

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 практике

Наименование
практики: Производственная практика:
научно-исследовательская работа

СТУДЕНТ

/ Карнаухов А.А.
(подпись) (Фамилия и инициалы)
Группа ЭР-15-16
(номер учебной группы)

**ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ ПО
ПРАКТИКЕ**

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

/ Шатилов А.Ю.
/
(подпись) (Фамилия и инициалы члена комиссии)

/ Куликов Р.С.
/
(подпись) (Фамилия и инициалы члена комиссии)

**Москва
2021**

ВВЕДЕНИЕ

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

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.

Конечная цель всего курсового проекта - получить библиотеку функций на Си++, позволяющую рассчитывать положение спутника Beidou по его эфемеридам.

Исходные данные: PRN спутника Beidou – C09

Этап 1. Использование сторонних средств

Описание этапа

На первом этапе подготовим вспомогательные данные для разработки: эфемериды и оценки положения спутника от сторонних сервисов (чтобы было с чем сравниваться на след. этапах).

На крыше корпуса Е МЭИ установлена трехдиапазонная антенна Narxон НХ-СХ601А. Она через 50-метровый кабель, сплиттер, bias-tee и усилитель подключена к трем навигационным приемникам:

- Javad Lexion 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

Определим какому спутнику соответствует выданный PRN спутника.

9	Компас IGSO-4	C09	26.07.2011 21:44	CZ-3A	2011-038A	37763	Геосинхронная, накл. 55°; 95° в. д.	действующий
---	---------------	-----	------------------	-------	-----------	-------	--	-------------

Рисунок 1 — Состав орбитальной группировки космической навигационной системы Beidou на 10 марта 2020 года [1]

Номер спутника C09 соответствует спутнику Компас IGSO-4, номер по спутниковому каталогу НОРАД (или SCN) равен 37763. Для проверки этой информации обратимся к ресурсу «Информационно-аналитического центра координатно-временного и навигационного обеспечения» [2].

C09	37763	IGSO-4	BDS-2	27.07.11	3515	Используется по ЦН
-----	-------	--------	-------	----------	------	--------------------

Рисунок 2 — Данные о состоянии космических аппаратов Beidou на 02.03.21 (источник «Информационно-аналитического центра координатно-временного и навигационного обеспечения»)

По рисункам 1 и 2 можно увидеть, что данные совпадают.

1.1. Определение формы орбиты и положения спутника

Определим формы орбиты и положения спутника на ней на начало рассматриваемого интервала времени по данным сервиса Celestrak: общий вид + положение спутника на 18:00 МСК 16 февраля 2021, так, чтобы было видно подспутниковую точку и время.

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.

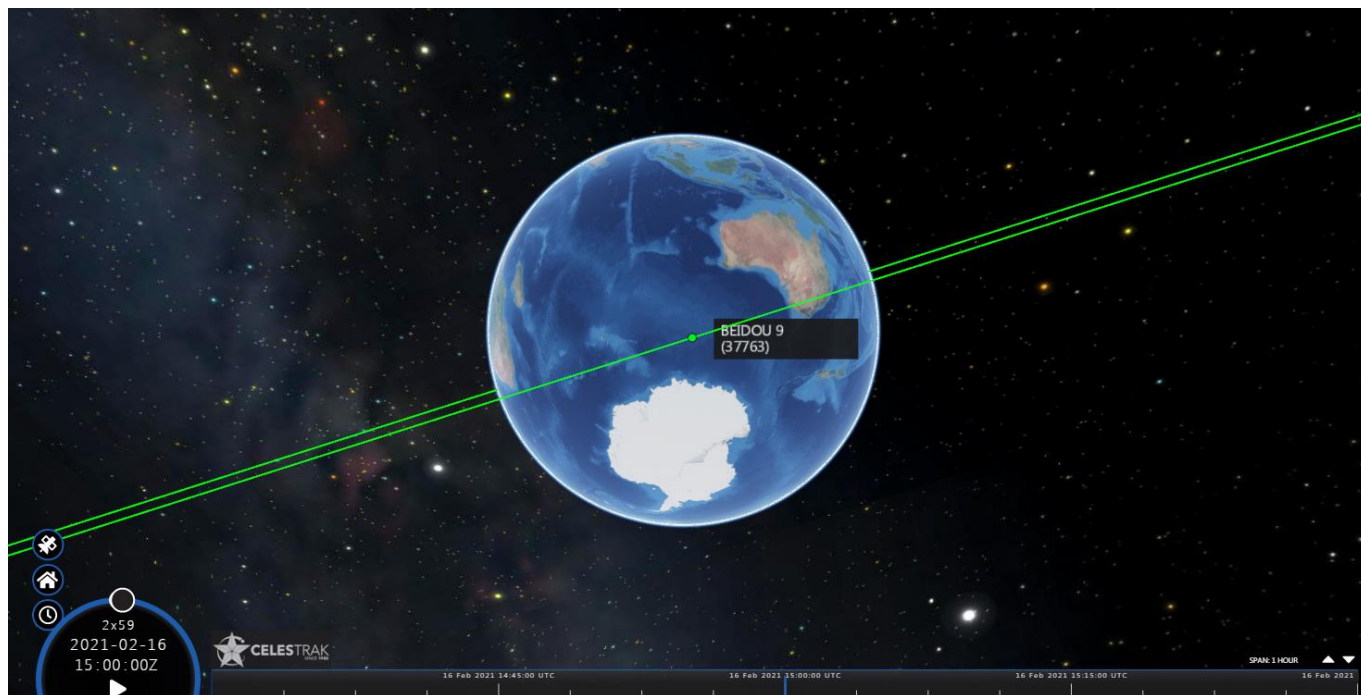


Рисунок 3 — Модель сервиса Celestrak, видно подспутниковую точку и время

1.2. Расчет графика угла места собственного спутника от времени

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.

Установили приблизительные координаты местоположения антенны и границы времени, также выбрали спутник C09.

Рисунок 4 — Экран настроек Trimble GNSS Planning Online

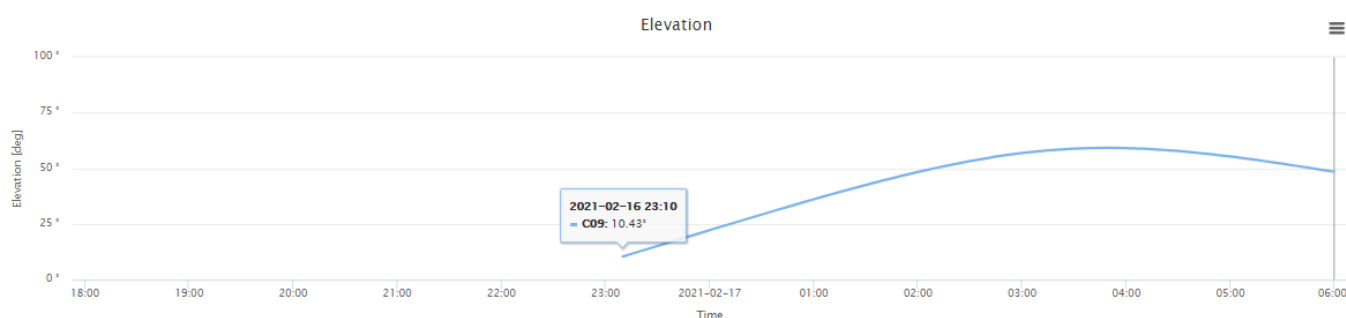


Рисунок 5 — График угла места спутника C09 от времени

По рисунку 5 видно, что спутник находился в зоне видимости в промежутке времени с 23:10 до 6:00.

1.3. Расчет диаграммы угла места и азимута спутника

Рассчитаем диаграммы угла места и азимута спутника (SkyView, он же SkyPlot) по данным Trimble GNSS Planning Online на интервал времени с 18:00 МСК 16 февраля до 06:00 МСК 17 февраля 2021 года.

Пронаблюдаем траекторию движения спутника.

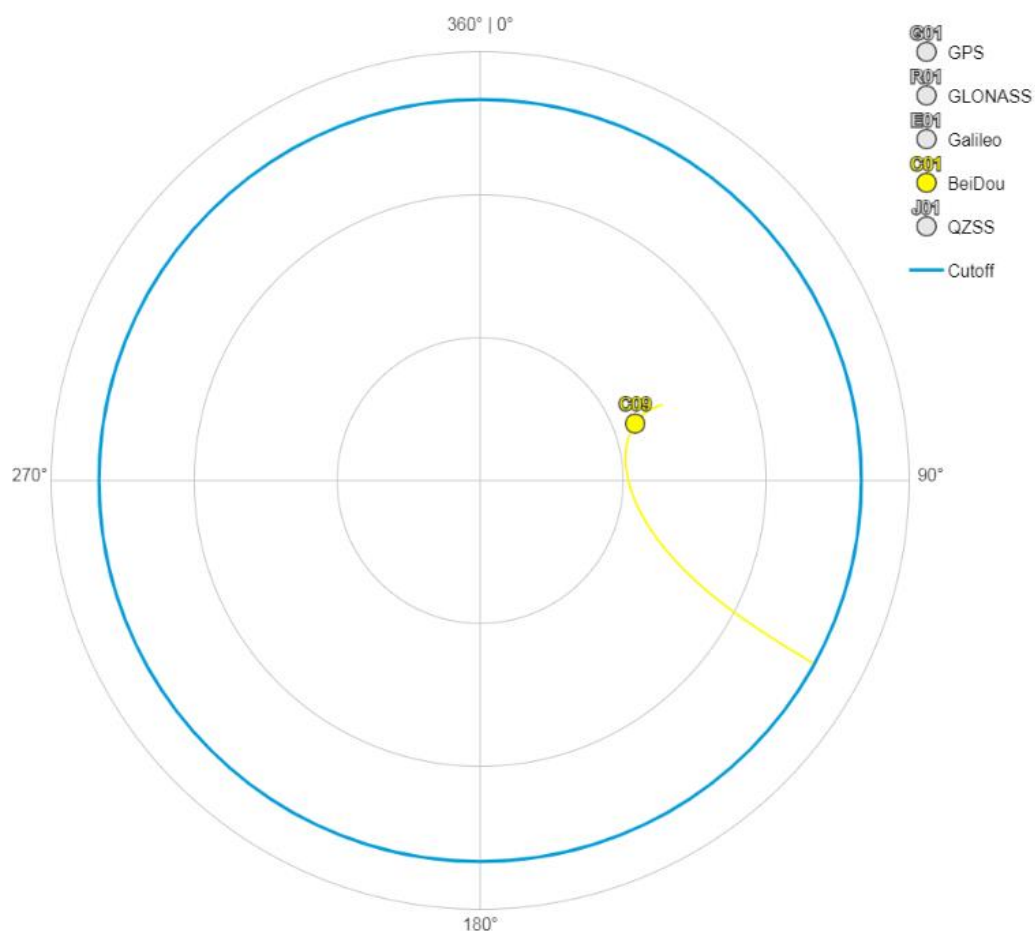


Рисунок 6 — Траектория движения спутника C09, построенная с помощью SkyPlot.

По рисунку 6 видно, что в конце наблюдаемого интервала времени спутник C09 все еще находился в зоне видимости.

1.4. Формирование списка и описание параметров

Сформируем список и описание параметров, входящих в состав эфемерид в сигнале B1I Beidou. Сформируем список эфемерид [3]:

Таблица 1 — Описание параметров эфемерид

Параметры	Определение
t_{oe}	Исходное время эфемерид
\sqrt{A}	Квадратный корень из большой полуоси
e	Эксцентриситет
ω	Аргумент перигея
$\square n$	Среднее отклонение движения от расчетного значения
M_0	Средняя аномалия в исходное время
Ω_0	Долгота восходящего узла орбитальной плоскости, вычисленная по исходному времени
$\dot{\Omega}$	Скорость прямого восхождения
i_0	Угол наклона в исходное время
$IDOT$	Скорость угла наклона
C_{uc}	Амплитуда косинусного гармонического корректирующего члена к аргументу широты
C_{us}	Амплитуда синусного гармонического корректирующего члена к аргументу широты
C_{rc}	Амплитуда косинусного гармонического корректирующего члена к радиусу орбиты
C_{rs}	Амплитуда синусного гармонического корректирующего члена к радиусу орбиты
C_{ic}	Амплитуда косинусного гармонического корректирующего члена к углу наклона
C_{is}	Амплитуда синусного гармонического корректирующего члена к углу наклона

Таблица 2 — Значения параметров эфемерид спутника C09

Параметр	Обозначение	Значение	Размерность
SatNum	PRN	9	-
toe	t_{oe}	244800000.000	мс
Crs	C_{rc}	-4.9843750000000000e+01	рад
Dn	∇n	7.87532799390033844e-13	рад/мс
M0	M_0	2.34731910518265385e+00	рад
Cuc	C_{uc}	-1.24704092741012573e-06	рад
e	e	7.69855396356433630e-03	-
Cus	C_{us}	2.29864381253719330e-05	рад
sqrtA	\sqrt{A}	6.49307452583312988e+03	м ^{1/2}
Cic	C_{ic}	-6.98491930961608887e-08	рад
Omega0	Ω_0	6.79084253670661386e-01	рад
Cis	C_{is}	8.84756445884704590e-08	рад
i0	i_0	9.50974837701972997e-01	рад
Crc	C_{rc}	-4.67468750000000000e+02	рад
omega	ω	-2.38950527529739887e+00	рад
OmegaDot	$\dot{\Omega}$	-1.75185868623179542e-12	рад/мс
iDot	\dot{I}	6.78599694972560970e-15	рад/с
Tgd	T_{GD}	2.0900000000000000e+05	мс
toc	t_{oc}	2.4480000000000000e+08	мс
af2	a_{f2}	0.0000000000000000e+00	мс/мс ²
af1	a_{f1}	-2.67101896156418661e-11	мс/мс
af0	a_{f0}	7.28702425956726074e-01	мс
URA	-	0	-
IODE	-	257	-
IODC	-	0	-
codeL2	-	0	-
L2P	-	0	-
WN	-	789	-

Этап 2. Моделирование

Описание этапа

Эфемериды - параметры некоторой модели движения спутника. В разных ГНСС эти модели разные, а значит отличается и формат эфемерид, и алгоритмы расчета положения спутника.

Одна из самых простых и удобных моделей - в системе GPS. Beidou наследует данную модель.

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, соответствующим номеру студента по списку. 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.

Приведём таблицу эфемерид, используемых в данном этапе:

Таблица 3 — Значения параметров эфемерид спутника C09

Параметр	Обозначение	Значение	Размерность
SatNum	PRN	9	-
toe	t_{oe}	244800000.000	мс
Crs	C_{rc}	-4.9843750000000000e+01	рад
Dn	\dot{n}	7.87532799390033844e-13	рад/мс
M0	M_0	2.34731910518265385e+00	рад
Cuc	C_{uc}	-1.24704092741012573e-06	рад
e	e	7.69855396356433630e-03	-
Cus	C_{us}	2.29864381253719330e-05	рад
sqrtA	\sqrt{A}	6.49307452583312988e+03	м ^{1/2}
Cic	C_{ic}	-6.98491930961608887e-08	рад
Omega0	Ω_0	6.79084253670661386e-01	рад
Cis	C_{is}	8.84756445884704590e-08	рад
i0	i_0	9.50974837701972997e-01	рад
Crc	C_{rc}	-4.6746875000000000e+02	рад
omega	ω	-2.38950527529739887e+00	рад
OmegaDot	$\dot{\Omega}$	-1.75185868623179542e-12	рад/мс
iDot	\dot{I}	6.78599694972560970e-15	рад/с
Tgd	T_{GD}	2.0900000000000000e+05	мс
toc	t_{oc}	2.4480000000000000e+08	мс
af2	a_{f2}	0.0000000000000000e+00	мс/мс ²
af1	a_{f1}	-2.67101896156418661e-11	мс/мс
af0	a_{f0}	7.28702425956726074e-01	мс
URA	-	0	-
IODE	-	257	-
IODC	-	0	-
codeL2	-	0	-

L2P	-	0	-
WN	-	789	-

2.1. Алгоритм расчёта

В таблице 3 приведены параметры эфемерид вещания Beidou для вычисления их спутниковых координат в любую эпоху наблюдения. Эти параметры периодически обновляются и не должны использоваться вне установленного времени (около четырех часов), поскольку ошибка экстраполяции экспоненциально растет за пределами срока ее действия.

Для вычисления спутниковых координат, по навигационному сообщению, необходимо использовать следующий алгоритм.

Вычислим время t_k из исходного времени эфемерид t_{oe} (t и t_{oe} выражаются в секундах в неделе GPS):

$$t_k = t - t_{oe}$$

Если $t_k > 302400$ секунд, вычитаем 604 800 секунд из t_k . Если $t_k < -302400$ секунд, то добавляем 604800 секунд.

Вычислим среднюю аномалию для t_k ,

$$M_k = M_0 + \left(\frac{\sqrt{\mu}}{\sqrt{a^3}} + \Delta n \right) t_k$$

Решим (итеративно) уравнение Кеплера для аномалии эксцентриситета E_k :

$$M_k = E_k - e \sin(E_k)$$

Вычислим истинную аномалию ν_k :

$$\nu_k = \arctan \left(\frac{\sqrt{1-e^2} \sin(E_k)}{\cos(E_k - e)} \right)$$

Вычислим аргумент широты u_k из аргумента перигея ω , истинной аномалии v_k и поправок C_{uc} и C_{us} :

$$u_k = \omega + v_k + C_{uc} \cos(2(\omega + v_k)) + C_{us} \sin(2(\omega + v_k))$$

Вычислим радиально расстояние r_k с учётом поправок C_{rc} и C_{rs} :

$$r_k = a(1 - e \cos(E_k)) + C_{rc} \cos(2(\omega + v_k)) + C_{rs} \sin(2(\omega + v_k))$$

Вычислим наклон i_k орбитальной плоскости по наклону i_o в исходное время t_{oe} и поправкам C_{ic} и C_{is} :

$$i_k = i_o + IDOT \cdot t_k + C_{ic} \cos(2(\omega + v_k)) + C_{is} \sin(2(\omega + v_k))$$

Вычислим долготу восходящего узла λ_k (относительно Гринвича). В этом расчете используется долгота восходящего узла по исходному времени (Ω_o), поправка от видимого изменения звездного времени по Гринвичу между началом недели и опорным временем $t_k = t - t_{oe}$, а также изменение долготы восходящего узла по сравнению с исходным моментом t_{oe} :

$$\lambda_k = \Omega_o + (\dot{\Omega} - \omega_E) t_k - \omega_E t_{oe}$$

$\omega_E = 7.2921151467 \cdot 10^{-5}$ - скорость вращения Земли

Расчетные положения спутников в орбитальной плоскости:

$$\begin{cases} x_k = r_k \cos(u_k) \\ y_k = r_k \sin(u_k) \end{cases}$$

Расчетные GEO / MEO / IGSO координаты спутника в BDCS:

$$\begin{cases} X_k = x_k \cos(\lambda_k) - y_k \cos(i_k) \sin(\lambda_k) \\ Y_k = x_k \sin(\lambda_k) + y_k \cos(i_k) \cos(\lambda_k) \\ Z_k = y_k \sin(i_k) \end{cases}$$

Построим трехмерный график множества положений заданного спутника Beidou:

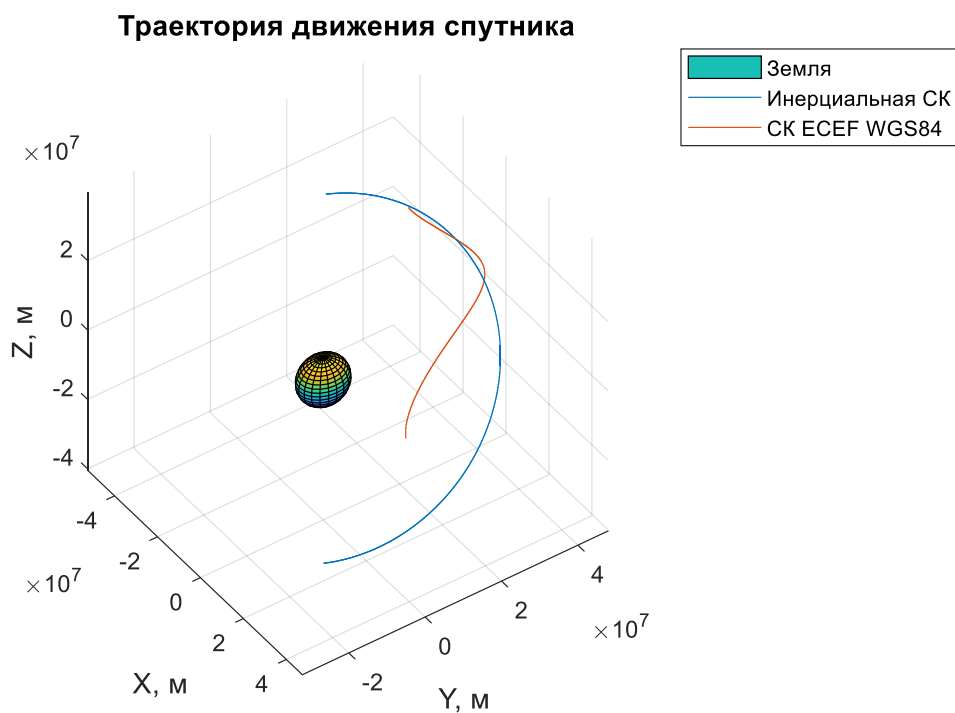


Рисунок 7 – Траектория движения спутника

Построим SkyView за интервал времени с 18:00 МСК 16 февраля до 06:00 МСК 17 февраля 2021 года:

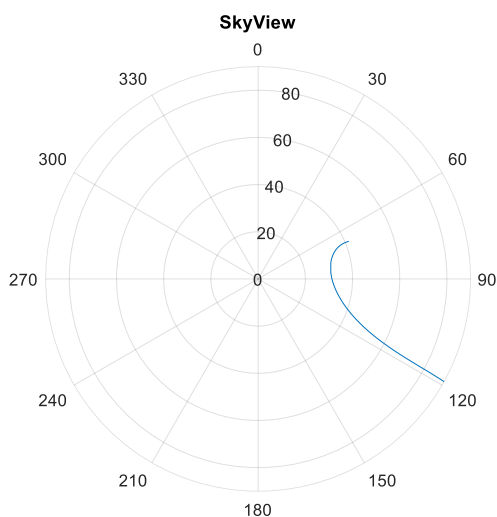


Рисунок 8 — Модель SkyView

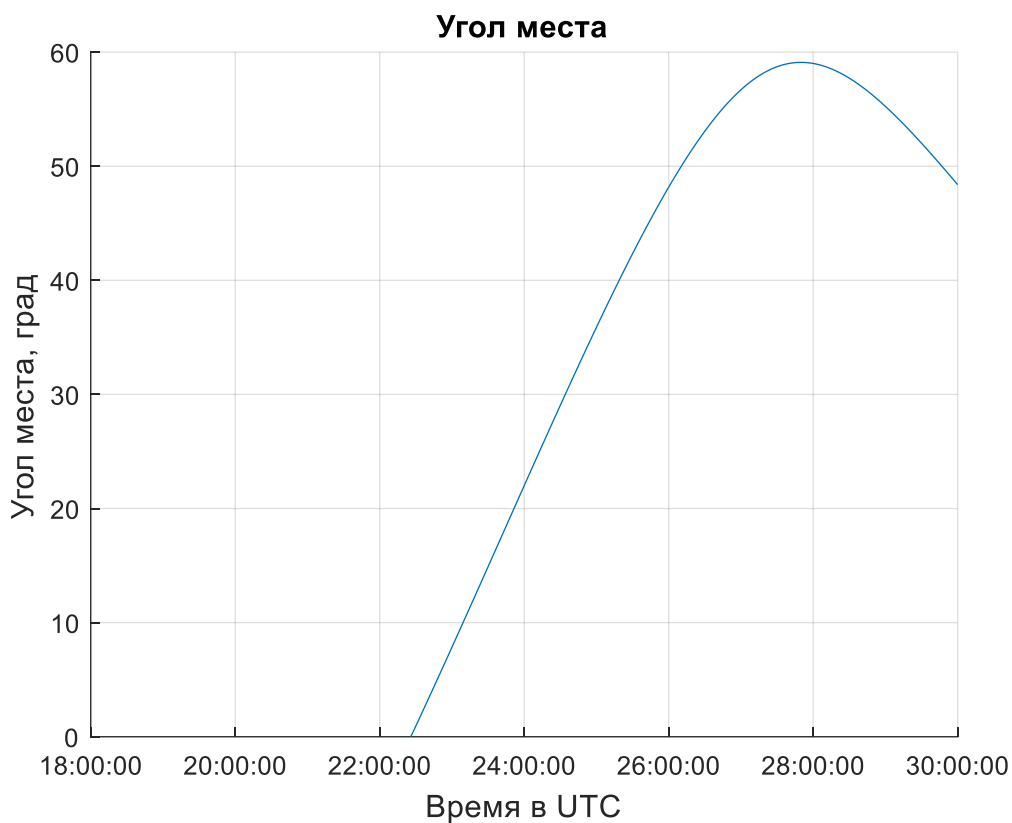


Рисунок 9 — График угла места

При сравнении рисунков, полученных на 2 этапе, с рисунками, полученными на 1 этапе, видно, что они практически совпадают. Однако, имеется погрешность, связанная с тем, что мы используем параметры эфемерид в установленном промежутке времени (около четырех часов).

Код программы приведен в приложении 1.

Этап 3. Реализация

Описание кода реализации

Задача была решена с помощью компилятора Code::Blocks. Программа представляет собой сборку из трех главных и двух заголовочных файлов. Далее будут рассмотрены данные файлы.

Файл `main.cpp` это основа проекта. В нем происходит запись координат и сравнение со значениями, полученными в Matlab на прошлом этапе. Его можно подробнее увидеть в приложении 2. Для сравнения значений, Matlab записывает координаты в файл `«coord.txt»`.

В файле `«Soordinate.cpp»` программный код происходит расчёт координат по эфемеридам, находящимся в `main.cpp`. С кодом можно ознакомиться в приложении 3.

Решение уравнения Кеплера реализуется в файле `«Kepler.cpp»`. Код можно увидеть в приложении 4. А также два заголовочных файла `«Kepler.h»` и `«Soordinate.h»`. Содержание данных файлов можно увидеть в приложениях 5 и 6.

3.1. Результат расчета

```

Консоль отладки Microsoft Visual Studio

Общее время расчёта, с: 4
Среднее время расчёта, с: 0.000093
Максимальная разность координат(значение домножено на 10^7): 5.476177
C:\Users\Андрей\Desktop\5 курс(2)\Курсач спутник\проект\Debug\ConsoleApplication
1.exe (процесс 3664) завершает работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, установите параметр
"Сервис" -> "Параметры" -> "Отладка" -> "Автоматически закрыть консоль при оста
новке отладки".
Чтобы закрыть это окно, нажмите любую клавишу:

```

Рисунок 10 – Результат работы программы

Следовательно, максимальная разность координат значения 548 нм. Общее время расчета равно 4 секундам, время одной итерации примерно 93 мкс.

3.2. Анализ работы программы

Дальнейший анализ производился с помощью Visual Studio где есть вкладка отладки. С помощью неё мы и будем проводить анализ программы.

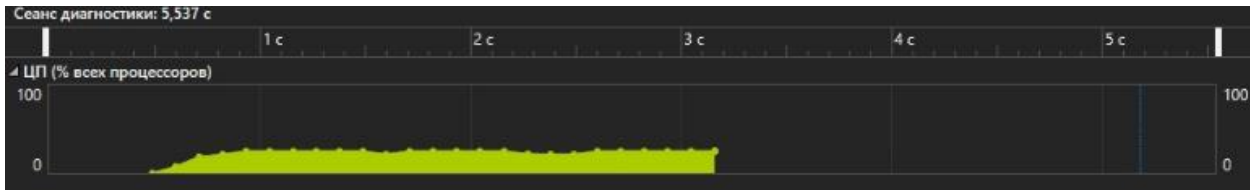


Рисунок 11 – График ЦП

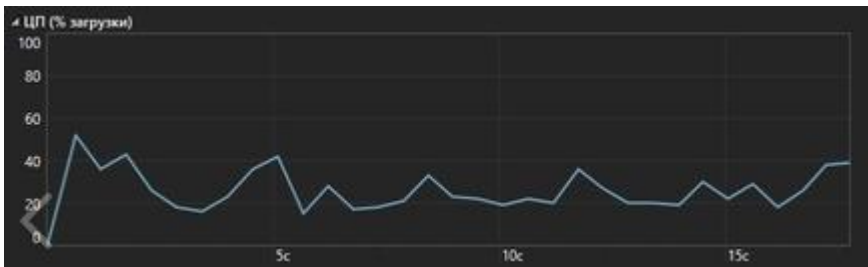


Рисунок 12 – График загрузки ЦП

Имя функции	% затраченного инклюзивного времени	% затраченного эксклюзивного времени
main	85.20	0.56
std::to_string	40.72	0.82
std::basic_string<char,std::char_traits<char>,std::allocator<char> >::basic_string<char,s...	14.24	0.83
std::basic_string<char,std::char_traits<char>,std::allocator<char> >::basic_string<char,s...	11.97	0.22
std::basic_string<char,std::char_traits<char>,std::allocator<char> >::~basic_string<cha...	5.54	0.64

Связанные представления: [Дерево вызовов](#) [Функции](#)

Функции, выполняющие максимальную индивидуальную работу

Имя	Эксклюзивное время %
__scrt_common_main	14.79
_RTC_CheckEsp	10.11
__CheckForDebuggerJustMyCode	7.65
??5?\$basic_istream@DU?\$char_traits@D@std@@@std@@@QAEAAV01@AAN@Z	7.22
_stdio_common_vsprintf	3.03

Рисунок 13 – Распределение вызовов и функций по времени

На рисунке 14 показано использование памяти процесса по времени. Утечки не были обнаружены.

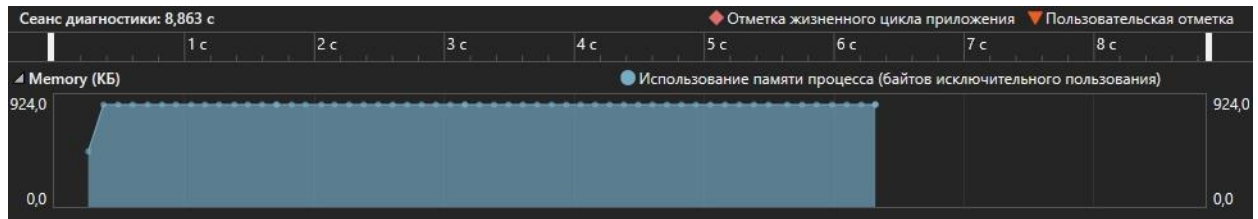


Рисунок 14 – Результат работы программы

При расчете в разных средах программирования или Matlab и C++ имеют некоторые отличия, которые, как и во времени выполнения (что можно заметить невооружённым взглядом), так и в точности.

ЗАКЛЮЧЕНИЕ

Ознакомительные работы с помощью таких сторонних средств как «Trimble GNSS Planning» и CelesTrak. Данные сервисы (средства) показывали то, что мы должны будем сделать в последующих этапах. Разработки в среде Matlab, был произведён расчет координат для заданного спутника, так же были получены сравнимые значения и графические отображения траекторий и основных характеристик. Была проанализирована разработанная программа, но и полученные ранее результаты программы как написанные на C++ так и в Matlab.

Список литературы и источников

1. Википедия. Бэйдоу - https://ru.wikipedia.org/wiki/Бэйдоу#Список_спутников

2. «Информационно-аналитический центр координатно-временного и навигационного обеспечения» - <https://www.glonass-iac.ru/BEIDOU/>
3. BeiDou Navigation Satellite System Signal In Space Interface Control Document Open Service Signal B1I (Version 3.0) - China Satellite Navigation Office February 2019

```

clear all;
clc;
close all;
a=(6.49307452583312988e+03)^2;
toe = 244800000.000 * 10^-3;
M0 =2.34731910518265385e+00 ;
dn = (7.87532799390033844e-13)/10^-3;
w= -2.38950527529739887e+00;
Cuc =-1.24704092741012573e-06;
Crc =-4.67468750000000000e+02;
Crs = -4.98437500000000000e+01;
Cic =-6.98491930961608887e-08;
Cis =8.84756445884704590e-08;
Cus =2.29864381253719330e-05;
i0 = 9.50974837701972997e-01;
IDOT =6.78599694972560970e-15;
Omega0 =6.79084253670661386e-01;
OmegaDot = -1.75185868623179542e-12;
e= 7.69855396356433630e-03;
n = 3.986004418e+14;
OMEGA_e = 7.2921151467*10^-5; %%
omegaE=OMEGA_e;%%
tstart = (24*2 + 18 - 3)*60*60;
tstop = (24*3 + 6 - 3)*60*60;
t_arr = tstart:1:tstop;
%
dt = 12*60*60;
for i = 1:length(t_arr)
coord(:, i) = CoordGPS(t_arr(i),toe, M0,e, n, a, dn,w, Cuc, Cus, Crc, Crs,Cic,
Cis,i0, IDOT, Omega0,OmegaDot, omegaE );
coord_eci_x(i)= coord(i).ECI(1);
coord_eci_y(i)=coord(i).ECI(2);
coord_eci_z(i) = coord(i).ECI(3);
coord_ecef_x(i) = coord(i).ECEF(1);
coord_ecef_y(i) = coord(i).ECEF(2);
coord_ecef_z(i) = coord(i).ECEF(3);
coord_eci(1,i) = coord(i).ECI(1);
coord_eci(2,i) = coord(i).ECI(2);
coord_eci(3,i) = coord(i).ECI(3);
coord_ecef(1,i) = coord(i).ECEF(1);
coord_ecef(2,i) = coord(i).ECEF(2);
coord_ecef(3,i) = coord(i).ECEF(3);
end
%
ppb = 1e-9;
mas = 1e-3/206264.8; % [рад]
MATRIX_WGS_84 = [-3*ppb -353*mas -4*mas;
353*mas -3*ppb 19*mas;
4*mas -19*mas -3*ppb];
crd_WGS_84 = [coord_ecef_x; coord_ecef_y; coord_ecef_z];
for i = 1:length(crd_WGS_84(1,:))
crd_WGS_84(:,i) = crd_WGS_84(:,i) + MATRIX_WGS_84 * crd_WGS_84(:,i) + [0.07; -0;
-0.77];
end
crd_WGS_84 = crd_WGS_84.'
figure (1)
E = wgs84Ellipsoid;
ellipsoid(0,0,0,E.SemimajorAxis, E.SemimajorAxis, E.SemiminorAxis);
hold on;
plot3(coord_eci_x, coord_eci_y,coord_eci_z);
hold on;
plot3(crd_WGS_84(:,1), crd_WGS_84(:,2), crd_WGS_84(:,3))
title('Траектория движения спутника', 'FontSize',12)

```

```

xlabel('X, м', 'FontSize',12)
ylabel('Y, м', 'FontSize',12)
zlabel('Z, м', 'FontSize',12)
lgd = legend('Земля', 'Инерциальная СК', 'СК ECEF WGS84');
% Географические координаты корпуса Е и их перевод в систему WGS-84
% Lantitude
N_gr = 55;
N_min = 45;
N_sec = 23.8178;
N = N_gr*pi/180 + N_min/3437.747 + N_sec/206264.8; % широта [рад]
% Longtitude
E_gr = 37;
E_min = 42;
E_sec = 12.2608;
E = E_gr*pi/180 + E_min/3437.747 + E_sec/206264.8; % долгота [рад]
H = 150; % высота [м]
llh = [N E H];
crd_PRM = llh2xyz(llh)';
% Построение SkyPlot
for i = 1:length(crd_WGS_84(:,1))
    [X(i) Y(i) Z(i)] =
ecf2enu(crd_WGS_84(i,1),crd_WGS_84(i,2),crd_WGS_84(i,3),N,E,H,wgs84Ellipsoid,'radian
s');
    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
% Skyplot
figure (2)
ax = polaraxes;
polarplot(ax,phi,teta*180/pi)
ax.ThetaDir = 'clockwise';
ax.ThetaZeroLocation = 'top';
title('SkyView')
% Угол места
th = hours(t_arr/60/60 - 2*24+3); % Перевод временной оси в формат hh:mm:ss
figure (3)
grid on
hold on
plot(th,(-teta)*180/pi+90,'DurationTickFormat','hh:mm:ss')
xlim([th(1) th(end)])
title('Угол места', 'FontSize',12)
xlabel('Время в UTC', 'FontSize',12)
ylabel('Угол места, град', 'FontSize',12)

```

```

#include <iostream>
#include <fstream>
#include <string>
#include "Soordinate.h"
#include "Kepler.h"
#include <ctime>
#include <cmath>

using namespace std;

int main()
{
    Efemeridi Efemer;
    Efemer.mu = 3.986004418E+14;
    Efemer.we = 7.2921151467E-5;
    Efemer.toe = 244800;
    Efemer.A = 4.216001679802313e+07;
    Efemer.e = 7.69855396356433630e-03;
    Efemer.M0 = 2.34731910518265385e+00;
    Efemer.omega = -2.38950527529739887e+00;
    Efemer.i0 = 9.50974837701972997e-01;
    Efemer.omega0 = 6.79084253670661386e-01;
    Efemer.del_n = 7.875327993900338e-10;
    Efemer.i_dot = 6.78599694972560970e-15;
    Efemer.omega_dot = -1.75185868623179542e-12;
    Efemer.c_uc = -1.24704092741012573e-06;
    Efemer.c_us = 2.29864381253719330e-05;
    Efemer.c_rc = -4.67468750000000000e+02;
    Efemer.c_rs = -4.98437500000000000e+01;
    Efemer.c_ic = -6.98491930961608887e-08;
    Efemer.c_is = 8.84756445884704590e-08;

    time_t start, end;
    double t = 226800;
    double delta_t = 1;
    double* coord = new double[3];
    double* coord_matlab = new double[3];
    double max_del = 0;
    ofstream out;
    out.open("cpp.txt");
    ifstream in("matlab.txt");
    time(&start);
    for (int i = 0; i < (43200 / delta_t); i++)
    {
        Ocoord(t, coord, Efemer);
        t += delta_t;
        string coord_str1 = to_string(coord[0]);
        string coord_str2 = to_string(coord[1]);
        string coord_str3 = to_string(coord[2]);
        out << coord_str1 << " " << coord_str2 << " " << coord_str3 << endl;
        in >> coord_matlab[0] >> coord_matlab[1] >> coord_matlab[2];
        for (int j = 0; j < 3; j++)
        {
            if (abs(coord[j] - coord_matlab[j]) > max_del)
            {
                max_del = abs(coord[j] - coord_matlab[j]);
            }
        }
    }
    max_del = max_del * 10000000;
    time(&end);
    in.close();
    delete[] coord;
    delete[] coord_matlab;
}

```

```

coord = nullptr;
coord_matlab = nullptr;
double seconds = difftime(end, start);
string seconds1 = to_string(seconds / 43200 / delta_t);
setlocale(LC_ALL, "rus");
cout << "\n\t\tОбщее время расчёта, с: " << seconds << endl;
cout << "\n\t\tСреднее время расчёта, с: " << seconds1 << endl;
string max_del1 = to_string(max_del);
cout << "\n\t\tМаксимальная разность координат(значение домножено на 10^7): " << max_del1
<< endl;
out.close();
in.close();
}

```

Приложение 3

```

#include "Soordinate.h"
#include <iostream>
#include <math.h>
#include <ostream>
#include "Kepler.h"

using namespace std;

void Ocoord(double t, double* coord, Efemeridi Ef)
{
    double tk = t - Ef.toe;
    double Mk = Ef.M0 + (sqrt(Ef.mu) / pow(sqrt(Ef.A), 3) + Ef.del_n) * tk;
    double Ek = kepler(Mk, Ef.e);
    double Vk = atan2(sqrt(1 - pow(Ef.e, 2)) * sin(Ek), cos(Ek) - Ef.e);
    double Uk = Ef.omega + Vk + Ef.c_uc * cos(2 * (Ef.omega + Vk)) + Ef.c_us * sin(2 *
(Ef.omega + Vk));
    double rk = Ef.A * (1 - Ef.e * cos(Ek)) + Ef.c_rc * cos(2 * (Ef.omega + Vk)) + Ef.c_rs *
sin(2 * (Ef.omega + Vk));
    double ik = Ef.i0 + Ef.i_dot * tk + Ef.c_ic * cos(2 * (Ef.omega + Vk)) + Ef.c_is * sin(2 *
(Ef.omega + Vk));
    double lambk = Ef.omega0 + (Ef.omega_dot - Ef.we) * tk - Ef.we * Ef.toe;
    coord[0] = (cos(-lambk) * cos(-Uk) - sin(-lambk) * cos(-ik) * sin(-Uk)) * rk;
    coord[1] = (-sin(-lambk) * cos(-Uk) - cos(-lambk) * cos(-ik) * sin(-Uk)) * rk;
    coord[2] = (sin(-ik) * sin(-Uk)) * rk;
}

```

Приложение 4

```

#include "Kepler.h"
#include <math.h>

double kepler(double Mk, double e) {
    double Ek = Mk;
    double Ek1;
    do {
        Ek1 = Ek;
        Ek = Mk + e * sin(Ek);
    } while (fabs(Ek1 - Ek) > 0.000001);
    return Ek; }

```

Приложение 5

```

#ifdef KEPLER_H

```

```
#define KEPLER_H

double kepler(double Mk, double e);

#endif /* #ifndef KEPLER_H */#pragma once
```

Приложение 6

```
#ifndef COORDINATE_H
#define COORDINATE_H
typedef struct {
    double mu;
    double we;
    double toe;
    double A;
    double e;
    double M0;
    double omega;
    double i0;
    double omega0;
    double del_n;
    double i_dot;
    double omega_dot;
    double c_uc;
    double c_us;
    double c_rc;
    double c_rs;
    double c_ic;
    double c_is;
} Efemeridi;
void Ocoord(double t, double* coord, Efemeridi Ef);

#endif /* #ifndef GPSSVPOS_H */#pragma once
```