 19 2 13 14 0 0.0 -.621117651463E-04 -.568434188608E-12 .000000000000E+00
.131070000000E+05 .620625000000E+02 .499342190860E-08 .951895577860E+00
.344403088093E-05 .816986314021E-02 .973232090473E-05 .515373414612E+04
.309600000000E+06 -.596046447754E-07 .198684878318E+01 .104308128357E-06
.939133299974E+00 .178218750000E+03 .594741691139E+00 -.834249035535E-08
-.275011455331E-09 .100000000000E+01 .204000000000E+04 .000000000000E+00
.240000000000E+01 .000000000000E+00 .279396772385E-08 .510000000000E+02
.302988000000E+06 .000000000000E+00

```

1.3 Использование интернет-ресурса Trimble

С помощью интернет-ресурса Trimble (<https://www.gnssplanning.com>) был построен график угла места для выбранного спутника на заданный период времени.

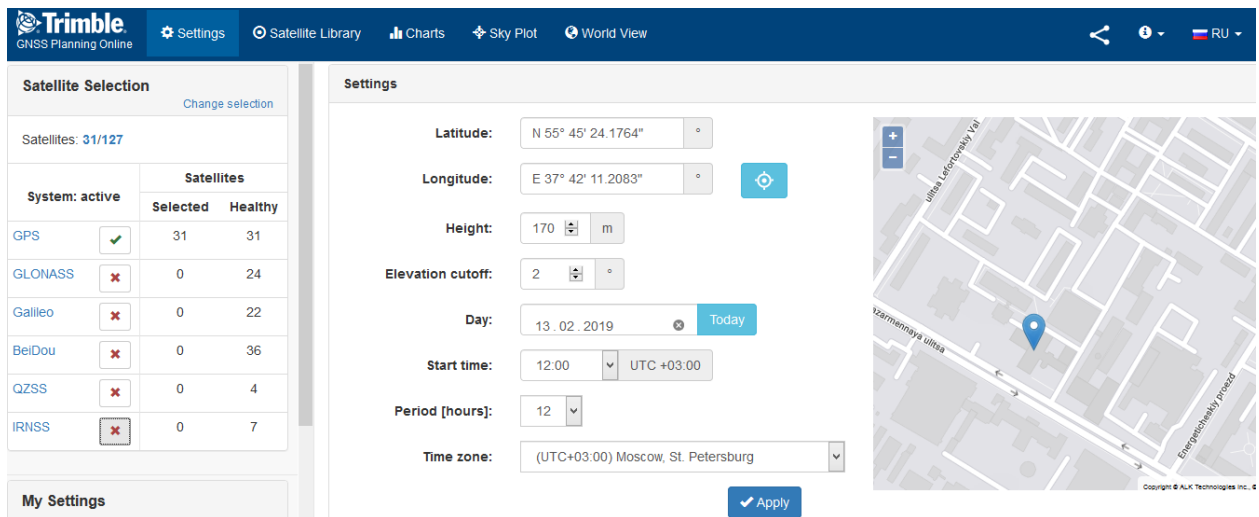


Рисунок 6 – Выставление параметров места и времени измерения

Из доступных спутников GPS был выбран необходимый. Таким образом, все спутники кроме №24 были отключены.

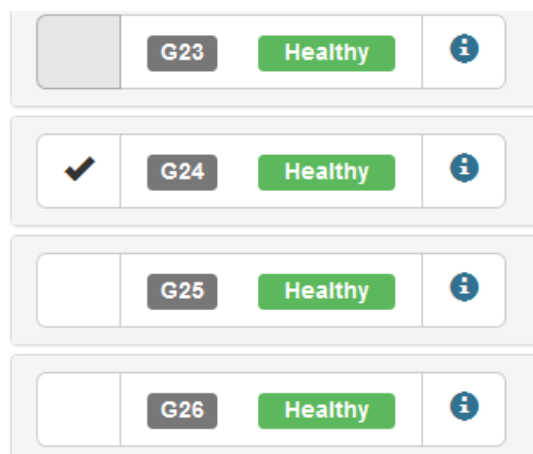


Рисунок 7 – Отключение сторонних спутников

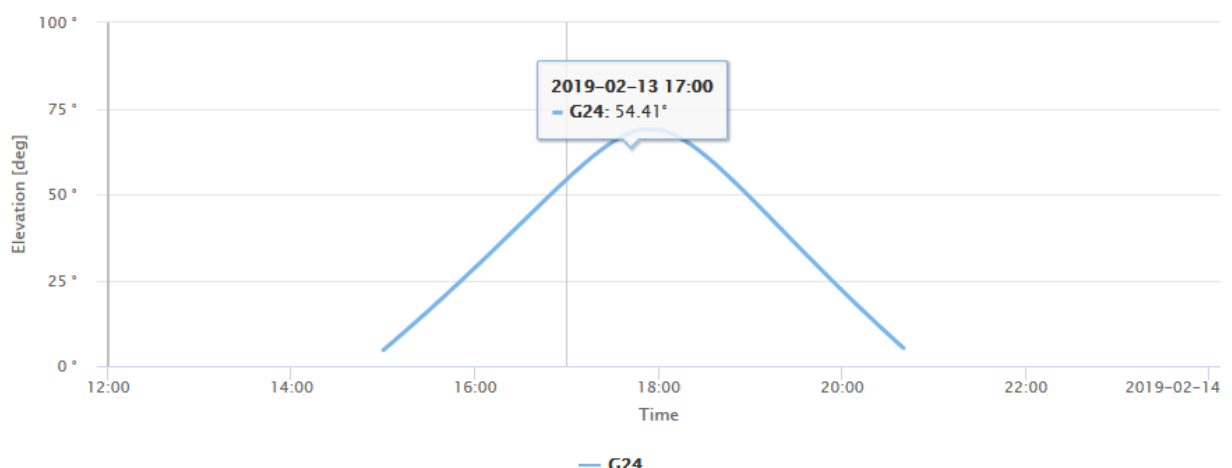


Рисунок 8 – График угла места

Из рисунка 8 видно, что лучше всего спутник был виден в районе 18:00.

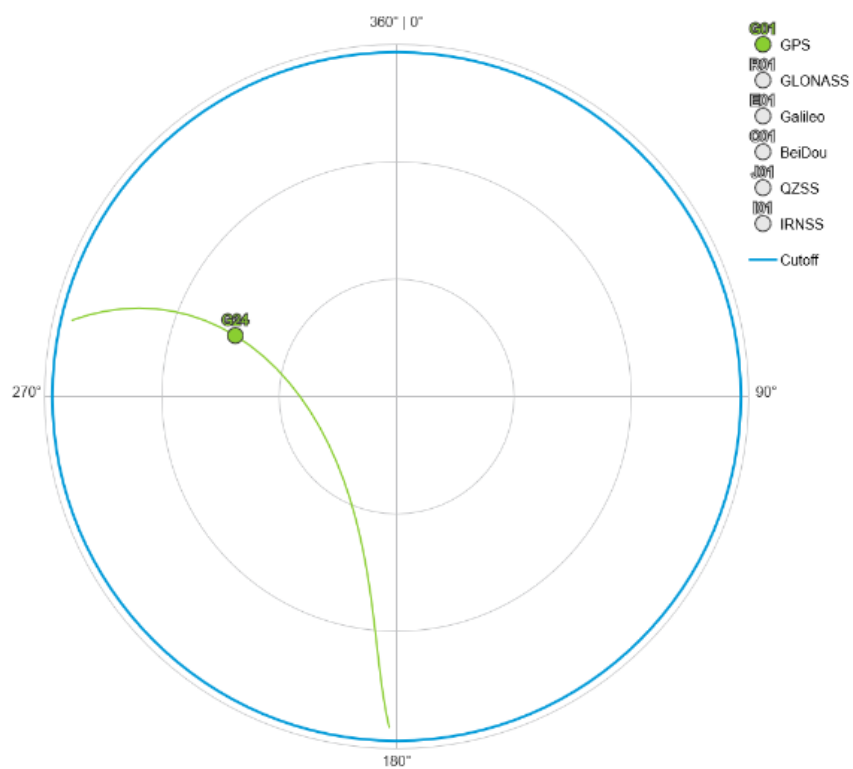


Рисунок 9 – Sky View

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. В качестве источника эфемерид использовать предоставленный на предыдущем этапе файл с логами за 13.02.19

2.2 Расчет текущего времени по шкале GPS

13 февраля в 18:00 спутник был виден лучше всего. Для этого времени .

Время по шкале GPS записывается в формате: WN:TOW.

Где WN – номер недели начиная с 6 января 1980 г, по модулю 1024 (10 бит);

TOW – количество секунд от начала текущей недели.

WN = 2040.

Тогда TOW будет равняться $TOW = 2 * 24 * 3600 + 18 * 3600 = 172800 + 64800 = 237600$

Время по шкале GPS: 2040: 237600.

2.3 Анализ данных .nav файла эфемерид

Файл эфемерид на заданный момент времени для спутника №24 выглядит следующим образом:

```
2.10          N: GPS NAV DATA          RINEX VERSION / TYPE
CONVBIN 2.4.3          20190220 192733 UTC PGM / RUN BY / DATE
log: C:\Users\Anna\Documents\Linux\LOGZZZ\BINR_evening.bin
COMMENT
format: NVS BINR          COMMENT
                          END OF HEADER

24 19  2 13 14  0  0.0 -.621117651463E-04 -.568434188608E-12 .000000000000E+00
      .131070000000E+05 .620625000000E+02 .499342190860E-08 .951895577860E+00
      .344403088093E-05 .816986314021E-02 .973232090473E-05 .515373414612E+04
      .309600000000E+06 -.596046447754E-07 .198684878318E+01 .104308128357E-06
      .939133299974E+00 .178218750000E+03 .594741691139E+00 -.834249035535E-08
      -.275011455331E-09 .100000000000E+01 .204000000000E+04 .000000000000E+00
      .240000000000E+01 .000000000000E+00 .279396772385E-08 .510000000000E+02
      .302988000000E+06 .000000000000E+00
```

В соответствии с форматом RINEX-файлов в таблицу были сведены необходимые для расчета положения КА параметры орбиты. Данные представлены в таблице 1.

t_{oe} , опорное время эфемерид	0. 309600000000E+06
\sqrt{a} , корень из большой полуоси	0. 515373414612E+04
e , эксцентриситет	0. 816986314021E-02
M_0 , средняя аномалия	0. 951895577860E+00
ω , аргумент перигея	0. 594741691139E+00
i_0 , наклонение орбиты	0. 939133299974E+00
Ω_0 , долгота восходящего узла	0. 198684878318E+01
Δn , среднее движение	0. 499342190860E-08
IDOT, скорость изменения наклонения	-0. 275011455331E-09
$\dot{\Omega}$, скорость изменения долготы узла	-0. 834249035535E-08
c_{uc} , поправки к истинной аномалии	0. 344403088093E-05
c_{us} ,	0. 973232090473E-05
c_{rc} , поправки к радиусу орбиты	0. 178218750000E+03
c_{rs} ,	0. 620625000000E+02
c_{ic} , поправки к наклонению	-0. 596046447754E-07
c_{is} ,	0. 104308128357E-06

Таблица 1 – Значения использованных эфемерид

По данным, представленным в Таблице 1 в соответствии с алгоритмом расчета координат в системе GPS (см. приложение) были получены координаты КА №24 на заданный интервал времени (18:00:00 – 00:00:00). Полученные траектории для двух случаев (Первый – в неинерциальной СК, с учетом вращения земли; второй – в инерциальной СК, где ось координат зафиксирована) изображены на рисунке 10:

2.4 Результат моделирования в среде MatLab

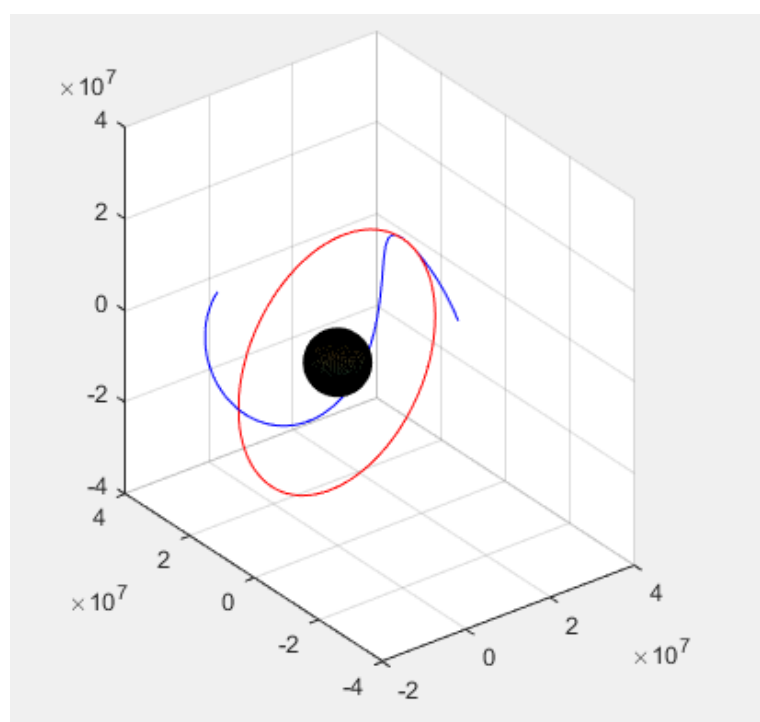


Рисунок 10 – Траектория движения спутника, голубая линия – для инерциальной СК, красная – для неинерциальной СК

На Рисунке 11 изображен вид SkyView на заданный момент времени, полученный с помощью GNSS Planing Online SkyView, а на Рисунке 12 изрбражен SkyView, построенный в MATLAB:

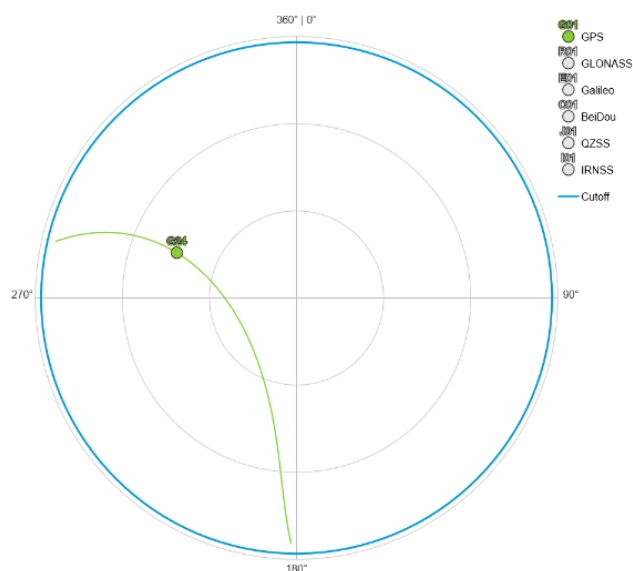


Рисунок 11 – SkyView на заданный момент времени

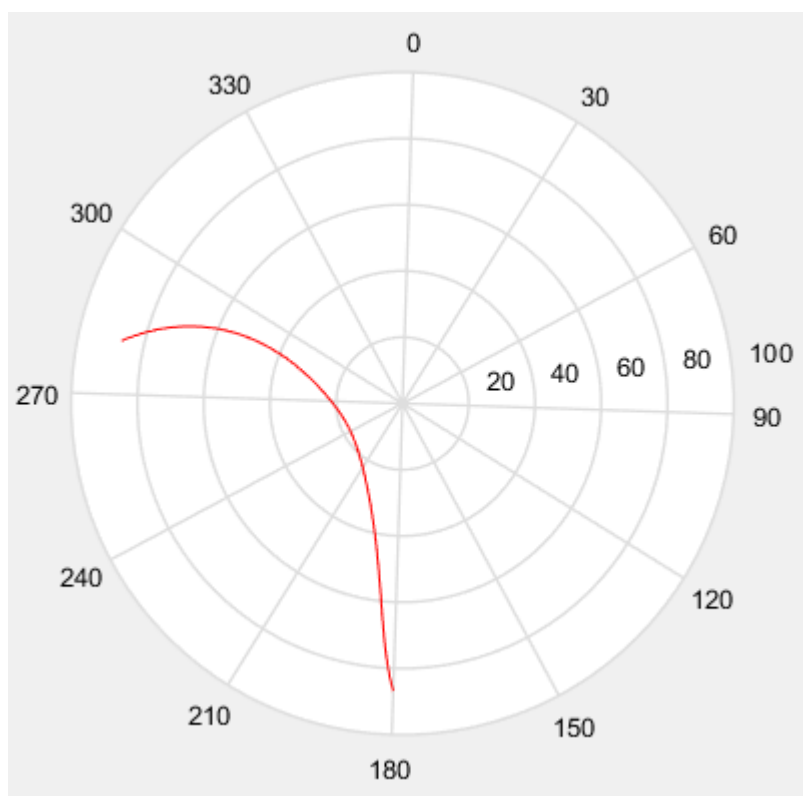


Рисунок 12 – Траектория заданного спутника, полученная при расчете в системе MatLab

Видно, что при сравнении графиков SkyView, представленных на Рисунке 11 и Рисунке 12 можно утверждать, что заданный спутник в анализируемый интервал времени был виден и его сигнал принимался антенной, расположенной

на крыше корпуса Е МЭИ. Видно, что полученный SkyView совпадает с построенным в MATLAB по эфемеридам НКА 24.

2.5 Код программы моделирования в среде MatLab

Ниже приведен код программы в Matlab, производящий расчет траектории КА.

```
clear all;
close all;
tic;
% считывание эфемерид из файла BINR_evening.nav на НКА с заданным номером
delimiterIn = ' ';
headerlinesIn = 5;
filename = 'C:\Users\Anna\Documents\Linux\LOGZZZ\BINR_evening.nav';
A = importdata(filename,delimiterIn,headerlinesIn);
ephemerides = A.data(:,1:6);
PRN = 24;
k = 0;
numbers = zeros(size(ephemerides,1)/8);
time = zeros(size(ephemerides,1)/8);
for i = 1:8:size(ephemerides,1)
    if ephemerides(i,1) == PRN
        k = k + 1;
        numbers(k) = i;
        time(k) = ephemerides(i,5);
    end
end

nomer = 1;
% запись необходимых эфемерид в матрицу
R = ephemerides((numbers(nomer)+1):(numbers(nomer)+5),:);
% запись эфемерид в текстовый файл
f = fopen('C:\Users\Anna\Documents\Linux\LOGZZZ\ephemeridy.txt','w');
fprintf(f,'%f %f %f %f \n',R);
fclose(f);
% задание эфемерид для расчета
toe = R(3,1);
sqrt_a = R(2,4);
e = R(2,2);
M0 = R(1,4);
omega = R(4,3);
i0 = R(4,1);
OMEGA0 = R(3,3);
delta_n = R(1,3);
IDOT = R(5,1);
OMEGADOT = R(4,4);
cuc = R(2,1);
cus = R(2,3);
crc = R(4,2);
crs = R(1,2);
cic = R(3,2);
cis = R(3,4);
% константы
mu=3.986004418E14;
OMEGADOTe = 7.2921151467E-5;
% АЛГОРИТМ РАСЧЕТА ПОЛОЖЕНИЯ СПУТНИКА В GPS
A = (sqrt_a)^2;
```

```

t = ((84-3)*3600):0.1:((96-3)*3600); % тут за 12 часов надо
N = size(t,2);
for k = 1:N
    tk = t(k) - toe;
    if tk > 302400
        tk = tk - 604800;
    end
    if tk < -302400
        tk = tk + 604800;
    end

    Mk = M0 + (sqrt(mu/A^3) + delta_n) * tk;
    % решение уравнения Кеплера
    m=1;
    w(1)=0;
    while 1
        w(m+1)=Mk+e*sin(w(m));
        if abs(w(m+1)-w(m))<10^(-5)
            break
        end
        m=m+1;
    end
    Ek=w(m+1);
    vk = atan2((sqrt(1-e^2)*sin(Ek)),cos(Ek)-e);
    uk = omega + vk + cuc*cos(2*(omega+vk))+cus*sin(2*(omega+vk));
    rk = A*(1-e*cos(Ek))+crc*cos(2*(omega+vk))+crs*sin(2*(omega+vk));
    ik = i0+IDOT*tk+cic*cos(2*(omega+vk))+cis*sin(2*(omega+vk));
    x_k = rk*cos(uk);
    y_k = rk*sin(uk);

    lk = OMEGA0 - OMEGADOTe*toe+(OMEGADOT-OMEGADOTe)*tk;
    lk1 = OMEGA0 - OMEGADOTe*toe;
    % расчет координат в неинерциальной системе
    x(k) = x_k*cos(lk)-y_k*cos(ik)*sin(lk);
    y(k) = x_k*sin(lk)+y_k*cos(ik)*cos(lk);
    z(k) = y_k*sin(ik);
    % расчет координат в инерциальной системе
    x1(k) = x_k*cos(lk1)-y_k*cos(ik)*sin(lk1);
    y1(k) = x_k*sin(lk1)+y_k*cos(ik)*cos(lk1);
    z1(k) = y_k*sin(ik);
end
% построение графиков
[X,Y,Z] = sphere(50);
surf(X*6400000,Y*6400000,Z*6400000)
hold on
plot3(x,y,z,'b')
plot3(x1,y1,z1,'r')
grid on
% colormap(gray)
daspect([1 1 1])
% сохранение траектории в текстовый файл
F = [x',y',z'];
f = fopen('C:\Users\Anna\Documents\Linux\LOGZZZ\testmatlab.txt','w');
fprintf(f,'%f %f %f\n',F');
fclose(f);
hold on
% координаты Москвы, рассчитанные в первом этапе
position = [2835903.37, 2192470.92, 5265921.01];
plot3(position(1),position(2),position(3),'*');
lat = atan(position(3)/sqrt(position(1)^2+position(2)^2));
lon = acos(position(1)/sqrt(position(1)^2+position(2)^2));
h = 180.59;

```

```

% пересчет координат из глобальной неинерциальной СК к локальной (ENU)
for k = 1:N
    [x0(k) y0(k) z0(k)] =
ecef2enu(x(k),y(k),z(k),lat,lon,h,wgs84Ellipsoid,'radians');
    if z0(k)>0
        teta(k) = atan(sqrt(x0(k)^2+y0(k)^2)/z0(k));
        r(k) = sqrt(x0(k)^2+y0(k)^2+z0(k)^2);
        if x0(k) > 0
            phi(k) = atan(y0(k)/x0(k));
        elseif (x0(k)<0)&&(y0(k)>0)
            phi(k) = -atan(y0(k)/x0(k))+3*pi/2;
        elseif (x0(k)<0)&&(y0(k)<0)
            phi(k) = -atan(y0(k)/x0(k))-pi/2;
        end
    else teta(k) = NaN;
        r(k) = NaN;
        phi(k) = NaN;
    end
    k;
end

% Построение SkyView
figure;
polar(phi,teta*180/pi-pi,'r')
toc;

```

Этап 3. Реализация

3.1 Постановка задачи программной реализации

Требуется разработать на языке C/C++ 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 Реализация на языке C++

Реализуем написанный на предыдущем этапе код на языке C++. Будем создавать библиотеку для расчета траектории НКА сразу в каркасе программного модуля на базе системы QMake (QT).

Наш код будет иметь две функции:

- 1) Функция решения уравнения Кеплера, входными параметрами для которой являются: средняя аномалия, эксцентриситет, точность (для итерационного метода).
- 2) Функция расчета положения спутника. Входные параметры: эфемериды, время в секундах от начала времени эфемерид, на которое необходимо рассчитать координаты НКА. Выходным параметром является три массива координат X,Y,Z.

Функция решения уравнения Кеплера вызывается в ходе расчета координат НКА.

Итого, были получены следующие коды:

Kepler.h

```
#ifndef KEPLER_H
#define KEPLER_H

double kepler(double Mk, double e, double eps);

#endif /* #ifndef KEPLER_H */
```



```

#ifndef GPSSVPOS_H
#define GPSSVPOS_H

typedef struct {
    double X;
    double Y;
    double Z;
} Vec3D;

typedef struct {
    double toe;
    double sqrt_a;
    double e;
    double M0;
    double omega;
    double i0;
    double OMEGA0;
    double delta_n;
    double IDOT;
    double OMEGADOT;
    double cuc;
    double cus;
    double crc;
    double crs;
    double cic;
    double cis;
    double mu;
    double OMEGADOTe;
} Eph;

int getSVPos(double t, Eph eph, Vec3D* pos);

#endif /* #ifndef KEPLER_H */

```

```

#include <math.h>
#include <kepler.h>

double kepler(double Mk, double e, double eps)
{
    int m = 0;
    double w[10];
    w[0] = 0;
    do
    {
        w[m+1] = Mk + e*sin(w[m]);
        m++;
    } while (fabs(w[m]-w[m-1])>eps);
    return w[m];
}

```

```

#include <gpssvpos.h>
#include <kepler.h>
#include <math.h>

int getSVPos(double t, Eph eph, Vec3D* pos) {
    double Mk = eph.M0 + (sqrt(eph.mu/pow(eph.sqrt_a,6)) + eph.delta_n)*t;
    double eps = pow(10,-5);
    double Ek = kepler(Mk,eph.e,eps);
    double vk = atan2((sqrt(1 - pow(eph.e,2))*sin(Ek)),cos(Ek) - eph.e);
    double uk = eph.omega + vk + eph.cuc*cos(2*(eph.omega + vk)) +
    eph.cus*sin(2*(eph.omega + vk));
    double rk = pow(eph.sqrt_a,2)*(1-
    eph.e*cos(Ek))+eph.crc*cos(2*(eph.omega+vk))+eph.crs*sin(2*(eph.omega+vk));
    double ik =
    eph.i0+eph.IDOT*t+eph.cic*cos(2*(eph.omega+vk))+eph.cis*sin(2*(eph.omega+vk));
    double x_k = rk*cos(uk);
    double y_k = rk*sin(uk);
    double lk = eph.OMEGA0 - eph.OMEGADOTe*eph.toe+(eph.OMEGADOT-
    eph.OMEGADOTe)*t;
    pos->X = x_k*cos(lk)-y_k*cos(ik)*sin(lk);
    pos->Y = x_k*sin(lk)+y_k*cos(ik)*cos(lk);
    pos->Z = y_k*sin(ik);
    return 0;
}

```

На основании составленных кодов был получен головной код main.cpp, представленный ниже

```

#include <iostream>
#include <fstream>
#include <gpssvpos.h>
#include <kepler.h>
#include <ctime>

using namespace std;

int main()
{
    Eph eph;
    eph.toe = 309600;
    eph.sqrt_a = 5.15373414612e+03;
    eph.e = 0.00816986314021;
    eph.M0 = 0.951895577860;
    eph.omega = 0.594741691139;
    eph.i0 = 0.939133299974;
    eph.OMEGA0 = 1.98684878318;
    eph.delta_n = 4.99342190860e-09;
    eph.IDOT = -2.75011455331e-10;
    eph.OMEGADOT = -8.34249035535e-09;
    eph.cuc = 3.44403088093e-06;
    eph.cus = 9.73232090473e-06;
    eph.crc = 1.78218750000e+02;
    eph.crs = 62.0625000000;
    eph.cic = -5.96046447754e-08;
}

```

```

eph.cis = 1.04308128357e-07;
eph.mu = 3.986004418E14;
eph.OMEGADOTe = 7.2921151467E-5;

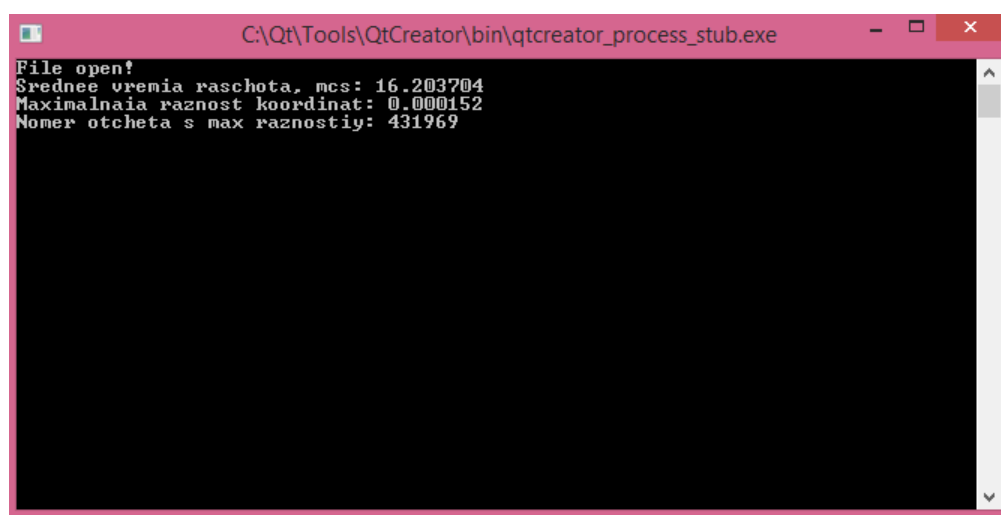
time_t start, end;
double t = (84-3)*3600 - eph.toe;
double delt = 0.1;

Vec3D pos;
double *koord_matlab = new double[3];
double max_del = 0;
int i_max = 0;
std::ofstream out;
out.open("C:\\rez_cpp.txt");
time(&start);
std::ifstream in("C:\\testmatlab.txt");
if (!in)
{
    std::cout << "File not open!" << std::endl;
} else {
    std::cout << "File open!" << std::endl;
}
for (int i = 0; i < (12*3600/delt); i++)
{
    getSVPos(t, eph, &pos);
    t += delt;
    std::string koord_str1 = std::to_string(pos.X);
    std::string koord_str2 = std::to_string(pos.Y);
    std::string koord_str3 = std::to_string(pos.Z);
    out << koord_str1 << " " << koord_str2 << " " << koord_str3 <<
std::endl;
    in >> koord_matlab[0] >> koord_matlab[1] >> koord_matlab[2];
    if (abs(pos.X-koord_matlab[0]) > max_del)
    {
        max_del = abs(pos.X-koord_matlab[0]);
        i_max = i;
    }
    if (abs(pos.Y-koord_matlab[1]) > max_del)
    {
        max_del = abs(pos.Y-koord_matlab[1]);
        i_max = i;
    }
    if (abs(pos.Z-koord_matlab[2]) > max_del)
    {
        max_del = abs(pos.Z-koord_matlab[2]);
        i_max = i;
    }
}
time(&end);
in.close();
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_dell = std::to_string(max_del);
cout << "Maximalnaia raznost koordinat: " << max_dell << std::endl;
std::string imax = std::to_string(i_max);
cout << "Nomer otcheta s max raznostiy: " << imax << std::endl;
out.close();
in.close();
}

```

Таким образом, данная программа считает максимальную разницу между расчетами, производимыми на языке C++, и полученными в среде MatLab результатами. Результат выводится в текстовое окно, представленное на Рисунке 13.

Видно, что время расчета и сравнения полученных результатов составило 16,2 мс, а точность совпадения результатов оставила 10^{-4} степени.



```
C:\Qt\Tools\QtCreator\bin\qtcreator_process_stub.exe
File open!
Srednee vremia raschota, mcs: 16.203704
Maximalnaia raznost koordinat: 0.000152
Nomer otcheta s max raznostiy: 431969
```

Рисунок 13 – Результат сравнения решения в QT (язык C++) и MatLab

3.3 Контроль утечек

Для контроля и устранения утечек памяти в среде QT используется утилита Необ. Скачав и установив ее с официального сайта, а также подключив в окне Анализ (Рисунок 14) было получено, что данный код утечек не дает (окно отладчика после запуска анализа утечек не выдало сообщений о проблеме).

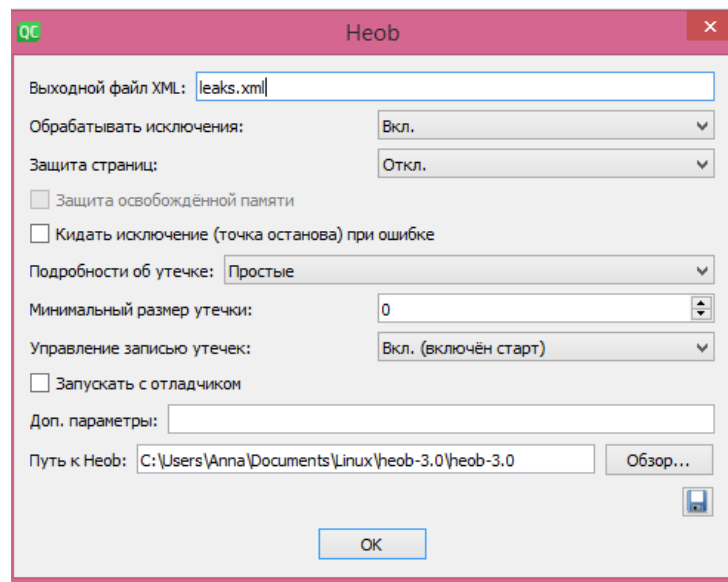


Рисунок 14 – Подключение контроля утечек Neob

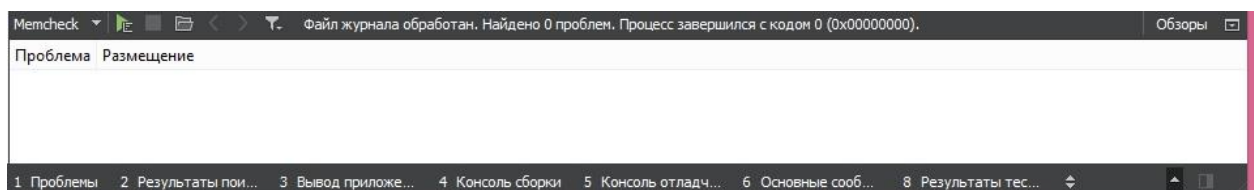


Рисунок 15 – Отсутствие ошибок после использования функции поиска утечек Neob

Чтобы проверить, действительно ли программа анализа утечек работает корректно в исходном коде удалили строчки:

```
delete[] koord_matlab;
koord_matlab = nullptr;
```

В данных строках производится удаление динамического массива и обнуление указателя на массив.

На Рисунке 16 представлено окно консоли отладчика, которое после анализа на утечки заполнилось ошибками. Это доказывает, что система верно регистрирует наличие утечек при наличии ошибок подобного рода в коде.

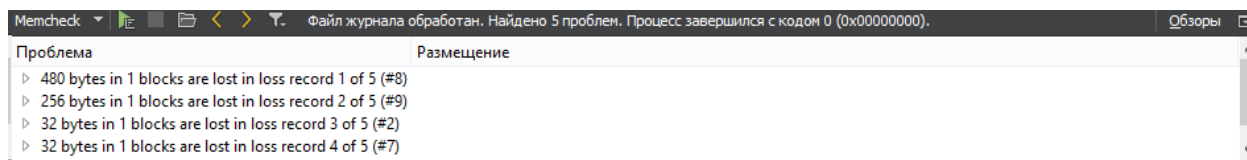


Рисунок 16 – Проверка работы функции поиска утечек Neob

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. Была изучена структура принимаемого от спутника альманаха, а также методы простейшего моделирования его движения с помощью онлайн-сервиса Trimble GNSS Planning Online. После, при использовании программы RTKLib были получены эфемериды собственного спутника по данным RTKNAVI из состава RTKLIB. Во втором этапе курсового проекта было произведено моделирование траектории движения спутника по полученным эфемеридам из файла формата .nav. На этапе выполнения третьего и завершающего этапа курсового проекта была произведена вторичная обработка движения спутника с помощью кода, написанного на языке программирования C++ при помощи программы QMake. Полученный результат был очень близок к результату, полученному во втором этапе, а также был произведен анализ утечек, давший отрицательный результат.