Министерство образования и науки Российской Федерации

Федеральное агентство по образованию
ГОУ ВПО
РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ
(РГГМУ)

Допущен к защите

Зав. кафедрой д.ф.-м. н, проф.

А.Д. Кузнецов

Кафедра экспериментальной физики атмосферы

# ДИПЛОМНЫЙ ПРОЕКТ

# Программный комплекс дистанционного зондирования атмосферы сигналами глобальной навигационной спутниковой системы

Выполнил Г. В. Григорьев,
гр. И-539
Руководитель канд. физ.-мат. наук,
доцент
В.В. Чукин

Санкт-Петербург 2009

Содержание


| | | Стр. |
|---|---|---|
| | Сокращения | 4 |
| | Введение | 6 |
| 1 | Глобальная навигационная спутниковая система | 8 |
| 1.1 | Исторические сведения | 8 |
| 1.2 | Основные элементы спутниковой системы навигации | 9 |
| 1.3 | Принцип работы спутниковых систем навигации | 9 |
| 1.4 | Технические детали работы систем GPS и ГЛОНАСС | 11 |
| 1.5 | Дифференциальный режим определения координат | 14 |
| 1.6 | Виды погрешностей определения дальности до спутника | 17 |
| 1.7 | Факторы, искажающие результаты навигационных измерений | 19 |
| 1.8 | Влияние среды на распространение навигационных сигналов | 20 |
| 1.8.1 | Распространение радиоволн в ионосфере | 22 |
| 1.8.2 | Распространение радиоволн в тропосфере | 23 |
| 1.9 | Методы учета факторов, влияющих на точность измерений дальности | 25 |
| 1.9.1 | Определение ионосферных поправок | 25 |
| 1.9.2 | Задержка радиосигнала в тропосфере | 27 |
| 2 | Методы вычисления местоположения спутников в пространстве | 35 |
| 2.1 | Параметры движения спутников по орбите | 35 |
| 2.2 | Система координат WGS-84 | 39 |
| 2.3 | Система координат ПЗ-90 | 41 |
| 2.4 | Система координат СК-95 | 43 |
| 2.5 | Переход от одной системы координат к другой | 44 |
| 2.5.1 | Преобразования из геоцентрических в геоцентрические координаты | 45 |
| 2.5.2 | Преобразования из географических в географические | 48 |

2 координаты
2.5. Преобразование между системами координат WGS-84 и ПЗ-90 49
3
2.6 Методы вычисления положения спутника в пространстве 50
3 Форматы обмена и хранения навигационных данных 54
3.1 Формат обмена навигационными данными RINEX 54
3.2 Формат SP3 57
3.3 Основные сведения о системах управления базами данных 59
3.4 Реляционные базы данных 63
3.5 Структуризованный язык запросов SQL 64
3.6 Проектирование базы данных для хранения информации о параметрах движения спутников по орбите 68
4 Разработка программного комплекса дистанционного зондирования атмосферы сигналами глобальной навигационной спутниковой системы «GPS Tropodelay» 70
4.1 Выбор операционной и программной среды, определение аппаратных требований 70
4.2 Алгоритм вычисления координат НКА GPS на моменты измерения дальности 71
4.3 Организация компонентов программы «GPS Trodelay» 78
4.4 Описание работы 85
Заключение 91
Список использованных источников 93
Приложение А 96
Приложение Б 97
Приложение В 114

## Сокращения

| | | |
|---|---|---|
| DGPS | – | дифференциальный режим глобальной системы позиционирования; |
| EGNOS | – | европейская система передачи поправок потребителям; |
| GPS | – | глобальная система позиционирования (США); |
| ISO | – | международный стандарт; |
| ITRF | – | система координат WGS-84, скорректированная на 1996г; |
| Linux | – | UNIX-подобные операционные системы с открытым кодом; |
| MSAS | – | японская система передачи поправок потребителям; |
| MySQL | – | свободная система управления базами данных; |
| NTRIP | – | протокол передачи дифференциальных поправок через Интернет; |
| RINEX | – | формат обмена навигационными данными; |
| SBAS | – | файл широкозонной корректирующей информации; |
| SP3 | – | формат обмена навигационными данными; |
| SQL | – | структуризованный язык запросов; |
| WAAS | – | американская система передачи поправок потребителям; |
| WGS-84 | – | всемирная система координат (1984г); |
| БД | – | базы данных; |
| ГЛОНАСС | – | навигационная спутниковая система (Россия); |
| ГНСС | – | глобальная навигационная спутниковая система; |
| ИСЗ | – | искусственный спутник Земли; |
| КГК | – | космический геодезический комплекс; |
| КГС | – | космическая геодезическая сеть; |
| МАГ | – | международная ассоциация геодезии; |
| НКА | – | навигационный космический аппарат; |
| ОЗЭ | – | общий земной эллипсоид; |
| ПЗ-90 | – | государственная геоцентрическая система координат (1990г); |
| РНС | – | радионавигационный сигнал; |
| СРНС | – | спутниковые радионавигационные системы; |
| СУБД | – | система управления базами данных; |

ЯМД – язык манипулирования данными.

# Введение

В последние десятилетия интенсивно развиваются космические радионавигационные методы для дистанционного зондирования атмосферы. С развитием глобальной навигационной спутниковой системы (ГНСС) появилась перспектива использования информации, получаемой на сети наземных базовых станций дифференциальных подсистем спутниковых навигационных систем ГЛОНАСС и GPS для задач дистанционного зондирования атмосферы.

В 2004г. в РГГМУ [1] предложен новый метод электромагнитного просвечивания, в котором изложены принципы использования сигналов космических источников для целей дистанционного зондирования атмосферы.

В 2008 году на кафедре ЭФА в рамках научной работы (Грант Правительства Санкт-Петербурга №327/08 от 05.11.2008) проведено исследование возможностей использования информации, получаемой на сети наземных базовых станций дифференциальных подсистем спутниковых навигационных систем ГЛОНАСС и GPS для осуществления дистанционного зондирования атмосферы методом электромагнитного просвечивания. Результаты проекта показали, что данные ГНСС могут использоваться для сбора информации *о интегральном содержании водяного пара*, что может быть применено для целей анализа и прогноза погоды. Также эти данные могут повышать точность определения координат пользователей дифференциальной подсистемы ГНСС за счет более точного определения тропосферных задержек навигационных радиосигналов.

Для обработки данных ГНСС необходимо разработать автоматизированную информационную систему, которая осуществляет сбор и преобразование информации. Один из этапов работы по созданию системы сбора информации – получение файлов ГНСС и расчет тропосферной задержки.

Основной *целью* работы является создание автоматизированного программного комплекса для расчета тропосферной задержки в атмосфере навигационного сигнала глобальной навигационной спутниковой системы Navstar GPS.

Для достижения поставленной цели необходимо решить следующие *задачи*:

1) ознакомится с глобальной навигационной спутниковой системой;
2) изучить методы определения положения спутника в пространстве;
3) разработать структуру и создать базу данных для хранения информации о параметрах движения спутников системы Navstar GPS по орбите;
4) изучить структуру навигационной информации формата RINEX;
5) разработать программный комплекс, предназначенный для расчета тропосферной задержки в атмосфере навигационного сигнала спутников глобальной навигационной спутниковой системы.

В первой главе рассматриваются основные элементы глобальной навигационной спутниковой системы, а также принципы работы спутниковых систем навигации, рассмотрены факторы, искажающие результаты измерений и приведены методы учета этих факторов.

Во второй главе проводятся методы вычисления местоположения спутников в пространстве, рассмотрены основные системы координат и методы перехода от одной системы координат к другой.

Третья глава посвящена форматам обмена и хранения навигационных сообщений, даны сведения о системах управления базами данных и языке структурированных запросов SQL. Представлен проект базы данных для хранения информации о параметрах движения спутников по орбите.

В четвертой главе представлены результаты разработки программного комплекса дистанционного зондирования атмосферы сигналами глобальной навигационной спутниковой системы «GPS Tropodelay».

# 1 Глобальная навигационная спутниковая система

## 1.1 Исторические сведения

Развитие спутниковых навигационных систем имеет практически пятидесятилетнюю историю, начало которой положено запуском 4.10.57 г. первого в истории человечества искусственного спутника Земли (ИСЗ). Наблюдения за спутником показали, что частота принимаемого сигнала увеличивается при приближении спутника и уменьшается при его отдалении [2]. Суть открытия заключалась в том, что, зная свои координаты на Земле, становится возможным измерить положение и скорость спутника, и наоборот, точно зная положение спутника, можно определить собственную скорость и координаты.

В США система получила название NAVSTAR GPS (NAVigation Satellites providing Time And Range; Global Positioning System). Первый тестовый спутник выведен на орбиту в 1974 г, а последний из всех 24 спутников, необходимых для полного покрытия земной поверхности, был выведен на орбиту в 1993 г. Система разработана, реализована и эксплуатируется Министерством обороны США.

В Советском Союзе система названа ГЛОНАСС (ГЛОбальная НАвигационная Спутниковая Система) Первый спутник был выведен на орбиту в 1982 году. В 1995 году спутниковая группировка составила 24 аппарата. Но впоследствии из-за недостаточного финансирования, а также из-за малого срока службы, число работающих спутников сократилось.

В конце марта 2008 года в России были скорректированы сроки развёртывания космического сегмента ГЛОНАСС. Орбитальная группировка по состоянию на 25 декабря 2008 года количество составляла 18 спутников.

## 1.2 Основные элементы спутниковой системы навигации

Спутниковая система навигации – комплексная электронно-техническая система, состоящая из совокупности наземного и космического оборудования, предназначенная для определения местоположения (географических координат и высоты), а также параметров движения (скорости и направления движения и т. д.) для наземных, водных и воздушных объектов.

К основным элементам спутниковой системы навигации относятся[2, 3]:

1) орбитальная группировка, состоящая из нескольких (от 2 до 30) космических аппаратов, состоящая из навигационных ИСЗ, излучающих специальные радиосигналы, называемая космическим сегментом;
2) наземная система управления и контроля, включающая блоки измерения текущего положения спутников и передачи на них полученной информации для корректировки информации об орбитах или сегмент управления (контрольный сегмент);
3) приёмное клиентское оборудование, используемое для определения координат (потребительский сегмент);
4) опционально: информационная радиосистема для передачи пользователям поправок, позволяющих значительно повысить точность определения координат.

Основные элементы ГНСС изображены на рисунке 1.1

## 1.3 Принцип работы спутниковых систем навигации

Принцип работы спутниковых систем навигации (см. рис.1.2) основан на измерении расстояния от антенны на объекте (координаты которого необходимо получить) до спутников, положение которых известно с большой точностью.

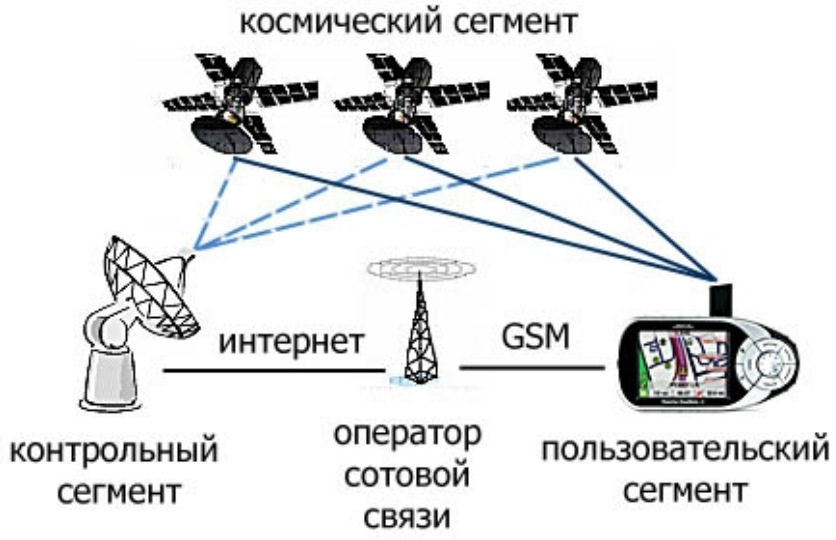



Рисунок 1.1 – Основные элементы ГНСС

Таблица положений всех спутников называется альманахом, которым должен располагать любой спутниковый приёмник до начала измерений. Обычно приёмник сохраняет альманах в памяти со времени последнего выключения и если он не устарел – мгновенно использует его. Каждый спутник передаёт в своём сигнале весь альманах. Таким образом, зная расстояния до нескольких спутников системы, с помощью обычных геометрических построений, на основе альманаха, можно вычислить положение объекта в пространстве.

Метод измерения расстояния от спутника до антенны приёмника основан на определённости скорости распространения радиоволн. Для осуществления возможности измерения времени распространения радиосигнала каждый спутник навигационной системы излучает сигналы точного времени, в составе своего сигнала используя точно синхронизированные с системным временем атомные часы. При работе

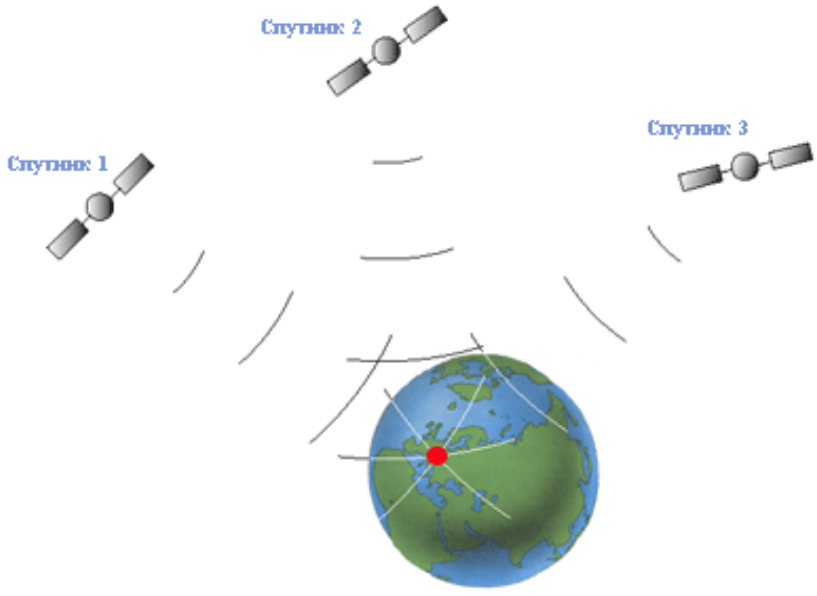



Рисунок 1. 2 – Принцип работы спутниковых систем навигации

спутникового приёмника его часы синхронизируются с системным временем и при дальнейшем приёме сигналов вычисляется задержка между временем излучения, содержащимся в самом сигнале, и временем приёма сигнала. Располагая этой информацией, навигационный приёмник вычисляет координаты антенны. Дополнительно накапливая и обрабатывая эти данные за определённый промежуток времени, становится возможным вычислить такие параметры движения, как скорость (текущую, максимальную, среднюю), пройденный путь и т. д.

## 1.4 Технические детали работы систем GPS и ГЛОНАСС

В настоящее время работают или готовятся к развёртыванию системы спутниковой навигации, представленные в табл. 1.1.

К основным системам спутниковой навигации, как наиболее развернутым, относятся системы GPS и ГЛОНАСС. Обе системы имеют двойное назначение – военное и гражданское, поэтому излучают два вида

Таблица 1.1 – Системы спутниковой навигации

| Название | Страна | Этап | Использование за пределами страны |
|---|---|---|---|
| GPS | США | Эксплуатация | да |
| ГЛОНАСС | Россия | Повторное развёртывание спутниковой группировки | да |
| Бэйдоу | Китай | Разработка | нет |
| Galileo | Европейская система | Создание спутниковой группировки | да |
| IRNSS | Индия | Разработка | нет |

сигналов: один с пониженной точностью определения координат (~100 м) для гражданского применения и другой высокой точности (~10-15 м и точнее) для военного применения. Для ограничения доступа к точной навигационной информации вводят специальные помехи, которые могут быть учтены после получения ключей от соответствующего военного ведомства (США для GPS и России для ГЛОНАСС). В настоящее время эти помехи отменены, и точный сигнал доступен гражданским приёмникам, однако в случае соответствующего решения государственных органов стран-владельцев военный код может быть снова заблокирован (в системе NAVSTAR это ограничение было отменено только в мае 2000 года и в любой момент может быть восстановлено).

Спутники GPS располагаются в шести плоскостях на высоте примерно 20 180 км. Спутники ГЛОНАСС (шифр «Ураган») находятся в трёх плоскостях на высоте примерно 19 100 км (см. рис. 1.3). Номинальное количество спутников в обеих системах – 24 [2, 3]. Группировка GPS полностью укомплектована в апреле 1994-го и с тех пор поддерживается, группировка ГЛОНАСС была полностью развёрнута в декабре 1995-го, но с тех пор значительно деградировала. В настоящий момент идёт её активное восстановление.

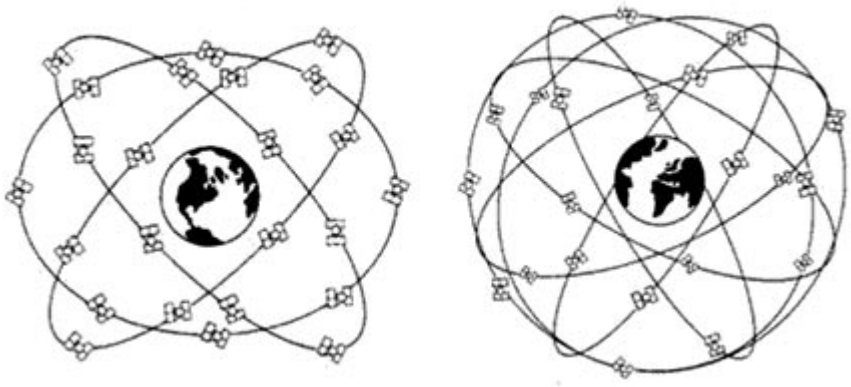


Рисунок 1.3 – Космический сегмент спутников ГЛОНАСС и GPS

Обе системы используют сигналы на основе т. н. «псевдошумовых последовательностей», применение которых придаёт им высокую помехозащищённость и надёжность при невысокой мощности излучения передатчиков.

В соответствии с назначением, в каждой системе есть две базовые частоты — L1 (стандартной точности) и L2 (высокой точности). Для GPS L1=1575.42 МГц и L2=1227.6 МГц. В ГЛОНАСС используется частотное разделение сигналов, т. е. каждый спутник работает на своей частоте и, соответственно, L1 находится в пределах от 1602.56 до 1615.5 МГц и L2 от 1246.43 до 1256.53. Сигнал в L1 доступен всем пользователям, сигнал в L2 – только военным (то есть, не может быть расшифрован без специального секретного ключа).

Каждый спутник системы, помимо основной информации, передаёт также вспомогательную, необходимую для непрерывной работы приёмного оборудования. В эту категорию входит полный альманах всей спутниковой группировки, передаваемый последовательно в течение нескольких минут. Таким образом, старт приёмного устройства может быть достаточно быстрым, если он содержит актуальный альманах (порядка 1-й минуты) — это называется «тёплый старт», но может занять и до 15-ти минут, если приёмник

вынужден получать полный альманах – т. н. «холодный старт». Необходимость в «холодном старте» возникает обычно при первом включении приёмника, либо если он долго не использовался.

Основные параметры систем спутниковой навигации (GPS и ГЛОНАСС) приведены в таблице 1.2 [4].

## 1.5 Дифференциальный режим определения координат

Спутниковые навигационные системы позволяют потребителю получить координаты с точностью порядка 10-15 м [4]. Однако для многих задач, особенно для навигации в городах, требуется большая точность. Один из основных методов повышения точности определения местонахождения объекта основан на применении известного в радионавигации принципа дифференциальных навигационных измерений.

Дифференциальный режим DGPS (Differential GPS) позволяет установить координаты с точностью до 3 м в динамической навигационной обстановке и до 1 м – в стационарных условиях. Дифференциальный режим реализуется с помощью контрольного GPS-приёмника, называемого опорной станцией. Она располагается в пункте с известными координатами, в том же районе, что и основной GPS-приёмник. Сравнивая известные координаты (полученные в результате прецизионной геодезической съёмки) с измеренными, опорная станция вычисляет поправки, которые передаются потребителям по радиоканалу в заранее оговоренном формате. Аппаратура потребителя принимает от опорной станции дифференциальные поправки и учитывает их при определении местонахождения потребителя.

Результаты, полученные с помощью дифференциального метода, в значительной степени зависят от расстояния между объектом и опорной

Таблица 1.2 – Сравнительные характеристики систем ГЛОНАСС и GPS

станцией. Применение этого метода наиболее эффективно, когда

| Показатель | ГЛОНАСС | GPS |
|---|---|---|
| Число КА в полной орбитальной группировке | 24 | 24 |
| Число орбитальных плоскостей | 3 | 6 |
| Число КА в каждой плоскости | 8 | 4 |
| Наклонение орбиты, град | 64.8º | 55.0º |
| Высота орбиты, км | 19 130 | 20 180 |
| Период обращения спутника | 11 ч. 15 мин. 44 с | 11 ч. 58 мин. 00 с |
| Система координат | ПЗ-90 | WGS-84 |
| Масса навигационного КА, кг | 1450 | 1055 |
| Мощность солнечных батарей, Вт | 1250 | 450 |
| Срок активного существования, лет | 3.0 | 7.5 |
| Число КА, выводимых за один запуск | 3 | 1 |
| Космодром | Байконур (Казахстан) | Мыс Канаверел (Cape Canaveral) |
| Эталонное время | UTC (SU) | UTC (NO) |
| Несущая частота, МГц :<br>L1<br>L2 | <br>1598.0625—1604.2500<br>7/9 L1 | <br>1575.4200<br>60/77 L1 |
| Поляризация | Правосторонняя | Правосторонняя |
| Тип псевдошумовой последовательности | m-последовательность | код Голда |
| Число элементов кода:<br>C/A<br>P | <br>511<br>51 1000 | <br>1023<br>2.35· 1014 |
| Скорость кодирования, Мбит/с:<br>C/A<br>P | <br>0.511<br>5.11 | <br>1.023<br>10.23 |
| Уровень внутрисистемных радиопомех, дБ | -48,0 | -21,6 |
| Структура навигационного сообщения | | |
| Скорость передачи, бит/с | 50 | 50 |
| Длина суперкадра, мин. | 2.5 (5 кадров) | 12.5 (25 кадров) |
| Длина кадра, с | 30 (15 строк) | 30 (5 строк) |
| Длина строки, с | 2 | 6 |

преобладающими являются систематические ошибки, обусловленные внешними (по отношению к приёмнику) причинами. По экспериментальным данным, опорную станцию рекомендуется располагать не далее 500 км от объекта.

В настоящее время существуют множество широкозонных, региональных и локальных дифференциальных систем.

Дифференциальная поправка основана либо на геостационарных спутниках, либо на наземных базовых станциях, может быть платной (расшифровка сигнала возможна только одним определённым приёмником после оплаты «подписки на услугу») или бесплатной. В настоящее время (2009 г.) существуют бесплатные американская система WAAS, европейская система EGNOS, японская система MSAS основанные на нескольких передающих коррекции геостационарных спутниках, позволяющих получить высокую точность (до 30 см). Эти системы используют геостационарные спутники для передачи поправок всем потребителям, находящимся в зоне их покрытия.

Региональные системы предназначены для навигационного обеспечения отдельных участков земной поверхности. Обычно региональные системы используют в крупных городах, на транспортных магистралях и судоходных реках, в портах и по берегу морей и океанов. Диаметр рабочей зоны региональной системы обычно составляет от 500 до 2000 км. Она может иметь в своём составе одну или несколько опорных станций [4].

Локальные системы имеют максимальный радиус действия от 50 до 220 км. Они включают обычно одну базовую станцию. Локальные системы обычно разделяют по способу их применения: морские, авиационные и геодезические локальные дифференциальные станции.

## 1.6 Виды погрешностей определения дальности до спутника

При оценке точности навигационных измерений различают погрешности измерения навигационного параметра радионавигационного сигнала (РНС) и погрешности определения соответствующих навигационных элементов, среди которых важнейшим является местоположение объекта.

Погрешности РНС зависят, прежде всего, от метода, положенного в основу построения радионавигационного средства, от особенностей технической реализации РНС, от условий их эксплуатации. В зависимости от причин, порождающих погрешности, их можно разделить на три группы [5].

К *первой* группе относят методические погрешности. Они обусловлены неточностью исходных математических моделей, описывающих сигналы и измеряемые процессы, и несовершенством метода измерений и алгоритмов, являющихся основой построения РНС. Например, в ряде случаев при разработке алгоритмов обработки принимаемых радионавигационных сигналов не учитываются динамика подвижного объекта в процессе измерения, изменение условий распространения радиоволн и т. д.

Во *вторую* группу включают инструментальные погрешности, вызванные несовершенством радионавигационной аппаратуры, и, в частности, тем, что принятые алгоритмы не могут быть точно реализованы. Причинами инструментальных погрешностей являются:

- схемные и конструктивные недостатки устройств;
- недостаточная чувствительность индикаторов;
- погрешности регулировок и калибровок, а также градуировок шкал приборов;
- зависимость показаний от непостоянства напряжений источников питания и т. п.

*Третью* группу составляют погрешности, вызванные условиями, в которых проводятся измерения. Источниками этих погрешностей являются: внешние помехи, условия распространения радиоволн, турбулентность атмосферы и т. д.

По характеру проявления погрешности РНС делятся на систематические и случайные.

*Систематические* погрешности являются постоянными или меняющимися по определенному закону. Они вызываются причинами,

которые действуют вполне определенным образом. Такие погрешности независимо от числа измерений имеют одно и то же значение и один и тот же знак. Их влияние в ряде случаев может быть установлено и заранее учтено. Примерами являются погрешности градуировки; погрешности за счет изменений условий распространения радиоволн.

*Случайные* погрешности вызываются большим числом причин, действующих при каждом отдельном измерении различным образом. Такие погрешности при измерениях полностью устранить не удается, но, используя соответствующие методы приема и обработки радиосигналов, их можно существенно снизить.

Количество причин, влияющих на погрешности измерения навигационных параметров, велико и степень воздействия их обычно примерно одинакова, поэтому во многих случаях считают, что погрешности радионавигационных измерений в соответствии с центральной предельной теоремой имеют гауссовский закон распределения. При этом основной мерой точности является среднеквадратическое значение $\sigma$ погрешности измерения навигационного параметра. Вероятность того, что погрешности измерений не превысят величины $\sigma$, равна 0.683. На практике точность навигационных средств часто характеризуют максимальной погрешностью, равной $2\sigma$, а в ряде случаев, например, применительно к системам посадки, - предельной погрешностью, которая равна $\sigma$. При гауссовском законе распределения погрешностей 95% всех измерений имеют погрешности меньше максимальной и 99.7% - меньше предельной [5].

## 1.7 Факторы, искажающие результаты навигационных измерений

Источники погрешностей, воздействующих на результаты измерений псевдодальностей, могут быть условно разделены на три группы, связанные,

соответственно, со спутником, приемником, а также со средой распространения сигнала.

Как бы точны ни были атомные часы на спутниках, все же и у них имеются источники небольших погрешностей. Специальные станции следят за этими часами и могут выверить их, если выявиться хотя бы незначительный уход.

Наши приемники на Земле также иногда ошибаются. Компьютер приемника может округлить математическую операцию, или электрические помехи могут привести к ошибочной обработке навигационных сигналов.

Когда сигналы, передаваемые со спутника, многократно переотражаются от окружающих предметов и поверхностей до того, как попадают в приемник, возникает эффект многолучевости [5].

Релятивистские эффекты. Их влияние сказывается на орбитах спутников, на распространении сигналов и на ходе часов спутника.

При прохождении через атмосферу скорость распространения радиоволн отличается от скорости света в вакууме, поэтому в вычисленные значения псевдодальностей необходимо ввести поправку. Принято значение этой поправки разбивать на три части, одна из которых относится к свободному пространству между спутником и ионосферой, вторая – к ионосфере, содержащей электрически заряженные частицы и третья – к нижней части атмосферы - тропосфере.

Все источники погрешностей, приведенные выше, суммируются и придают каждому измерению навигационных параметров некоторую неопределенность.

## 1.8 Влияние среды на распространение навигационных сигналов

Радиосигнал, распространяющийся от НКА до потребителя, проходит три существенно различные среды, влияющие на поглощение энергии навигационного сигнала, его скорость и траекторию.

Расчет амплитудных и траекторных характеристик радиоволны, распространяющейся в какой-либо среде, производится на основе решения уравнений Максвелла, которые в дифференциальном виде записываются следующим образом [5]:

$$rotE = \frac{\partial B}{\partial t}, \quad (1.1)$$

$$rotH = j + \frac{\partial D}{\partial t}, \quad (1.2)$$

$$divE = \rho, \quad (1.3)$$

$$divB = 0. \quad (1.4)$$

Уравнения (1.1)-(1.4) называются уравнениями поля.

$$D=\varepsilon_a \cdot E, \quad (1.5)$$

$$B=\mu \cdot H, \quad (1.6)$$

$$j=\sigma(E+E^{ст}), \quad (1.7)$$

где $\mu$ – магнитная проницаемость, Гн/м;
$E$ – напряженность электрического поля, В/м;
$j$ – плотность электрического тока, А/м$^2$;
$\sigma$ – электропроводность среды, 1/(Ом· м);
$\varepsilon$ – диэлектрическая проницаемость среды, Ф/м;
$\mu$ – магнитная проницаемость среды;
$H$ – Напряженность магнитного поля, А/м;
$B$ – магнитная индукция, Тл;
$D$ – электрическая индукция, Кл/м$^2$;
$\rho$ – плотность стороннего электрического заряда, Кл/м$^3$;
*ro* – дифференциальный оператор ротора;
*t*
*di* – дифференциальный оператор дивергенции.
*v*

Уравнения (1.5)-(1.7) называются уравнениями состояния и характеризуют свойства среды, в которой происходят электромагнитные

процессы. Характеристиками среды являются величины: $\varepsilon_a=\varepsilon\cdot\varepsilon_0$, $\mu\cdot a=\mu\cdot\mu_0$ и $\sigma$, Эти величины с нулевым индексом относятся к вакууму или так называемому свободному пространству и имеют значения:

$$\varepsilon_0=1/36\,\pi\cdot 109\ \text{Ф/м} \quad \text{и}$$

$$\mu_0=4\pi\cdot 10\text{-}7\ \text{Гн/м}.$$

В общем случае $\varepsilon$, $\mu$ и $\sigma$ могут быть функциями времени и координат, это относится к величинам $\varepsilon$ и $\sigma$ для реальной земной поверхности и атмосферы, свойства которых неоднородны и непостоянны во времени.

Также следует отметить, что среда влияет на поглощение, траекторию (отражение, дифракция, преломление) и скорость распространения радиоволн. Магнитное поле Земли в ионизированной плазме приводит к расщеплению радиосигнала на обыкновенную и необыкновенную составляющие, которые имеют специфические дисперсионные характеристики взаимодействия со средой.

Даже от одного излучателя в пункт приема приходит несколько волн, распространяющихся различными траекториями (модами). Векторное сложение этих волн приводит к таким эффектам как интерференция, вращение плоскости поляризации, сопровождаемые нежелательными федингами амплитуды суммарного сигнала.

1.8.1 Распространение радиоволн в ионосфере

Ионосфера является частью атмосферы и имеет сложный высотный профиль температуры, электронной концентрации и распределения газовых составляющих.

Распространение радиоволн в ионосферной плазме, находящейся в магнитном поле Земли, имеет сложный характер. Наиболее важными параметрами ионосферы, влияющими на распространение электромагнитных

волн, являются электронная концентрация и частота соударений электронов с нейтральными частицами атмосферного газа.

Одним из наиболее часто используемых приближений для расчетов распространения радиоволн в ионосфере является условие, при котором вектор магнитного поля лежит в плоскости распространения [5, 6].

В ионосфере показатель преломления электромагнитных волн определяется влиянием свободных электронов [1] и его можно выразить через электронную концентрацию:

$$n_u = \sqrt{1 - \frac{80.8 \cdot N_e}{f^2}}, \tag{1.8}$$

где $n_u$ – показатель преломления электромагнитных волн в ионосфере;
$f$ – частота, Гц.

Ионизированная среда влияет на распространение электромагнитных волн различной частоты не одинаковым образом. Существует критическая частота, ниже которой плазма не пропускает электромагнитные волны. Такой частотой является плазменная частота. При распространении электромагнитных волн с частотой много выше плазменной ($f >> f_P$) показатель преломления мало отличается от единицы и условия распространения волн в плазме мало отличаются от условий распространения в вакууме.

Электромагнитные волны с частотой много меньше плазменной частоты ($f << f_P$) отражаются плазмой и, следовательно, такие волны не будут распространяться в плазме. Электромагнитные волны с частотой равной плазменной частоте ($f = f_P$) поглощаются плазмой в результате явления резонанса. В этом случае показатель преломления равен нулю. Таким образом, распределение показателя преломления в ионосфере $n(z)$ определяется полностью распределением электронной концентрации $Ne(z)$.

1.8.2 Распространение радиоволн в тропосфере

Тропосфера искривляет первоначальную траекторию радиолуча от НКА и уменьшает скорость его распространения. Дополнительная задержка сигнала НКА в тропосфере может достигать 8 – 80 нс (экспериментальные данные для НКА GPS)[3].

Показатель преломления радиоволн в нейтральной атмосфере может быть представлен в виде:

$$n = 1 + \frac{k_1 \cdot (P - e)}{T} + \frac{k_2 \cdot e}{T} + \frac{k_3 \cdot e}{T^2}, \tag{1.9}$$

где $k_1, k_2, k_3$ – эмпирические коэффициенты (см. таблицу 1.2);
$P$ – атмосферное давление, Па;
$T$ – температура воздуха, К;
$e$ – парциальное давление водяного пара, Па.

Таблица 1.3 – Значения коэффициентов $\mu k_1, \mu k_2, k_3$ формулы для расчета показателя преломления радиоволн [1]

| № | $\mu k_1$, К/Па | $\mu k_2$, К/Па | $k_3$, К$^2$/Па | Год | Источ-ник |
|---|---|---|---|---|---|
| 1 | $(7.761 \pm 0.001) \cdot 10^{-7}$ | $(7.20 \pm 0.90) \cdot 10^{-7}$ | $(3.750 \pm 0.030) \cdot 10^{-3}$ | 1953 | [7] |
| 2 | $(7.759 \pm 0.008) \cdot 10^{-7}$ | $(7.20 \pm 1.10) \cdot 10^{-7}$ | $(3.750 \pm 0.030) \cdot 10^{-3}$ | 1963 | [8] |
| 3 | $(7.760 \pm 0.001) \cdot 10^{-7}$ | $(6.479 \pm 0.008) \cdot 10^{-7}$ | $(3.776 \pm 0.004) \cdot 10^{-3}$ | 1974 | [9] |

Уравнение для расчета показателя преломления можно представить как зависимость от плотности воздуха $(\rho = \rho_C + \rho_П)$ и плотности водяного пара:

$$n = 1 + k_1 \cdot R_C \cdot \rho_C + k_2 \cdot R_П \cdot \rho_П + \frac{k_3 \cdot R_П \cdot \rho_П}{T} = \tag{1.10}$$

$$= 1 + k_1 \cdot R_C \cdot \rho + \left( k_2 - k_1 \frac{R_C}{R_П} \right) \cdot R_П \cdot \rho_П + \frac{k_3 \cdot R_П \cdot \rho_П}{T} =$$

$$= 1 + k_1 \cdot R_C \cdot \rho + \left( k_2' \cdot R_П + \frac{k_3 \cdot R_П}{T} \right) \cdot \rho_П = 1 + N_C + N_П ,$$

где $R_C = 287.0538$ Дж/(кг · К);

$R_П = 461.526$ Дж/(кг · К);

$k_2' = k_2 - k_1 \dfrac{R_C}{R_П}$.

Общий вид задержек в атмосфере представлен на рисунке 1.4

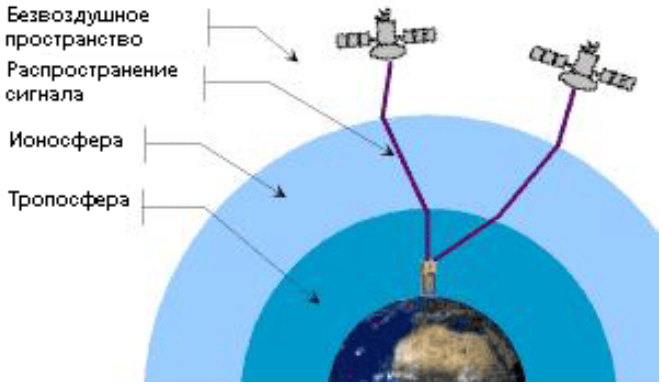



Рисунок 1.4 – Общий вид атмосферных задержек радиосигнала

1.9 Методы определения факторов, влияющих на точность измерений дальности

1.9.1 Определение ионосферных поправок

Поправка к псевдодальности, учитывающая влияние ионосферы, достаточно точно выражается эмпирической формулой

$$\Delta L_{ion} = \pm \frac{k}{f_2}, \qquad (1.11)$$

где $\Delta L_{ion}$ – ионосферная поправка, м;

$k$ – коэффициент, зависящий от концентрации электронов вдоль трассы распространения сигнала;

$f_2$ – частота, Гц.

Знак в этой формуле зависит от того, какой сигнал используется для вычисления псевдодальности: фаза несущей частоты или кодовая последовательность, содержащая группу волн в некоторой полосе частот. Первому случаю соответствует фазовая скорость распространения, второму - групповая скорость, меньшая по величине. Поправки $\Delta L_{ion}$ оказываются, согласно теории, одинаковыми по абсолютной величине в обоих случаях, но при фазовых измерениях их следует вычесть, а при кодовых - прибавить к значению расстояния, рассчитанного с использованием значения скорости света в вакууме.

Концентрация электронов на пути следования радиоволн в ионосфере сильно зависит от солнечной активности, от времени года и времени суток, от географического положения станции. Значение поправки $\Delta L_{ion}$ в направлении зенита может колебаться от 1 до 20 м, а при низких углах наклона радиолуча - достигать 100 м [5]. Эти значения необходимо учитывать, чтобы точно проводить спутниковые радионавигационные измерения. При использовании двухчастотных приемников, способных измерять псевдодальности по кодовым сигналам или по фазе несущей на двух частотах $f_1$, $f_2$, можно так скомбинировать измерения, что влияние ионосферы исключится. Пусть $\Delta L'_1$ и $\Delta L'_2$ - результаты кодовых измерений псевдодальности $L'$ на двух частотах:

$$\Delta L'_1 = L - c \cdot (\Delta t_R - \Delta t_S) + \frac{k}{f_1^2} + \Delta L_{TP}, \qquad (1.12)$$

$$\Delta L'_2 = L - c \cdot (\Delta t_R - \Delta t_S) + \frac{k}{f_2^2} + \Delta L_{TP}, \qquad (1.13)$$

Если умножить первое из этих выражений на $\frac{f_1^2}{(f_1^2 - f_2^2)}$, а второе на -

$\frac{f_2^2}{(f_1^2 - f_2^2)}$ и затем сложить, то получим:

$$\Delta L'_3 = L - c \cdot (\Delta t_R - \Delta t_S) + \Delta L_{TP} . \qquad (1.14)$$

Комбинированный результат измерения псевдодальности $P_3$ не зависит от влияния ионосферы. Точно так же можно образовать комбинированный «ионосферно-независимый» результат фазовых измерений $L_3$. Для учета ионосферной поправки ведется передача сигналов на двух частотах.

1.9.2 Задержка радиосигнала в тропосфере

Дополнительная задержка радиосигнала, связанная с прохождением через тропосферный слой, может быть определена по формуле (см. рисунок 1.5):

$$\Delta L_{TP} = \int_0^L (n-1)\,dl , \qquad (1.15)$$

где: $\Delta L_{TP}$ – пространственная задержка сигнала в тропосфере, м;
$L$ – расстояние до спутника, м;
$n$ – показатель преломления электромагнитных волн;
$l$ – путь вдоль траектории радиолуча, м.

С учетом формулы для показателя преломления получается:

$$\Delta L_{TP} = \int_0^L N_C\,dl + \int_0^L N_П\,dl . \qquad (1.16)$$

Таким образом, из формулы (1.16) следует, что задержку сигнала в тропосфере можно представить в виде суммы гидростатической задержки,

связанной с прохождением радиосигнала сквозь тропосферу, где давление с высотой убывает в соответствии с гидростатическим законом, и задержки за счет распространения в водяном паре:

$$\Delta L_{TP} = \Delta L_C + \Delta L_{\ddot{I}} \ , \qquad (1.17)$$

где $\Delta L_C$ – гидростатическая часть пространственной задержки сигнала, м;

$\Delta L_П$ – пространственная задержка сигнала, обусловленная распространением радиоволн в водяном паре, м.;

«Сухая» часть задержки составляет около 90% от полной, и она достаточно точно определяется по метеоданным, измеренным вблизи приемника. «Влажная» часть зависит от давления водяных паров, которое очень неустойчиво во времени и пространстве.

В среднем тропосферная поправка в зените составляет 2.3 м, а при угле наклона над горизонтом 15° - около 10 м.

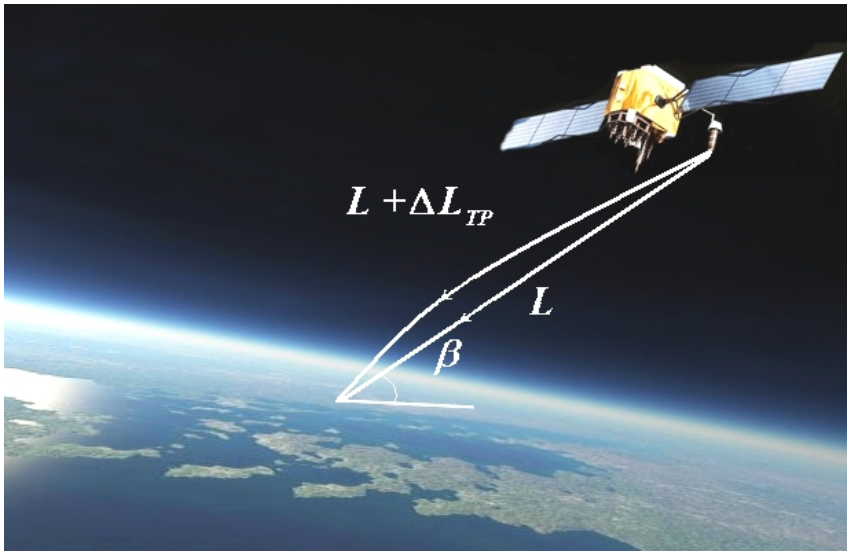


Рисунок 1.5 – Дополнительная задержка радиосигнала при распространении в тропосфере

*Гидростатическая задержка радиосигнала в тропосфере*

Гидростатическая задержка сигнала может быть представлена следующим образом:

$$\Delta L_C(\beta) = k_1 \cdot R_C \cdot \int_0^L \rho dl = \Delta L_C(90^\circ) \cdot m_C(\beta). \qquad (1.18)$$

где: $\Delta L_C(90^\circ)$ – - гидростатическая задержка радиосигнала в тропосфере при вертикальном распространении;

$m_C(\beta)$ – - отображающая функция, зависящая от угла места навигационного спутника;

$\beta$ – - угол места спутника.

Величина тропосферной задержки зависит от угла наклона направления на спутник. Чем сильнее наклон, тем длиннее путь через тропосферный слой, а вертикальный луч «прошивает» тропосферу по самому короткому пути. Закономерность формулы (1.18): когда спутник находится в зените поправка минимальна по величине.

В случае вертикального распространения сигнала от спутника до приемника, гидростатическая часть задержки равна:

$$\Delta L_C(90^\circ) = k_1 \cdot R_C \cdot \int_0^\infty \rho dz. \qquad (1.19)$$

Поскольку приземное атмосферное давление определяется как [9]:

$$P_0 = \int_0^{\infty} \rho \cdot g dz = g_m \int_0^{\infty} \rho dz, \qquad (1.20)$$

где $g_m = 9.784 \cdot \left[1 - 0.00266 \cdot \cos(2\varphi) - 0.00028 \cdot 10^{-3} \cdot h\right]$.

Тогда, с учетом $\int_0^{\infty} \rho dz = \frac{P_0}{g_m}$, получается расчетная формула для определения гидростатической части задержки сигнала при вертикальном распространении [10]:

$$\Delta L_C(90^\circ) = \frac{k_1 \cdot R_C \cdot P_0}{g_m} = \frac{2.2768 \cdot 10^{-5}}{\left[1 - 0.00266 \cdot \cos(2\varphi) - 0.00028 \cdot 10^{-3} \cdot h\right]} \cdot P_0, \qquad (1.21)$$

где $\varphi$ – широта места;
$h$ – высота над уровнем моря, м;
$P_0$ – атмосферное давление у поверхности Земли, Па.

Точность определения гидростатической задержки радиосигнала определяется точностью информации об атмосферном давлении и составляет 2.3 мм/гПа.

Отображающая функция по определению равна:

$$m_C(\beta) = \frac{\Delta L_C(\beta)}{\Delta L_C(90^\circ)} = \frac{\int_0^{L} \rho dl}{\int_0^{\infty} \rho dz}. \qquad (1.22)$$

Ифадисом [11] получена эмпирическая зависимость, применимая в диапазоне углов места от 90º до 5º (при $\beta = 5^\circ$ среднеквадратическая ошибка определения задержки составляет 2.2 см):

$$m_C(\beta) = \frac{1}{\sin\beta + \dfrac{a}{\sin\beta + \dfrac{b}{\sin\beta + c}}}, \qquad (1.23)$$

где $a = 1.237 \cdot 10^{-3} + 1.316 \cdot 10^{-9} \cdot \left(P_0 - 10^5\right) + 1.378 \cdot 10^{-6} \cdot \left(T_0 - 288.15\right) +$

$+ 8.057 \cdot 10^{-7} \cdot \sqrt{e_0}$;

$b = 3.333 \cdot 10^{-3} + 1.946 \cdot 10^{-9} \cdot \left(P_0 - 10^5\right) + 1.040 \cdot 10^{-7} \cdot \left(T_0 - 288.15\right) +$

$+ 1.747 \cdot 10^{-8} \cdot \sqrt{e_0}$;

$c = 0.078$;

$T_0$ – приземная температура воздуха, К;

$e_0$ – приземное значение парциального давления водяного пара, Па.

*Задержка радиосигнала во влажном воздухе*

При прохождении сигнала через атмосферу, содержащую водяной пар, возникает задержка радиосигнала, равная:

$$\Delta L_П(\beta) = \int_0^L \left( k_2' \cdot R_П + \frac{k_3 \cdot R_П}{T} \right) \cdot \rho_П dl = \Delta L_П(90^\circ) \cdot m_П(\beta). \qquad (1.24)$$

где $\Delta L_П(90^\circ)$ – задержка радиосигнала в водяном паре при вертикальном распространении;

$m_П(\beta)$ – отображающая функция, зависящая от угла места навигационного спутника.

Для расчета задержки сигнала при вертикальном распространении, обусловленной наличием водяного пара в атмосфере, необходимо интеграл представить в виде:

$$\Delta L_П(90^\circ) = \int_0^\infty \left( k_2' \cdot R_П + \frac{k_3 \cdot R_П}{T} \right) \cdot \rho_П dz = \left( k_2' \cdot R_П + \frac{k_3 \cdot R_П}{T_m} \right) \cdot \int_0^\infty \rho_П dz, \qquad (1.25)$$

где $T_m$ – взвешенное значение температуры воздуха, К.

Поскольку интегральное содержание водяного пара в столбе воздуха определяется выражением $v = \int_0^\infty \rho_П dz$, то получается:

$$\Delta L_П(90^\circ) = \left( k_2' \cdot R_П + \frac{k_3 \cdot R_П}{T_m} \right) \cdot v \,. \qquad (1.26)$$

Так как по определению параметр $T_m$ равен [10]:

$$T_m = \frac{\int_0^\infty \frac{e}{T} dz}{\int_0^\infty \frac{e}{T^2} dz} \,, \qquad (1.27)$$

то значения данного параметра определяются вертикальным распределением влажности и сильно зависят от приземной температуры воздуха. На основе статистической обработки данных радиозондирования над территорией США в диапазоне широт от 27º до 65º с.ш. получена следующая зависимость [12]:

$$T_m = 70.2 + 0.72 \cdot T_0 \,, \qquad (1.28)$$

где $T_0$ – приземная температура воздуха (на высоте 2 м), К.

Задержка радиосигнала во влажном воздухе зависит в первую очередь от значений интегрального содержания водяного пара (0.6 мм/кг/м$^2$) и в меньшей мере от приземного значения температуры воздуха (0.5 мм/К).

Отображающая функция, показывающая зависимость задержки сигнала во влажном воздухе от угла места спутника, может быть определена по формуле Ифадиса [11]:

$$m_{\Pi}(\beta) = \frac{1}{\sin\beta + \dfrac{a}{\sin\beta + \dfrac{b}{\sin\beta + c}}}, \qquad (1.29)$$

где $a = 5.236 \cdot 10^{-3} + 2.471 \cdot 10^{-9} \cdot \left(P_0 - 10^5\right) + 1.724 \cdot 10^{-7} \cdot \left(T_0 - 288.15\right) +$

$+ 1.328 \cdot 10^{-6} \cdot \sqrt{e_0}$,

$b = 1.705 \cdot 10^{-3} + 7.384 \cdot 10^{-9} \cdot \left(P_0 - 10^5\right) + 3.767 \cdot 10^{-7} \cdot \left(T_0 - 288.15\right) +$

$+ 2.147 \cdot 10^{-6} \cdot \sqrt{e_0}$,

$c = 0.05917$.

На основе полученных формул можно формулу определения задержки в тропосфере можно представить в виде:

$$\Delta L_{\grave{O}\grave{D}} = m_c(\beta) \cdot k_1 \cdot R_c \cdot \int_0^{\infty} \rho \cdot dz +$$
$$+ m_{\ddot{I}}(\beta) \cdot k_1 \cdot R_c \cdot \int_0^{\infty} (k'_2 \cdot R_{\ddot{I}} + \frac{k_3 \cdot R_{\ddot{I}}}{\grave{O}}) \cdot \rho_{\ddot{I}} \cdot dz, \qquad (1.33)$$

где $k_1, k_2, k_3$ – эмпирические коэффициенты (см. таблицу 1.2);

$m_C(\beta)$ – отображающая функция, зависящая от угла места навигационного спутника;

$m_{\Pi}(\beta)$ – отображающая функция, зависящая от угла места навигационного спутника;

$T$ – температура воздуха, К;

$\rho_{\ddot{I}}$ – абсолютная влажность;

$\beta$ – угол места спутника.

Выводы

В первой главе рассмотрены основные характеристики глобальных навигационных спутниковых системам GPS, ГЛОНАСС, описан принцип работы спутниковых систем навигации, указаны виды погрешностей определения дальности до спутника, показано влияние среды на распространение навигационных сигналов (ионосферные, тропосферные задержки радиосигналов). Приведены формулы для вычисления задержек радиосигнала во влажном воздухе. В следующей главе более подробно рассматривается методы определения положения спутников в пространстве, необходимые для измерения задержек радиосигнала в тропосфере.

# 2 Методы вычисления местоположения спутников в пространстве

## 2. 1 Параметры движения спутников по орбите

Искусственный спутник Земли (ИСЗ) – искусственно созданное небесное тело, обращающееся вокруг Земли только под действием силы ее притяжения, без участия каких-либо других сил.

Уравнение движения спутников описываются тремя дифференциальными уравнениями второго порядка [13]:

$$\frac{d^2\tilde{o}}{d^2t} = -\gamma \cdot M \cdot \frac{x}{r^3}, \quad (2.1)$$

$$\frac{d^2y}{d^2t} = -\gamma \cdot M \cdot \frac{y}{r^3}, \quad (2.2)$$

$$\frac{d^2z}{d^2t} = -\gamma \cdot M \cdot \frac{z}{r^3} \quad (2.3)$$

где $x, y, z$ – координаты центра спутника по отношению к центру Земли;
$M$ – общая масса тел, участвующих в движении;
$r$ – расстояние между центрами тел.
$\gamma$ – универсальная гравитационная постоянная, $\gamma = 6.668 \cdot 10^{-8}$ см$^3$/(г$\cdot$ с$^2$)

Плоскость, по которой движется спутник, называется орбитальной плоскостью. Пассивное движение в центральном поле тяготения называется *кеплеровским движением*, а эллиптические, параболические и гиперболические траектории – *кеплеровскими орбитами*.

В общем случае орбита ИСЗ является не круговой, а эллиптической, которая характеризуется большой (*а*) и малой (*в*) полуосью эллипса (см. рис. 2.1). Перигей (*П*) – ближайшая к центру Земли точка орбиты; апогей (*А*) – наиболее удаленная точка орбиты ИСЗ от центра Земли.

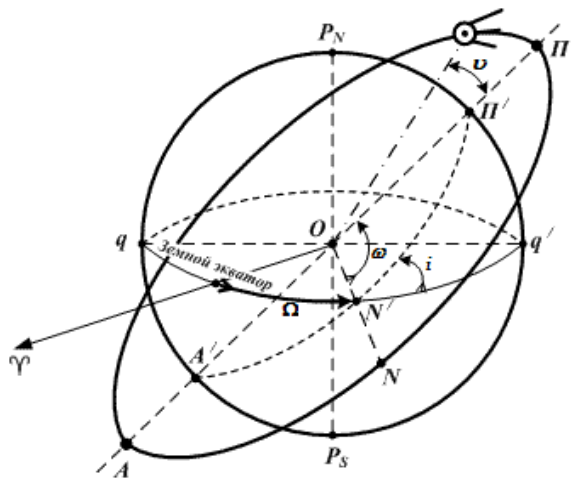



Рисунок 2.1 – Характеристика эллиптической орбиты ИСЗ

Эллиптическая орбита характеризуется 6-ю элементами Кеплера (рис. 2.1):

1) наклонение плоскости орбиты к плоскости экватора ($i$) - угол между плоскостью земного экватора и плоскостью орбиты ИСЗ;
2) прямое восхождение восходящего узла орбиты ($\Omega$) - угол между направлениями на точку Овна и восходящий узел орбиты ИСЗ;
3) угловое расстояние перигея от восходящего узла орбиты $N$ ($\omega$) → угол между направлениями на восходящий узел орбиты ИСЗ и точку его перигея;
4) эксцентриситет орбиты ($e$):

$$e = \frac{\sqrt{a^2 - b^2}}{a^2}, \tag{2.4}$$

5) параметр орбиты:

$$p = \frac{b^2}{a}, \tag{2.5}$$

6) время прохождения ИСЗ через перигей ($П$) или восходящий ($N$) узел ($\tau$).

Содержание элементов с 1 по 5 характеризуют пространственное положение орбиты ИСЗ и направление его движения.

Элемент № 6 ($\tau$) – характеризует положение ИСЗ на орбите.

Для определения места по ИСЗ необходимо знать его координаты в момент наблюдений. По известным значениям параметров орбиты ИСЗ можно вычислить его пространственные координаты и положение его относительно земной поверхности на любой момент времени.

Мгновенное положение ИСЗ относительной земной поверхности можно определить координатами *геозенита* – точки на земной поверхности, для которой ИСЗ в данный момент находится в зените и длиной радиус-вектора его орбиты:

$$\rho_0 = R + H, \tag{2.6}$$

где $\rho_0$ – радиус орбиты ИСЗ (расстояние до ИСЗ от центра Земли);

В геометрической интерпретации именно точка геозенита является опорным пунктом, относительно которого определяются навигационные параметры.

В соответствии с законом Кеплера скорость движения ИСЗ по эллиптической орбите определяется соотношением:

$$v = \sqrt{k \cdot (2 \cdot r - 1/a)}\ , \tag{2.7}$$

где $k$ – гравитационный параметр, характеризующий поля тяготения (для Земли $k=3.986 \cdot 10^5$ км$^3$/с$^2$).

Скорость движения по круговой орбите постоянна и определяется по формуле:

$$Vкр = \sqrt{k / r} \tag{2.8}$$

Скорость, при которой начинается движения ИСЗ по окружности вокруг Земли, называется круговой или первой космической скоростью. При $H = 0 \rightarrow V_1 = 7.91$ км/с.

В случае предельно вытянутой эллиптической орбиты формула имеет следующий вид:

$$Vосв = \sqrt{2 \cdot k / r} \tag{2.9}$$

Значение скорости освобождения (параболической скорости) у поверхности Земли носит название второй космической скорости и составляет 11.19 км/с.

Период обращения ИСЗ вокруг Земли определяется по формуле:

$$T = \frac{2\pi \cdot \rho_0}{v_1}. \tag{2.10}$$

В зависимости от высоты орбиты (*H*) значения скорости ИСЗ ($V_1$) и его период (*T*) различны (см. табл. 2.1) [13]:

Таблица 2.1 – Зависимость значений скорости ИСЗ ($V_1$) и его периодов (*T*) от высоты орбиты (*H*)

| *H*, км | 0 | 250 | 1000 | 20240 | 35870 |
|---|---|---|---|---|---|
| *T*, мин. | 84 | 89 | 106 | 12 час | 24 час |
| $V_1$, км/с | 7.91 | 7.76 | 7.35 | 3.87 | 3.07 |

Вследствие того, что Земля не шар, а геоид и того, что на ИСЗ влияет ряд возмущающих сил (сопротивление атмосферы Земли, притяжение Луны и Солнца, давление солнечной радиации, магнитное поле Земли), кеплеровские элементы орбиты ИСЗ непрерывно изменяются.

## 2.2 Система координат WGS-84

WGS84 (англ. World Geodetic System 1984) – трёхмерная система координат для позиционирования на Земле. Её важность заключается в том, что она в отличие от локальных систем является единой системой для всей Земли. Предшественниками WGS-84 были системы WGS-72, WGS-64 и WGS-60. WGS-84 определяет координаты относительно центра масс Земли, ошибка составляет менее 2 см. В WGS-84, нулевым меридианом считается IERS Reference Meridian. Он расположен на 5.31 секунды к востоку от Гринвичского меридиана.

Всемирная геодезическая система является стандартом для использования в картографии, геодезии и навигации. Она включает в себя стандартные координат на Земле, стандартный сфероидальных ссылкой поверхности (в нуль или ссылки эллипсоидом) для сырых высоте данных, а также гравитационных эквипотенциальных поверхности (на геоида), который определяет номинальный уровень моря.

В последней редакции является WGS-84 (начиная с 1984 года и последний раз пересматривался в 2004 году), которые будут действительны до примерно 2010 года [15]. Ранее схемы включали WGS-72, WGS-66, и WGS-60. WGS-84 является исходной системе координат, используемых в глобальной системе позиционирования.

Начальный пункт системы WGS-84 OXYZ расположен в центре масс Земли с ориентацией общего земного эллипсоида (ОЗЭ) вдоль главных осей Земли: ось OZ направлена вдоль оси вращения Земли в сторону Северного полюса, ось OX лежит на пересечении плоскостей экватора и Гринвичского меридиана, ось OY дополняет систему координат до правой.

Согласование системы координат, связанной с эфемеридами спутниковой навигационной системы регулярно осуществляется тем, что по результатам наблюдений на постоянно действующих станциях мировой сети GPS для геодинамики (IGS – International GPS Survice for Geodynamics), координаты пунктов которой определены в СК WGS-84, определяются

эфемериды спутников GPS. В процессе многолетних наблюдений с использованием лазерной локации спутников и радиоинтерферометрии уточняются координаты следящих станций IGS. Эти уточнения периодически принимаются в качестве более современных и точных реализаций СК WGS-84. В свою очередь и текущие эфемериды спутников GPS определяются в соответствующих реализациях WGS-84.

Например, ITRF 96 – это система координат WGS-84, скорректированная на данные, накопленные к 1996 г.

Таким образом, ITRF является совокупностью координат и скоростей перемещения станций глобальной спутниковой сети GPS для геодинамики. Эта координатная основа реализуется ежегодно по данным спутниковых наблюдений на более чем 150 станциях, с точностью порядка 12 см. Международная служба GPS для геодинамики регулярно обеспечивает ITRF точными эфемеридами спутников и другими необходимыми данными. Многие государства, в частности, европейские, используют данные ITRF для установления собственных национальных геодезических систем, а также для изучения движений, деформаций земной поверхности и изменений уровней морей. Одним из основных результатов деятельности IGS являются точные эфемериды спутников глобальных навигационных систем, определяемые в настоящее время с ошибкой порядка 0.1 – 0.2 м.

## 2.3 Система координат ПЗ-90

Параметры Земли 1990 года (ПЗ-90) – государственная геоцентрическая система координат, использующаяся в целях геодезического обеспечения орбитальных полётов и решения навигационных задач (в частности, для обеспечения работы глобальной навигационной спутниковой системы ГЛОНАСС) [16].

В настоящее время начинает внедряться уточнённая версия – ПЗ-90.02, представляющая собой систему «взаимосогласованных геодезических параметров, включающих фундаментальные геодезические постоянные, параметры общеземного эллипсоида, параметры гравитационного поля Земли, общеземную систему координат и параметры её связи с другими системами координат по состоянию на 1 января 2002 года».

Эта геоцентрическая система OXYZ, ориентированная аналогично WGS-84, предназначалась для функционирования навигационной спутниковой системы ГЛОНАСС.

Отсчетной поверхностью является эллипсоид вращения - общий земной эллипсоид (ОЗЭ), параметры которого приведены в таблице 2.2.

С 1985 г. в России функционирует специальный космический геодезический комплекс (КГК) ГЕОИК, с помощью которого осуществляются дальномерные траекторные наблюдения в КГС (космическая геодезическая сеть) и альтиметрические измерения морской поверхности. К 1988-1989 гг. была накоплена измерительная информация КГК ГЕОИК, по которой в ВТУ ГШ МО была выведена координатная система «Параметры Земли 1990 г.». Носителями этой системы являются пункты КГС; 26 таких пунктов расположены на территории бывшего СССР.

В таблице 2.2 приведены фундаментальные постоянные WGS-84, ПЗ-90 и постоянные, установленные комиссией МАГ (Международная ассоциация геодезии) по фундаментальным постоянным [14].

Таблица 2.2 – Фундаментальные постоянные систем координат

| Фундаментальные постоянные | ПЗ-90 | WGS-84 | Установленные комиссией МАГ |
|---|---|---|---|
| Скорость света с, м·с$^{-1}$ | 299 792 458 | 299 792 458 | 299 792 458 |
| Угловая скорость вращения Земли, рад·с$^{-1}$ | 7 292 115·10$^{-11}$ | 7 292 115·10$^{-11}$ | 7 292 115·10$^{-11}$ |
| Геоцентрическая гравитационная постоянная GM, м$^{3}$·с$^{-2}$ | 398 600.44· 10$^{9}$ | 398 600.5· 10$^{9}$ | — |
| Большая полуось эллипсоида а | 6 378 136 м | 6 378 137 м | (6 378 136.6±0,1) м |
| Полярное сжатие | 1/ 298.257 839 303 | 1/ 298.257223563 | 1/ 298,25231 ± 0.00001 |
| Нормальный потенциал Uo, м$^{2}$·с$^{-2}$ | 62 636 861.074 | — | 62636857.0±1.0 |
| Нормированный зональный гармонический коэффициент гравитационного потенциала С20 | -484,164953· 10$^{-6}$ | -484,16685· 10$^{-6}$ | — |

Передаваемые каждым НКА системы ГЛОНАСС в составе оперативной информации эфемериды описывают положение фазового центра передающей антенны данного НКА в связанной с Землей геоцентрической системекоординат ПЗ-90.02, определяемой следующим образом:

- начало координат расположено в центре масс Земли;
- ось Z направлена на Условный полюс Земли, как определено в рекомендации Международной службы вращения Земли (IERS);
- осЬ X направлена в точку пересечения плоскости экватора и начального (нулевого) меридиана, установленного Международным бюро времени;
- ось Y дополняет геоцентрическую прямоугольную систему координат до правой;
- в этой системе координат положение точки в пространстве определяется значениями координат X, Y, Z;
- геодезические координаты точки в системе координат ПЗ-90.02 относятся к эллипсоиду;

– геодезическая широта B точки M определяется как угол между нормалью к поверхности эллипсоида и плоскостью экватора;
– геодезическая долгота L точки M определяется как угол между плоскостью нулевого меридиана и плоскостью меридиана, проходящего через точку M. Положительное направление счета долгот - от нулевого меридиана к востоку от 0 до 360 градусов;
– геодезическая высота H определяется как расстояние по нормали от поверхности эллипсоида до точки M.

## 2. 4 Система координат СК-95

Система СК-95 связана с отсчетным эллипсоидом Красовского. Является уточнением системы координат СК-42. Элементы ориентирования этого эллипсоида должны сохранять неизменными координаты исходного пункта Пулково. Ориентация осей эллипсоида Красовского принята совпадающей с ориентацией общего земного эллипсоида (ОЗЭ) в системе координат ПЗ-90 (КГС).

Эллипсоидальные координаты - геодезические широта (B), долгота (L) и высота (H) связаны с декартовыми прямоугольными и прямолинейными координатами следующими формулами [15]:

$$X = (N + H) \cdot \cos B \cdot \cos L \qquad (2.11)$$

$$Y = (N + H) \cos B \cdot \sin L \qquad (2.12)$$

$$Z = [(b^2/a^2 \cdot N + H)] \cdot \sin B \qquad (2.13)$$

где a, b – полуоси эллипсоида,
$N = a^2/ p$,
$p^2 = a^2 \cdot \cos^2 B + b^2 \cdot \sin^2 B$.

Носителем этой системы координат на общем земном эллипсоиде являются 164 тысячи пунктов совместно уравненных сетей КГС, ДГС, АГС, покрывающих всю территорию бывшего СССР, из которых 134 пункта сетей

КГС и ДГС совмещены с пунктами АГС при среднем расстоянии между ними 400-450 м.

Пункты КГС определены по доплеровским, фотографическим, дальномерным радиотехническим и лазерным наблюдениям системы ГЕОиК. Средняя квадратическая ошибка определения взаимного положения при среднем расстоянии между ними около 1- 1.5 тыс. км имеет величину порядка 0.2-0.3 м. Пункты доплеровской сети (ДГС) определялись по доплеровским наблюдениям ИСЗ системы TRANZIT. Средние квадратические ошибки взаимного положения пунктов при средних расстояниях между ними 500 - 700км составляют 0.4-0.6 м. В России готовится создание качественно новой государственной системы координат, которое завершится после построения ФАГС, ВГС и в значительном объеме СГС-1, как это предусмотрено переводом системы геодезического обеспечения России на спутниковые методы в «Федеральной целевой подпрограмме по использованию глобальной навигационной спутниковой системы ГЛОНАСС для геодезического обеспечения территории России»

Правительственным постановлением №568 от 28 июля 2000 г. системы координат СК-95 и ПЗ-90 приняты в качестве общегосударственных систем координат.

## 2.5 Переход от одной системы координат к другой

Системы координат WGS-84 и ПЗ-90 относятся к геоцентрическим системам координат, а СК-95 к географическим (геодезическим). Координаты любой точки земной поверхности в разных системах координат будут различаться, переход от одной системы координат к другой осуществляется с помощью формул преобразований.

### 2.5.1 Преобразования из геоцентрических в геоцентрические координаты.

Эти преобразования могут использоваться как посредник между преобразованием из географических в географические координаты по схеме (см. рис. 2.2):

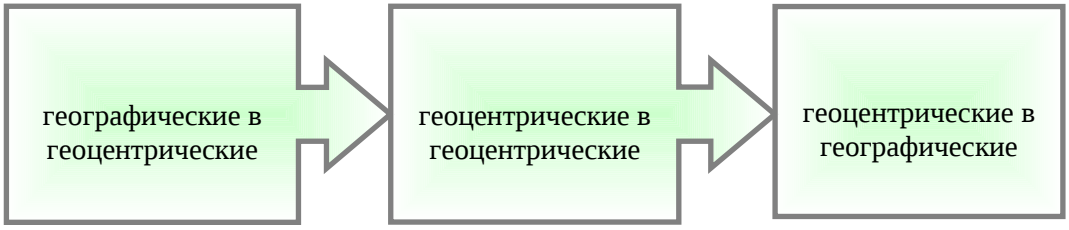



Рисунок 2.2 – Схема преобразования географических координат в географические.

*Геоцентрическое преобразование*

Параметры преобразования:

- смещение по оси X;
- смещение по оси Y;
- смещение по оси Z.

Если исходная и конечная система координат геоцентрические, оси эллипсоидов параллельны, главный меридиан – Гринвичский и нет разницы в масштабах, это преобразование позволяется вычислить координаты в конечной системе координат простым прибавлением смещения соответствующим координатам в исходной системе координат (2.12)-(2.14).

$$X^t = X^s + dX, \quad (2.14)$$
$$Y^t = Y^s + dY, \quad (2.15)$$
$$Z^t = Z^s + dZ. \quad (2.16)$$

*Векторное положение.*

Параметры преобразования:

- смещение по оси X;

- смещение по оси Y;
- смещение по оси Z;
- поворот по оси X;
- поворот по оси Y;
- поворот по оси Z;
- масштабирование.

Одно из 7-параметрических преобразований Хельмерта, использующее формулу Бурса-Вольфа [16]:

$$\begin{pmatrix} X_t \\ Y_t \\ Z_t \end{pmatrix} = M \begin{pmatrix} 1 & -R_z & +R_y \\ +R_z & 1 & -R_x \\ -R_y & +R_x & 1 \end{pmatrix} \begin{pmatrix} X_s \\ Y_s \\ Z_s \end{pmatrix} + \begin{pmatrix} dX \\ dY \\ dZ \end{pmatrix}, \qquad (2.16)$$

где $X_s$, $Y_s$, $Z_s$ – координаты точки в исходной системе координат;

$X_t$, $Y_t$, $Z_t$ – координаты точки в конечной системе координат:

dX, dY, dZ – вектор смещения добавляемый к исходной точке, также является координатами начала координат исходной системы координат в конечной системе координат;

$R_x$, $R_y$, $R_z$ – повороты добавляемые к вектору смещения, положительное значение означает поворот по часовой стрелке исходя из начала координат вдоль положительного хода соответствующей оси. Углы измеряются в радианах;

M – масштабирование вектора преобразования в исходной системе координат необходимое, чтобы получить правильный масштаб в конечной системе.

$M = 1+dS\cdot10^{-6}$,

где dS – масштабирование выражаемое в частях на миллион.

Это преобразование может использоваться как промежуточное между преобразованием из географических в географические координаты.

*Координаты системы вращения.*

Параметры преобразования:

- смещение по оси X;
- смещение по оси Y;
- смещение по оси Z;
- поворот по оси X;
- поворот по оси Y;
- поворот по оси Z;
- масштабирование.

В качестве параметрических преобразований Хельмерта используется формула (2.16).

Преобразование аналогичное векторному положению, но отличающееся инвертированными значениями поворотов $R_x$, $R_y$, $R_z$. Международная ассоциация геодезии (МАГ) и международный стандарт ISO 19111 (Географическая информация – Пространственные привязки по координатам) рекомендуют для описания преобразования использовать векторное положение. Это преобразование может использоваться как посредник между преобразованием из географических в географические координаты.

2.5.2 Преобразования из географических в географические координаты

Данный вид преобразований позволяет перейти от географических координат к географическим сразу, без этапа пересчета из одной геоцентрической системы координат в другую. Параметры перехода из геоцентрической в геоцентрическую СК используются как часть общего набора параметров. При использовании метода Молоденского [16] используются следующие параметры преобразования:

- смещение по оси X;
- смещение по оси Y;
- смещение по оси Z;
- разница в длине малой полуоси;
- разница в уплощении.

Система дифференциальных уравнений преобразований Молоденского (2.17)-(2.19):

$$d\varphi = -dX \cdot \sin\varphi \cdot \cos\lambda - dY \cdot \sin\varphi \cdot \sin\lambda + dZ \cdot \cos\varphi + da \cdot \frac{v \cdot e^2 \cdot \sin\varphi \cdot \cos\varphi}{a} + \\ + df \cdot (\rho \cdot (\frac{a}{b}) + v \cdot (\frac{b}{a})) \cdot \sin\varphi \cdot \cos\varphi, \quad (2.17)$$

$$d\lambda = \frac{-dX \cdot \sin\lambda + dY \cdot \cos\lambda}{(v+h) \cdot \cos\varphi \cdot \sin 1''}, \quad (2.18)$$

$$dh = dX \cdot \cos\varphi \cdot \cos\lambda + dY \cdot \cos\varphi \cdot \sin\lambda + dZ \cdot \sin\varphi - da \cdot (\frac{a}{v}) + df \cdot (\frac{b}{a}) \cdot v \cdot \sin^2\varphi, \quad (2.19)$$

где $\varphi$ – разница по широте в угловых секундах;
$\lambda$ – разница долготе в угловых секундах;
$h$ – разница по высоте, м;
dX, dY, dZ – параметры геоцентрического смещения
ρ – горизонтальный (меридиональный) радиус кривизны на данной широте первого эллипсоида
ν – вертикальный (широтный) радиус кривизны на данной широте первого эллипсоида
da – разница между длинами малых полуосей ($a_1$ - $a_2$) исходного и конечного эллипсоидов
df – разница между уплощениями этих эллипсоидов.

Горизонтальный и вертикальные радиусы кривизны первого эллипсоида можно найти по формулам (2.20), (2.21), и разницу межу уплощениями исходного и конечного эллипсоидов (2.22) :

$$\rho = \frac{a \cdot (1 - e^2)}{(1 - e \cdot \sin^2 \cdot \varphi)^{\frac{3}{2}}}, \tag{2.20}$$

$$v = \frac{a}{(1 - e \cdot \sin^2 \varphi)^{\frac{1}{2}}}, \tag{2.21}$$

$$df = f_t - f_s = \frac{1}{1/f_t} - \frac{1}{1/f_s}, \tag{2.22}$$

где $f_s$ – уплощение исходного эллипсоида
$f_t$ – уплощение конечного эллипсоида
е – эксцентриситет.

2.5.3 Преобразование между системами координат WGS-84 и ПЗ-90

Различия между двумя системами координат обусловлены различиями в фундаментальных геодезических константах и параметрах общеземных эллипсоидов. Фундаментальные геодезические константы и основные параметры общеземных эллипсоидов, принятые в системе координат ПЗ-90 и WGS-84 были представлены в табл. 2.2.

Но даже если бы геодезические константы и параметры эллипсоидов были идентичны, реализация систем координат двумя различными СРНС будет различна. По пунктам с известными координатами в ПЗ-90 и WGS-84 определяют набор параметров преобразования между двумя системами.

Параметры преобразования за счёт поворота относительно осей Z и Y и смещения начала координат приводят к соотношениям, которые могут быть представлены в матричном виде [3]:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{WGS-84} = \begin{bmatrix} 0 \\ 2.5 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & -0.39'' & 0 \\ 0.39'' & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{ÏÇ-90}, \tag{2.16}$$

Используя набор ГЛОНАСС спутниковых положений, определенных в ПЗ-90 и WGS-84, мы можем также установить параметры перехода от одной системы координат к другой [18]:

$$,\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{WGS-84} = \begin{bmatrix} 1 & -0.39'' & 0 \\ 0.39'' & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{ÏÇ-90}, \qquad (2.17)$$

Смещение координат точки на земной поверхности в системах ПЗ-90 и WGS-84 относительно друг друга в среднем не превышает 5 метров [3].

## 2.6 Методы вычисления положения спутника в пространстве

Эффективным способом представления орбитального движения НКА ГЛОНАСС/GPS является представление в элементах орбиты в силу того, что единственным измененяемым параметром при невозмущённом движении является средняя аномалия в эпоху – M. Обозначим в качестве начальной эпохи момент $t_0$, а в качестве эпохи, на которую необходимо получить прямоугольные координаты НКА, – t. Тогда средняя аномалия на любой момент излучения электромагнитного сигнала НКА

$$M = M_0 + n \cdot (t - t_0), \qquad (2.18)$$

где n – среднее движение НКА или средняя угловая скорость НКА по орбите

$$n = \frac{\sqrt{\mu}}{a \cdot \sqrt{a}}, \qquad (2.19)$$

Методом последовательных приближений находят эксцентрическую аномалию:

$$E_k + 1 = M + e \cdot \sin(E_k), \qquad (2.20)$$

В качестве начального приближения $E_1$ берут среднюю аномалию, то есть $E_1 = M$.

По формуле Бруте Чефола находят численное значение для υ – E:

$$v - E = 2 \cdot arctg\left(\frac{e \cdot \sin(E)}{1 + \sqrt{1 - e^2} - e \cdot \cos(E)}\right) \qquad (2.21)$$

На основании соотношений (2.18) и (2.21) мы вправе записать:

$$v = E + (v - E), \qquad (2.22)$$

Радиус-вектор НКА на момент обсервации находим из следующего соотношения:)

$$r = \frac{a \cdot (1 - e^2)}{1 + e \cdot \cos(v)}, \qquad (2.23)$$

Пусть $U$ – аргумент широты, тогда

$$U = \upsilon + \omega, \qquad (2.24)$$

Пусть cos (r, X) = α, cos (r, Y) = β, cos (r, Z) = γ. Распишем через элементы орбиты α, β, γ:

$$\alpha = \cos(U) \cdot \cos(\Omega) - \sin(U) \cdot \sin(\Omega) \cdot \cos(i), \qquad (2.25)$$

$$\beta = \cos(U) \cdot \sin(\Omega) - \sin(U) \cdot \cos(\Omega) \cdot \cos(i), \qquad (2.26)$$

$$\gamma = \sin(U) \cdot \sin(i), \qquad (2.27$$

$$\alpha^2 + \beta^2 + \gamma^2 = 1, \qquad (2.28)$$

Тогда:

$$X = r \cdot \alpha, \qquad (2.29)$$

$$Y = r \cdot \beta, \qquad (2.30)$$

$$Z = r \cdot \gamma, \qquad (2.31)$$

Продифференцировав по времени выражения (2.29) – (2.31), получим формулы для проекций скорости НКА на оси координат [2]:

$$V_x = \left(\frac{X}{r}\right) \cdot V_r + (\sin(U) \cdot \cos(\Omega) - \cos(U)\sin(\Omega)\cos(i)) \cdot V_n, \qquad (2.32)$$

$$V_y = \left(\frac{Y}{r}\right) \cdot V_r + (-\sin(U) \cdot \sin(\Omega) + \cos(U)\cos(\Omega)\cos(i)) \cdot V_n, \qquad (2.33)$$

$$V_z = \left(\frac{Z}{r}\right) \cdot V_r + (\cos(U) \cdot \sin(\Omega) \cdot \cos(i)) \cdot V_n, \qquad (2.34)$$

где $V_r$ – радиальная скорость НКА, $V_n$ – трансверсальная скорость НКА:

$$V_r = \sqrt{\frac{\mu}{p}} \cdot e \cdot \sin(v), \qquad (2.35)$$

$$V_n = \sqrt{\frac{\mu}{p}} \cdot (1 + e \cdot \cos(v)), \qquad (2.36)$$

где p – параметр орбиты, определяемый по формуле

$$p = a \cdot (1 - e^2), \qquad (2.37)$$

Выводы

Во второй главе рассмотрены параметры движения спутников по орбите, геоцентрические системы координат ПЗ-90 (Россия), WGS-84 (США) и географическая СК-95 (Россия), приведены их общие характеристики и различия, указаны способы преобразования параметров местоположения из одной системы координат в другую. Представлены основные формулы для вычисления положения спутника в пространстве на основе элементов Кеплера.

# 3 Форматы обмена и хранения навигационных данных

## 3. 1 Формат обмена навигационными данными RINEX

Стандарт RINEX широко используется *при обработке* ГНСС-данных в режиме постобработки для получения высокой точности определения местоположения.

Аппаратнонезависимый формата обмена навигационными данными (Receiver Independent Exchange Format, RINEX) был создан в Астрономическом институте Университета Берна в 1993г. В дальнейшем были разработаны версия 2.10 (1999 г.) и 2.11 (2004 г.) [19]. В настоящее время вводится в эксплуатацию версия 3.

Формат содержит семь типов ASCII-файлов:

1) Файл данных наблюдений;
2) Файл навигационных сообщений;
3) Файл метеорологических данных;
4) Файл навигационных сообщений ГЛОНАСС;
5) Файл навигационных сообщений GEO;
6) Файл данных часов спутников и приемников;
7) Файл широкозонной корректирующей информации SBAS.

Каждый тип файла состоит из *секции заголовка* и *секции данных*. Заголовок файла содержит глобальную информацию для всего файла и помещается в начале файла. Заголовок содержит текст меток заголовка в столбцах 61-80 для каждой строки заголовка. Эти метки обязательны и должны располагаться точно в той последовательности, как указано в описании и примерах.

Формат оптимизирован для минимизации объема хранимой информации, независимо от числа различных типов наблюдений,

производимых определенным приемником, путем указания в заголовке типов сохраненных данных наблюдений. В операционных системах, позволяющих хранить записи переменной длины, записи данных наблюдений могут быть сокращены насколько это возможно. Завершающие пробелы могут быть удалены из записей. Максимальная длина записи составляет 80 байт на запись.

Правила при указании имен RINEX файлов указаны на следующей схеме (рис. 3.1).

```
s s s s d d d f . y y t
|   |   |   |   |
|   |   |   |   + - -   t :   тип файла:
|   |   |   |   O :   Файл данных наблюдений
|   |   |   |   M :   Файл метеорологических данных
|   |   |   |   N :   Навигационный файл GPS
|   |   |   |   G :   Навигационный файл ГЛОНАСС
|   |   |   |   L :   Навигационный файл Galileo
|   |   |   |   H :   Навигационный файл GPS с геостационарных спутников
|   |   |   |   B :   Файл корректирующих данных SBAS
|   |   |   |   C:    Файл времени
|   |   |   |   S:    Обобщенный  файл
|   |   |   |
|   |   |   |
|   |   |   |
|   |   |   + - - -   yy: год (две цифры)
|   |   |
|   |   + - - - - -   f :   номер файла в течении дня (цифра или символ)
|   |   файл содержит все данные за текущий день: f = 0
|   |   данные за один час:
|   |   f = a: 1-ый час с 00 по 01; f = b: 2-ой час с 01 по 02;
|   |   f = x: 24-ый час с 23 по 24
|   |
|   + - - - - - - -   ddd: день в году для первой  записи
|
+ - - - - - - - - - - ssss: 4-символьное имя станции
```

Рисунок 3.1 – Правила при указании имен RINEX файлов

Для 15-минутных высокоскоростных данных схема представлена на рисунке 3.2.

```
s s s s d d d h m m . y y o
|   |   | |   |   |
|   |   | |   |   + -   o :   файл данных наблюдений
|   |   | |   |
|   |   | |   + - - -   y y :   год (две цифры)
|   |   | |
|   |   | + - - - - - -   m m :   первая минута наблюдений (00, 15, 30, 45)
|   |   |
|   |   + - - - - - - -   h :   символ для n-го часа в сутках
|   |   ( a =   1-ый час: с 00 по 01, b= 2-ой час: с 01 по 02,...,
|   |   x = 24-ый час: с 23 по 24. 0= за весь день)
|   |
|   + - - - - - - - - -   d d d :   день в году
|
+ - - - - - - - - - - - -   s s s s :   4-символьное имя приемника/антенны
```

Рисунок 3.2– Правила при указании имен RINEX файлов для 15-минутных высокоскоростных данных

Версия 2 предназначена для представления наблюдений ГЛОНАСС или других спутниковых систем. Для этого используется символ для каждой спутниковой системы и номер спутника (две цифры).

snn s: идентификатор спутниковой системы

| | |
|---|---|
| G или пробел | GPS |
| R : | ГЛОНАСС |
| E : | Galileo |
| S : | сигнал с геостационарных спутников |

nn: - PRN (GPS, Galileo), номер слота (ГЛОНАСС)
- PRN минус число 100 (GEO)

Пример файла навигационных сообщений GPS представлен на рисунке 3.3.

```
     2.10           NAVIGATION DATA                         RINEX VERSION /
TYPE
SPIDER V3,2,0,3217                      2009 06 02 01:00    PGM / RUN BY /
DATE
   6.5193D-09  2.2352D-08 -5.9605D-08 -1.1921D-07           ION ALPHA
   8.6016D  9.8304D -6.5536D -5.2429D           ION BETA
   -5.587935447693D-09-1.687538997430D-14   405504     1534 DELTA-UTC:
A0,A1,T,W
    15                                                      LEAP SECONDS
                                                            END OF HEADER
 1 08 03 17 23 59 12.0 1.929965801537D-04 2.614797267597D-12 0.000000000000D
    2.600000000000D 1.060937500000D 3.799443870633D-09-3.128013239404D
    5.569308996201D-06 7.017389754765D-03 8.894130587578D-06 5.153617078781D
    1.727520000000D 1.750886440277D-07-3.040775617584D1.136213541031D-07
    9.915466235883D-01 2.207812500000D1.781935002437D7.682462666025D-09
    2.250093722456D-10 1.000000000000D 1.471000000000D 0.000000000000D
    2.000000000000D 1.000000000000D3.725290298462D-09 2.600000000000D
    1.727520000000D 0.000000000000D
 2 09 06 02 00 00  0.0 1.553031615913D-04 9.094947017729D-13 0.000000000000D
    2.700000000000D 6.393750000000D 4.912347417729D-09 4.408143929574D-01
    3.194436430931D-06 9.050375665538D-03 8.951872587204D-06 5.153663352966D
    1.728000000000D 1.732259988785D-07-5.339438017711D-01-7.823109626770D-08
    9.414450777321D-01 1.970937500000D 2.828494234687D8.352848190896D-09
```

Рисунок 3.3 – Пример файла навигационных сообщений GPS

## 3. 2 Формат SP3

Формат Standard Product 3 (SP3) используется для точной оценки спутниковых эфемерид. Он является международным стандартом, регулирующим формат спутниковой информации, например, точные данные об орбите и спутниковых часов.

Система времени SP3 представляет собой 60-значный ASCII файл в двух разделах: заголовка и данных. В заголовке содержится информация о наблюдении сессии, таких как, как дата и количество спутников. Данные организованы в эпохи. Каждая эпоха содержит эфемериды и часы корректировки каждого спутника (GPS или ГЛОНАСС). Каждая строка из ASCII файла данных определяет свой спутник

Уточненный формат SP3 приведен на рисунке 3.4

```
        123456789012345678901234567890123456789012345678901234567890

1       #aV2006  5 28  0  0   .00000000      96 DD+AD WGS84 FIT  NGA
2       ## 1377       .00000000   900.00000000 53883  .0000000000000
3       +   29     1  2  3  4  5  6  7  8  9 10 11 13 14 15 16 17 18
4       +         19 20 21 22 23 24 25 26 27 28 29 30  0  0  0  0  0
5       +          0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
6       +          0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
7       +          0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
8       ++         0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
9       ++         0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
10      ++         0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
11      ++         0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
12      ++         0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
13      %c cc cc ccc ccc cccc cccc cccc cccc ccccc ccccc ccccc ccccc
14      %c cc cc ccc ccc cccc cccc cccc cccc ccccc ccccc ccccc ccccc
15      %f  0.0000000  0.000000000  0.00000000000  0.000000000000000
16      %f  0.0000000  0.000000000  0.00000000000  0.000000000000000
17      %i    0    0    0    0      0      0      0      0         0
18      %i    0    0    0    0      0      0      0      0         0
19      /*      NGA, ST. LOUIS,MO.
20      /*   EPHEMERIS COMPUTED FROM  3 DAYS OF DATA
21      /* CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
22      /* CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
23      *  2006  5 28  0  0   .00000000
24      P  1  13752.206309 -20369.024073  10204.025024     55.610230
25      V  1  10790.304814  -6827.236053 -28761.711131       .025297
26      P  2  -4207.781938  14727.204491  21440.989691       .356215
27      V  2 -26576.583126  -8421.718506    356.729184       .028568
28                .            .            .            .
29                .            .            .            .
30                .            .            .            .
31                .            .            .            .
32                     .                 .                 .
.
33                .            .            .            .
34      *  2006  5 28 23 45   .00000000
35      P  1  12973.835241 -19889.736391  12041.421093     55.838782
36      V  1  12878.846997  -7784.945215 -27217.727966       .025432
37                .            .            .            .
38                .            .            .            .
39                .            .            .            .
40                .            .            .            .
41                .            .            .            .
42                .            .            .            .
43      P 30  15849.387414   9776.120258  18703.529651    265.656142
44      V 30   3338.217385  24174.222185 -15120.093763       .139144
```

Рисунок 3.4 – Уточненный формат SP3

В SP3 формат выделяет точность до 1 мм и 1 пикосекунды, для скорости – $10^{-4}$ мм / сек.

Формат был модифицирован, разработаны форматы SP3-a, SP3-b, SP3-c. На 12 февраля 2007 года, этот файл SP3- с был обновлен, в него внесено дополнение.

## 3.3 Основные сведения о системах управления базами данных

Система управления базами данных (СУБД) - комплекс программных и лингвистических средств общего или специального назначения, реализующий поддержку создания баз данных, централизованного управления и организации доступа к ним различных пользователей в условиях принятой технологии обработки данных [20].

СУБД характеризуется используемой моделью, средствами администрирования и разработки прикладных процессов.

СУБД обеспечивает:

- описание и сжатие данных;
- манипулирование данными;
- физическое размещение и сортировку записей;
- защиту от сбоев, поддержку целостности данных и их восстановление;
- работу с транзакциями и файлами;
- безопасность данных.

СУБД определяет модель представления данных.

Основные *функции* СУБД представлены на рисунке 3.5

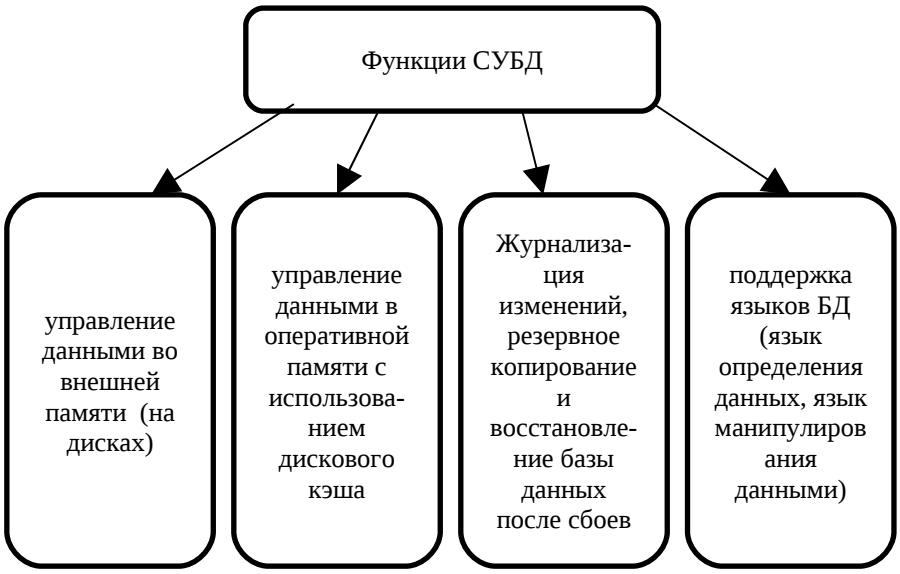



Рисунок 3.5- Основные функции СУБД

Обычно современная СУБД содержит следующие *компоненты*:

1) ядро, которое отвечает за управление данными во внешней и оперативной памяти и журнализацию,
2) процессор языка базы данных, обеспечивающий оптимизацию запросов на извлечение и изменение данных и создание, как правило, машинно-независимого исполняемого внутреннего кода,
3) подсистему поддержки времени исполнения, которая интерпретирует программы манипуляции данными, создающие пользовательский интерфейс с СУБД
4) а также сервисные программы (внешние утилиты), обеспечивающие ряд дополнительных возможностей по обслуживанию информационной системы.

СУБД можно классифицировать (см. рис. 3.6):

1) По *модели* данных

По типу управляемой базы данных СУБД разделяются на:

1) Иерархические

2) Сетевые
3) Реляционные
4) Объектно-реляционные
5) Объектно-ориентированные

2) По *архитектуре* организации хранения данных
1) локальные СУБД (все части локальной СУБД размещаются на одном компьютере)
2) распределенные СУБД (части СУБД могут размещаться на двух и более компьютерах)

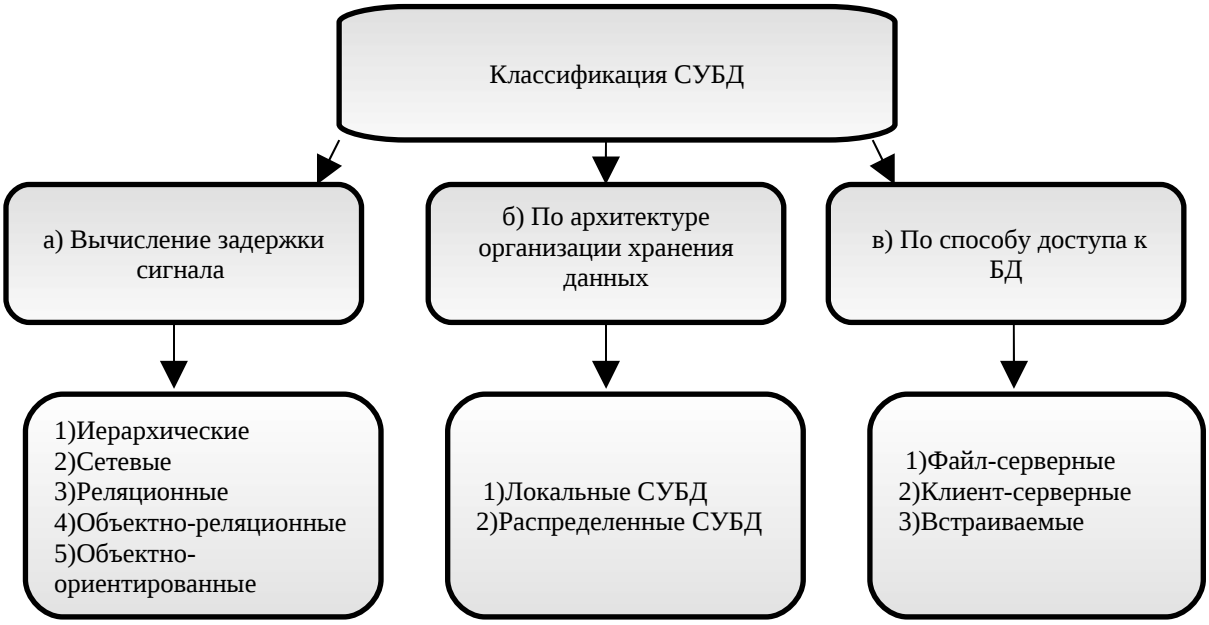



Рисунок 3.6 – Классификация СУБД

3) По способу доступа к БД
1) Файл-серверные

В файл-серверных СУБД файлы данных располагаются централизованно на файл-сервере. Ядро СУБД располагается на каждом клиентском компьютере. Доступ к данным осуществляется через локальную сеть. Синхронизация чтений и обновлений осуществляется

посредством файловых блокировок. Преимуществом этой архитектуры является низкая нагрузка на центральный процессор сервера, а недостатком – высокая загрузка локальной сети.

На данный момент файл-серверные СУБД считаются устаревшими.

Примеры: Microsoft Access, Borland Paradox.

2) Клиент-серверные

Такие СУБД состоят из клиентской части (которая входит в состав прикладной программы) и сервера. Клиент-серверные СУБД, в отличие от файл-серверных, обеспечивают разграничение доступа между пользователями и мало загружают сеть и клиентские машины. Сервер является внешней по отношению к клиенту программой, и по надобности его можно заменить другим. Недостаток клиент-серверных СУБД в самом факте существования сервера (что плохо для локальных программ – в них удобнее встраиваемые СУБД) и больших вычислительных ресурсах, потребляемых сервером.

Примеры: Firebird, Interbase, IBM DB2, MS SQL Server, Sybase, Oracle, PostgreSQL, MySQL, ЛИНТЕР.

3) Встраиваемые

Встраиваемая СУБД – библиотека, которая позволяет унифицированным образом хранить большие объёмы данных на локальной машине. Доступ к данным может происходить через SQL либо через особые функции СУБД. Встраиваемые СУБД быстрее обычных клиент-серверных и не требуют установки сервера, поэтому востребованы в локальном ПО, которое имеет дело с большими объёмами данных (например, геоинформационные системы).

Примеры: OpenEdge, SQLite, BerkeleyDB, один из вариантов Firebird, один из вариантов MySQL, Sav Zigzag, Microsoft SQL Server Compact, ЛИНТЕР.

## 3.4 Реляционные базы данных

Реляционная база данных - база данных, основанная на реляционной модели. Слово «реляционный» происходит от английского «relation» (отношение). Для работы с реляционными БД применяют *реляционные СУБД*.

Использование реляционных баз данных было предложено доктором Коддом из компании IBM в 1970 году.

Реляционная база данных характеризуются простотой структуры данных, удобным для пользователя табличным представлением и возможностью использования формального аппарата алгебры отношений и реляционного исчисления для обработки данных.

Реляционная модель ориентирована на организацию данных в виде двумерных таблиц. Каждая реляционная таблица представляет собой двумерный массив и обладает следующими свойствами:

- каждый элемент таблицы — один элемент данных
- все ячейки в столбце таблицы однородные, то есть все элементы в столбце имеют одинаковый тип (числовой, символьный и т. д.)
- каждый столбец имеет уникальное имя
- одинаковые строки в таблице отсутствуют
- порядок следования строк и столбцов может быть произвольным

Базовые понятия реляционных СУБД представлены в таблице 3.2

Таблица 3.1 – Базовые понятия реляционных СУБД

| Понятие | Применение |
|---|---|
| Атрибут | столбец или поле |
| Отношения | таблица |
| Кортеж | Для строки или записи |

## 3.5. Структуризованный язык запросов SQL

Все языки манипулирования данными (ЯМД), созданные до появления реляционных баз данных и разработанные для многих СУБД персональных компьютеров, были ориентированы на операции с данными, представленными в виде логических записей файлов. Это требовало от пользователей детального знания организации хранения данных и достаточных усилий для указания не только того, какие данные нужны, но и того, где они размещены и как шаг за шагом получить их.

Язык SQL (Structured Query Language - структуризованный язык запросов) ориентирован на операции с данными, представленными в виде логически взаимосвязанных совокупностей таблиц. Особенность предложений этого языка состоит в том, что они ориентированы в большей степени на конечный результат обработки данных, чем на процедуру этой обработки. SQL сам определяет, где находятся данные, какие индексы и даже наиболее эффективные последовательности операций следует использовать для их получения: не надо указывать эти детали в запросе к базе данных [20].

Появление теории реляционных баз данных и предложенного Коддом языка запросов "alpha", основанного на реляционном исчислении, инициировало разработку ряда языков запросов, которые можно отнести к двум классам:

Алгебраические языки, позволяющие выражать запросы средствами специализированных операторов, применяемых к отношениям (JOIN - соединить, INTERSECT - пересечь, SUBTRACT - вычесть и т.д.).

Языки исчисления предикатов представляют собой набор правил для записи выражения, определяющего новое отношение из заданной совокупности существующих отношений. Исчисление предикатов есть метод определения того отношения, которое желательно получить (как ответ на запрос) из отношений, уже имеющихся в базе данных.

В начале 80-х годов SQL стал фактическим стандартом языков для профессиональных реляционных СУБД. В 1987 году он стал международным стандартом языка баз данных и начал внедряться во все распространенные СУБД персональных компьютеров

Реализация в SQL концепции операций, ориентированных на табличное представление данных, позволило создать компактный язык с небольшим (менее 30) набором предложений. SQL может использоваться как интерактивный (для выполнения запросов) и как встроенный (для построения прикладных программ). В нем существуют:

- предложения определения данных (определение баз данных, а также определение и уничтожение таблиц и индексов);
- запросы на выбор данных (предложение SELECT);
- предложения модификации данных (добавление, удаление и изменение данных);
- предложения управления данными (предоставление и отмена привилегий на доступ к данным, управление транзакциями и другие). Кроме того, он предоставляет возможность выполнять в этих предложениях:
- арифметические вычисления (включая разнообразные функциональные преобразования), обработку текстовых строк и выполнение операций сравнения значений арифметических выражений и текстов;
- упорядочение строк и (или) столбцов при выводе содержимого таблиц на печать или экран дисплея;
- создание представлений (виртуальных таблиц), позволяющих пользователям иметь свой взгляд на данные без увеличения их объема в базе данных;

– запоминание выводимого по запросу содержимого таблицы, нескольких таблиц или представления в другой таблице (реляционная операция присваивания).
– агрегатирование данных: группирование данных и применение к этим группам таких операций, как среднее, сумма, максимум, минимум, число элементов и т.п.

В SQL используются основные типы данных, форматы которых могут несколько различаться для разных СУБД (см. табл. 3.2):

В некоторых СУБД еще существует тип данных LOGICAL, DOUBLE и ряд других. СУБД INGRES предоставляет пользователю возможность самостоятельного определения новых типов данных, например, плоскостные или пространственные координаты, единицы различных метрик, пяти- или шестидневные недели (рабочая неделя, где сразу после пятницы или субботы следует понедельник), дроби, графика, большие целые числа (что стало очень актуальным для российских банков) и т.п.

Ориентированный на работу с таблицами SQL не имеет достаточных средств для создания сложных прикладных программ. Поэтому в разных СУБД он либо используется вместе с языками программирования высокого уровня (например, такими как Си или Паскаль), либо включен в состав команд специально разработанного языка СУБД (язык систем dBASE, R:BASE и т.п.). Унификация полных языков современных профессиональных СУБД достигается за счет внедрения объектно-ориентированного языка четвертого поколения 4GL. Последний позволяет организовывать циклы, условные предложения, меню, экранные формы, сложные запросы к базам данных с интерфейсом, ориентированным как на алфавитно-цифровые терминалы, так и на оконный графический интерфейс (X-Windows, MS-Windows).

В качестве исходных данных была выбрана информация, получаемая с спутников GPS в формате RINEX.

Таблица 3.2 – Основные типы данных SQL

| | |
|---|---|
| INTEGER | целое число (до 10 значащих цифр и знак) |
| SMALLINT | "короткое целое" (до 5 значащих цифр и знак) |
| DECIMAL(p,q) | десятичное число, имеющее p цифр (0 < p < 16) и знак; с помощью q задается число цифр справа от десятичной точки (q < p, если q = 0, оно может быть опущено) |
| FLOAT | вещественное число с 15 значащими цифрами и целочисленным порядком, определяемым типом СУБД |
| CHAR(n) | символьная строка фиксированной длины из n символов (0 < n < 256) |
| VARCHAR(n) | символьная строка переменной длины, не превышающей n символов (n > 0 и разное в разных СУБД, но не меньше 4096) |
| DATE | дата в формате, определяемом специальной командой (по умолчанию mm/dd/yy); поля даты могут содержать только реальные даты, начинающиеся за несколько тысячелетий до н.э. и ограниченные пятым-десятым тысячелетием н.э. |
| TIME | время в формате, определяемом специальной командой, (по умолчанию hh.mm.ss) |
| DATETIME | комбинация даты и времени |
| MONEY | деньги в формате, определяющем символ денежной единицы ($, руб, ...) и его расположение (суффикс или префикс), точность дробной части и условие для показа денежного значения. |

## 3.6 Проектирование базы данных для хранения информации о параметрах движения спутников по орбите

Для решения задачи получения параметров орбит спутников с учетом увеличения объемов и сложности хранимых данных необходимо было выполнить следующие этапы:

1) выбор конкретной модели данных и СУБД для реализации;
2) проектирование логической схемы БД;
3) разработка физической структуры БД;
4) разработка технологии и процедур начального создания и заполнения БД;
5) разработка технологии и процедур сопровождения БД.

Для решения первого этапы была выбрана модель данных SQL. На втором этапе сформированы объект БД, его атрибут и взаимосвязи. Далее (3 этап) была разработана структура БД (см. табл. 3.3) для хранения информации о параметрах движения спутников по орбите. Для создания структуры (4 этап) были созданы выполняемые SQL-запросы (приложение А). Технология заполнения и сопровождения БД ( 5 этап) будет описана в глава 4.

Выводы

В данной главе рассмотрены форматы обмена навигационными данными (RINEX, SP3). Для решения задачи сбора навигационных сообщений выбран формат файлов RINEX (данных наблюдений и навигационных сообщений).

Приведены сведения о системах управления базами данных. В качестве системы управления базами данных выбрана модель данных SQL. Выполнено проектирование БД для хранения информации о параметрах движения навигационных спутников GPS по орбите.

Таблица 3.3 – Структура электронной таблицы GPS

| Наименование поля | Тип данных | Краткое описание поля |
|---|---|---|
| TIME | Время | Время , UTC |
| PRN | Целый | Номер спутника |
| AF0 | Вещественный | Сдвиг часов спутника, с |
| AF1 | Вещественный | Скорость сдвига часов, с/с |
| AF2 | Вещественный | Ускорение сдвига часов, с/($с^2$) |
| IODE | Вещественный | Номер выпуска эфемерид |
| CRS | Вещественный | Коррекция радиус вектора, м |
| DELN | Вещественный | Изменение среднего движения, рад/с |
| EM0 | Вещественный | Средняя аномалия, рад |
| CUC | Вещественный | Коррекция аргумента широты, рад |
| E | Вещественный | Эксцентриситет орбиты |
| CUS | Вещественный | Коррекция аргумента широты, рад |
| SQRTA | Вещественный | Квадратный корень из полуоси орбиты, м |
| T0E | Вещественный | Эпоха эфемерид, с |
| CIC | Вещественный | Коррекция наклона орбиты, рад |
| OMG0 | Вещественный | Долгота восходящего узла в $0^h$ UTC, рад |
| CIS | Вещественный | Коррекция угла наклона орбиты, рад |
| EYE0 | Вещественный | Наклон орбиты к плоскости экватора, рад |
| CRC | Вещественный | Коррекция радиус вектора, м |
| W | Вещественный | Расстояние перицентра от узла, рад |
| OMGDOT | Вещественный | Прецессию восходящего узла, рад/с |
| EYEDOT | Вещественный | Скорость изменения угла наклона орбиты к плоскости экватора, рад/с |
| L2CD | Вещественный | Коды в диапазоне L2 (двоичные, 01=P-код, 10=C/A-код) |
| WKNO | Вещественный | Номер GPS недели |
| L2PD | Вещественный | Флаг данных L2 (0-норма, 1-выключен) |
| SVAC | Вещественный | Точность положения спутника, м |
| SVHL | Вещественный | Исправность спутника |
| TGD | Вещественный | Групповая задержка L1-L2 коррекции, с |
| IODC | Вещественный | Выпуск данных бортового генератора |
| TOT | Вещественный | Время передачи сообщения, с |

# 4 Разработка программного комплекса дистанционного зондирования атмосферы сигналами глобальной навигационной спутниковой системы «GPS Tropodelay»

## 4.1 Выбор операционной и программной среды, определение аппаратных требований

Для решения задач, связанных с разработкой модулей дешифрирования RINEX файлов и выделения необходимых параметров орбиты GPS спутников чтения необходимо выбрать операционную систему, систему программирования и систему хранения информации.

В качестве операционной системы была выбрана *Linux*, как наиболее гибкая и динамичная система с открытым кодом.

Для выбора системы программирования были рассмотрены следующие вопросы:

1) возможности, предоставляемые языком для решения данной задачи;
2) простота компиляции программ;
3) скорость работы программ;
4) Internet/Intranet технологии;
5) свободная лицензия.

В качестве системы программирования выбрана *Java* (компания Sun Microsystems) содержащая:

- универсальный язык программирования Java;
- компилятор;
- виртуальную машину Java's Virtual Machine (JVM).

Компилятор транслирует исходный Java-текст в байт-код, который затем интерпретируется виртуальной машиной [21, 22]. Язык программирования Java является *межплатформенным*, интерпретируемым,

объектно-ориентированным. Он используется для создания различных приложений и характеризуется высоким уровнем интерактивности.

В качестве СУБД была выбрана MySQL (компания Sun Microsystems), как свободная система управления базами данных, которая поддерживается большим количеством типов таблиц. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы. Характеристики СУБД были рассмотрены в третьей главе.

Для сбора информации необходимо следующие аппаратное обеспечение:

- наземная базовая ГНСС станция, которая состоит из антенны и ГНСС приемного устройства.;
- сервер данных;
- источник бесперебойного питания;
- серверный шкаф;
- компьютер пользователя минимальной конфигурации (Pentium II и выше)

Необходимое аппаратное обеспечение имеется в РГГМУ.

## 4.2 Алгоритм вычисления координат НКА GPS на моменты измерения дальности

С целью получения координат НКА GPS на моменты измерения псевдодальности используется алгоритм, представленный в [18, 23, 24].

Определяется поправка для перехода от бортовой шкалы измерения времени к системной шкале (4.1):

$$\Delta t_S = A0 + A1 \cdot \left( Ts - t_{OC} \right) + A2 \cdot \left( Ts - t_{OC} \right)^2 + \Delta t_r, \quad (4.1)$$

где *A0*, *A1*, *A2* – коэффициенты полинома для расчёта поправки для перехода от бортовой шкалы измерения времени к системной шкале (для каждого НКА разные),

*Ts* – время в UTC, на момент которого производится вычисление координат НКА. Для получения координат НКА ежесекундно к *Ts* необходимо на каждом новом шаге прибавлять одну секунду;

Δ*tr* – поправка на релятивистский эффект, вычисляется в пункте 5 настоящего алгоритма.

Получается время излучения метки времени по системной шкале:

$$tk = Ts - \Delta t_S - t_{OC}, \quad (4.2)$$

если tk > 302400 с, то истинное tk = tk – 604800 с, если tk < 302400 с, то истинное tk = tk + 604800.

Вычисляется средняя аномалия НКА на момент излучения метки времени (средняя аномалия эпохи tk):

$$n_0 = \sqrt{\frac{\mu}{a^3}} + \Delta n, \quad (4.3)$$

где $n_0$ – среднее движение НКА или средняя угловая скорость НКА;

μ – гравитационная постоянная для WGS-84;

*a* – – большая полуось орбиты НКА GPS;

Δn – изменение среднего движения НКА, передаётся в кадре навигационного сообщения.

Далее:

$$M = M_0 + n \cdot tk, \tag{4.4}$$

где $M_0$ – средняя аномалия НКА, передаётся в кадре навигационного сообщения.

Средняя аномалия M эпохи tk – это угол между линией апсид и направлением на предполагаемое положение НКА на орбите, в котором он находился бы при равномерном движении.

Затем решается уравнение Кеплера методом итераций для расчёта эксцентрической аномалии НКА:

$$E_0 = M + (e - \frac{e^3}{8}) \cdot \sin(M) + \frac{e^2 \cdot \sin\ (2 \cdot M)}{2} + \frac{3 \cdot e^3}{8} \cdot \sin\ (3 \cdot M), \tag{4.5}$$

где $E_0$ - начальное приближение эксцентрической аномалии,
e - эксцентриситет орбиты НКА.

Здесь

$$E_{j+1} = E_j + \frac{\left(e \cdot \sin\ (E_j) + M - E_j\right)}{\left(1 - e \cdot \sin\ (E_j)\right)}, \tag{4.6}$$

где $j_0$ – номер итерации.

Процесс продолжается до тех пор, пока $| E_{j+1} - E_j| < 1 \cdot 10^{-7}$.

Критерием истинности значения полученной эксцентрической аномалии является вычисленное значение средней аномалии M эпохи tk по полученному значению:

$$M = E - e \cdot \sin(E). \tag{4.7}$$

Полученное значение средней аномалии M эпохи tk должно быть равно значению расчитаному по формуле (4.4).

После этого поправка на релятивистский эффект в уравнение поправки перехода от бортовой шкалы измерения времени к системной шкале:

$$\Delta tr = \left( \frac{-2\sqrt{\mu} \cdot e \cdot \sqrt{a} \cdot \sin(E)}{c^2} \right). \tag{4.8}$$

После вычисления Δtr повторяются все вычисления с самого начала.

Производится расчёт истинной аномалии НКА на момент излучения временной метки по системной шкале:

$$\cos(\upsilon) = \frac{\cos(E) - e}{1 - e \cdot \cos(E)}, \tag{4.9}$$

$$\sin(\upsilon) = \frac{\sqrt{1 - e^2} \cdot \sin(E)}{1 - e \cdot \cos(E)}. \tag{4.10}$$

Далее с учётом знака sin и cos определяется номер четверти угла υ. И с учётом номера четверти для нахождения υ вычисляется либо arcsin (υ), либо arccos (υ).

Вычисляется предварительный аргумент широты:

$$U' = \upsilon + \omega, \tag{4.11}$$

где ω – аргумент перигея, передаётся в кадре навигационного сообщения НКА GPS.

Аргумент перигея – это угол между направлением между направлением восходящего узла орбиты и направлением перигея.

Вычисляется исправленный аргумент широты за счёт гравитационного возмущения орбиты НКА:

1

$$U = U' + Cuc \cdot \cos(2 \cdot U') + Cus \cdot \sin(2 \cdot U'), \tag{4.12}$$

где Cuc, Cus – коэффициенты коррекции аргумента широты, передаваемые в кадре навигационного сообщения НКА GPS.

Вычисляется радиус-вектор НКА GPS на момент излучения метки времени по системной шкале:

$$r = a \cdot \left(1 - e \cdot \cos(E)\right) + Crc \cdot \cos(2 \cdot U) + Crs \cdot \sin(2 \cdot U), \tag{4.13}$$

где Crc, Crs – – коэффициенты коррекции радиус-вектора НКА, передаются в кадре навигационного сообщения НКА GPS.

Вычисляется наклон возмущённой плоскости орбиты к плоскости экватора:

$$i = i_0 + Vi \cdot tk + Cic \cdot \cos(2 \cdot U) + Cis \cdot \sin(2 \cdot U), \tag{4.14}$$

где $i_0$ – наклон орбиты к плоскости экватора, передаваемый в кадре навигационного сообщения НКА GPS;

Vi – скорость изменения наклона орбиты, передаваемая в кадре навигационного сообщения НКА GPS;

Cic, Cis – коэффициенты коррекции угла наклона орбиты НКА, передаваемые в кадре навигационного сообщения НКА GPS.

Вычисляются координаты НКА в плоскости орбиты:

$$X' = r \cdot \cos(U), \tag{4.15}$$

$$Y' = r \cdot \sin(U), \tag{4.16}$$

Вычисляется долгота восходящего узла орбиты НКА:

$$\lambda = \Omega + \left(V\Omega - \Omega e\right) \cdot tk - \Omega e \cdot t_{oc}, \tag{4.17}$$

где Ω – восходящий узел орбиты, передаваемый в кадре навигационного сообщения НКА GPS;

Ωe – угловая скорость вращения Земли;

VΩ – прецессия восходящего узла орбиты НКА, передаваемая в кадре навигационного сообщения НКА GPS.

Вычисляется матрица преобразования геоцентрических координат в геодезические:

$$Mor_g = \begin{pmatrix} \cos(\lambda) & -\sin(\lambda) & 0 \\ \sin(\lambda) & \cos(\lambda) & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(i) & -\sin(i) \\ 0 & \sin(i) & \cos(i) \end{pmatrix} \times \tag{4.18}$$

$$\times \begin{pmatrix} \cos(u) & -\sin(u) & 0 \\ \sin(u) & \cos(u) & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Вычисляются прямоугольные координаты НКА GPS на момент tk без учёта времени распространения сигнала до потребителя:

$$rs = Mor_g \cdot \begin{pmatrix} r \\ 0 \\ 0 \end{pmatrix}, \tag{4.19}$$

где элементы матрицы rs представляют собой соответственно:

$rs_1 = X$, $rs_2 = Y$, $rs_3 = Z$.

Вычисляются прямоугольные координаты НКА GPS на момент tk с учётом времени распространения сигнала от НКА до потребителя (вычисления для НКА ГЛОНАСС аналогичны):

$$\tau = \frac{D}{c}, \tag{4.20}$$

где τ – время распространения сигнала от НКА до потребителя;
D – – измеренная псевдодальность НКА – потребитель.

$$M\tau = \begin{pmatrix} \cos(\Omega e \cdot \tau) & -\sin(\Omega e \cdot \tau) & 0 \\ \sin(\Omega e \cdot \tau) & \cos(\Omega e \cdot \tau) & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{4.21}$$

$$Rs = M\tau \cdot rs, \tag{4.22}$$

где элементы матрицы rs представляют собой соответственно: $Rs_1 = X$, $Rs_2 = Y$, $Rs_3 = Z$ с учётом времени распространения сигнала до потребителя.

Для совместного использования двух СРНС ГЛОНАСС/GPS вычисление координат НКА необходимо производить в одной из общеземных систем координат в ПЗ-90 или WGS-84.

Таким образом, общую схему вычисления положения ИСЗ и обработки информации по получению тропосферной задержки радиосигнала можно представить следующим образом (рис. 4.1):

## 4.3 Организация компонентов программы «GPS Tropodelay»

Программный комплекс дистанционного зондирования состоит из двух управляющих модулей и модулей, которые выполняют служебные функции (рис 4.2).

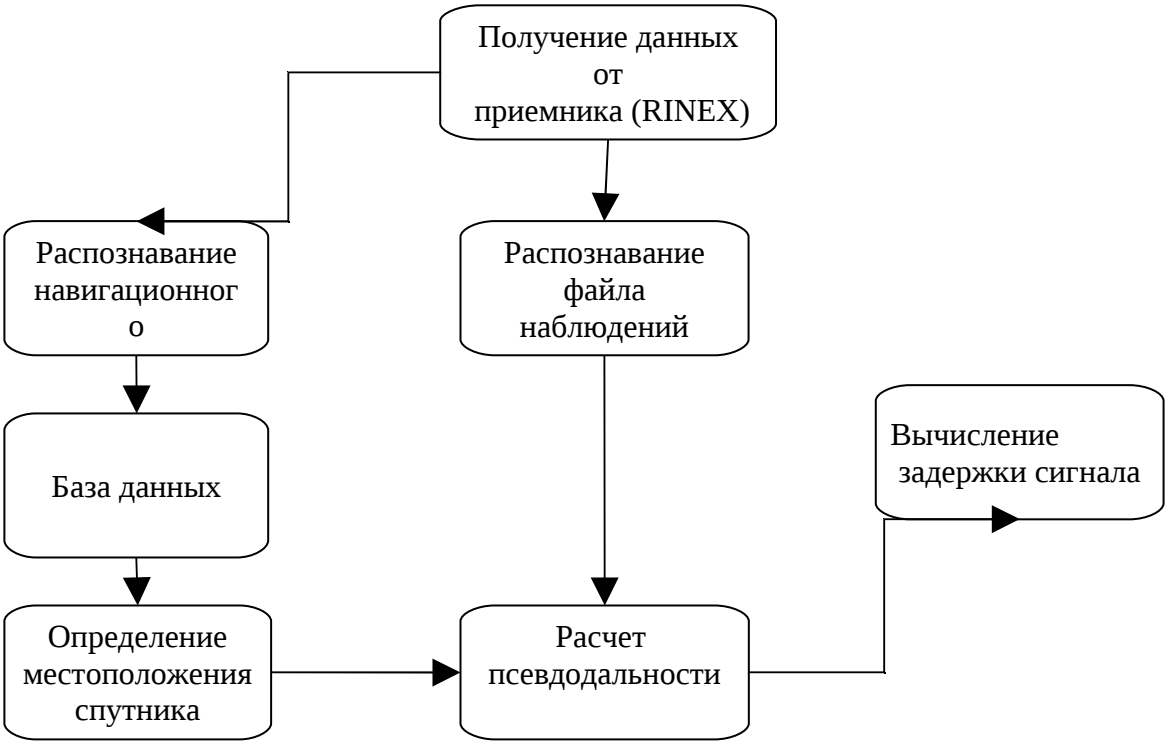



Рисунок 4.1 – Схема получения тропосферной задержки радиосигнала

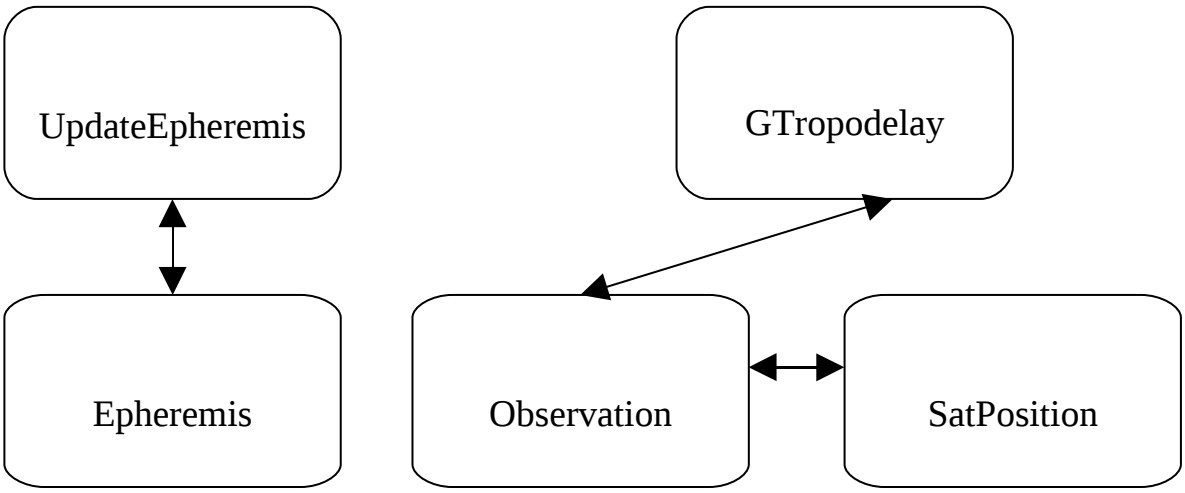



Рисунок 4.2 – Схема проекта программного комплекса

В таблице 4.1 приведен список модулей и приведено назначение программ.

Таблица 4.1 – Список программных модулей

| Имя программы | Назначение |
|---|---|
| Управляющие модули | |
| UpdateEpheremis | Чтение текстовых файлов навигации в формате RINEX. Обработка и запись параметров орбит навигационных спутников в БД. |
| GTropodelay | Чтение текстовых файлов наблюдений в формате данных RINEX.<br>Чтение параметров орбит из БД.<br>Вычисление положения спутника на определенный момент времени.<br>Вычисление азимута и угла места.<br>Расчет тропосферной задержки в атмосфере. |
| Служебные модули | |
| Ephemeris | Распознавание заголовка навигационного файла и блока данных (эфемерид) в формате RINEX с последующей |

| | |
|---|---|
| | записью в массив данных. |
| Observation | Распознавание заголовка файла наблюдений и блока данных (значений дальности до спутников) в формате RINEX.<br>Чтение необходимых записей из базы данных. |
| SatPosition | Перевод даты и времени UTC во время GPS недели. Вычисление координат спутника.<br>Перевод координат места приема из декартовой в географическую систему координат.<br>Вычисление азимута и угла места ИСЗ.<br>Расчет тропосферной задержки. |

Для описания частей программного комплекса использовался структурный принцип – “по элементам и компонентам”.

Элементами являются модули, представленные выше (табл. 4.1), а компонентами – функции и подпрограммы модулей.

*Элемент 1* – управляющая программа UpdateEpheremis. Выполняет функции, указанные в таблице 4.1. Текст программы приведен в приложении Б. Содержит метод main, который создает объект класса Epheremis.

Принцип работы чтения и обработки навигационных файлов указан на схеме (см. рис 4.3).

*Элемент 2* – служебный модуль Epheremis (см. табл. 4.1). Состоит из семнадцати компонент. Ниже перечисленные компоненты являются методами класса Epheremis:

1) компонент setHeaderAndData – выполняет чтение заголовка файла RINEX и вызов методов распознавания заголовка и блока данных;
2) компонент setRinexVersionType - выполняет чтение заголовка файла и запись номера версии и типа файла RINEX.
3) компонент setProgramAgencyDateTime - выполняет чтение заголовка файла и запись даты, организации и название программы создавшей файл.

4) компонент setComment - выполняет чтение заголовка файла и запись комментарий к файлу.

5) компонент setIonAlpha, setIonBeta - выполняют чтение заголовка файла и запись параметров ионосферы.

6) компонент setDeltaUTC - выполняет чтение заголовка файла и запись параметров коррекции времени.

7) компонент setLeapSeconds - выполняет чтение заголовка файла и запись информации о стабильности генератора частоты.

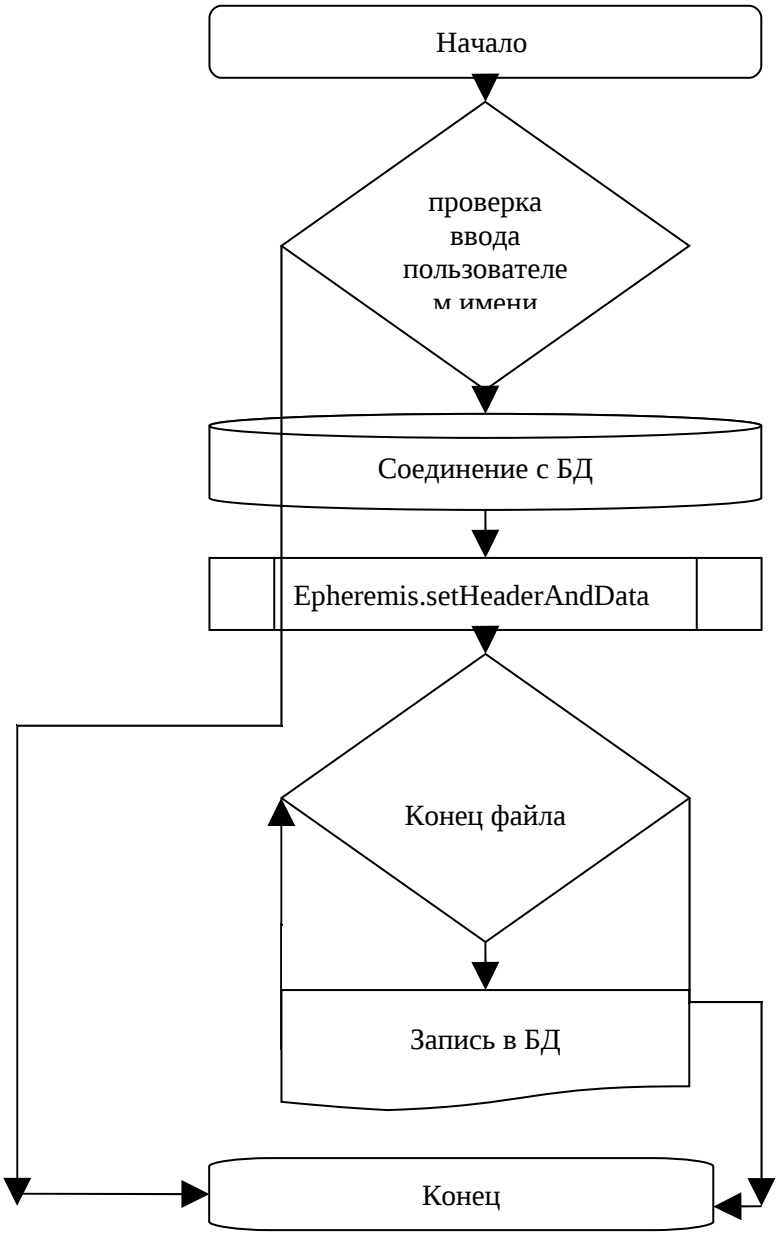


Рисунок 4.3 – Схема алгоритма выполнения UpdateEpheremis

8) компонент setData - выполняет вызов методов чтения времени и блока навигационных данных, запись полученных данных в массив строк qstring.

9) компонент setTime - выполняет чтение даты и времени передаваемых эфемерид.

Компоненты setBroadcastOrbit1 - setBroadcastOrbit7 - выполняют чтение с 1-го по 7-й блоков параметров орбиты спутников. Текст программы приведен в приложении XX.

*Элемент 3* – управляющая программа GTropodelay, осуществляет чтение файлов наблюдений RINEX и передачу управления модулю Observation. Текст программы приведен в приложении XX.

*Элемент 4* – служебный модуль Observation, осуществляет чтение файла наблюдений в формате RINEX, чтение из БД параметров орбиты спутников, вызов методов вычисления положения спутника, определение географических координат места наблюдения и расчет псевдодальности до спутников из модуля SatPosition, а также вычисление тропосферной задержки. Модуль Observation состоит из десяти компонент представленных ниже, которые являются методами класса.

1) Компонент setHeaderAndData – выполняет чтение заголовка файла наблюдений RINEX и вызов методов распознавания заголовка и блока передаваемых данных.
2) Компонент setRinexVersionType - выполняет чтение заголовка файла и запись номера версии и типа файла RINEX.
3) Компонент setProgramAgencyDateTime - выполняет чтение заголовка файла, запись даты, организации, имя наблюдателя и агентства.
4) Компонент setObservationNumber - записывает тип передаваемых данных и код последовательности данных.

5) Компонент setComment - выполняет чтение заголовка файла и запись комментария к файлу.
6) Компонент setData - выполняет вызов методов чтения времени и дальности до спутников.
7) Компонент setEpoch –выполняет чтение даты, времени и количества видимых спутников, для которых передается псевдодальность.
8) Компонент getFrequency - возвращает частоту передачи радиосигнала.
9) Компонент getWaveLength - возвращает длину волны радиосигнала.
10) Компонент setObservations –выполняет:
    1) чтение блока данных расстояний до спутников;
    2) исключение ионосферной задержки
    3) чтение записей параметров орбит из БД;
    4) вызов метода перевода UTC времени во время от начала GPS недели (метод setTime класса SatPosition);
    5) вызов метода определения координат спутника (метод getSatelliteXYZ класса SatPosition);
    6) вызов метода перевода координат места наблюдения в географические координаты (метод setLatLonHeight класса SatPosition);
    7) вызов метода определения азимута и угла места (метод getLonlat класса SatPosition);
    9) вызов метода вычисления псевдодальности (метод psdistance класса SatPosition);
    10) вычисление задержки в тропосфере.

Текст программы приведен в приложении XX.

*Элемент 5* – служебный модуль SatPosition, который осуществляет: перевод UTC времени во время от начала GPS недели; определение координат спутника; перевод координат места наблюдения в географические

координаты; определение азимута и угла места и вычисление псевдодальности до спутника. Модуль SatPosition состоит из семи компонент представленных ниже, которые являются методами класса.

Компонент setTime - осуществляет преобразование времени UTC во время от начала GPS недели.

Компонент getSatelliteXYZ - выполняет вычисление координат спутника, решая уравнение Кеплера.

Компонент kepEQ –решает уравнения Кеплера для эксцентрической аномалии,

Компонент eccano – производит вычисление эксцентрической аномалии в определенный момент времени.

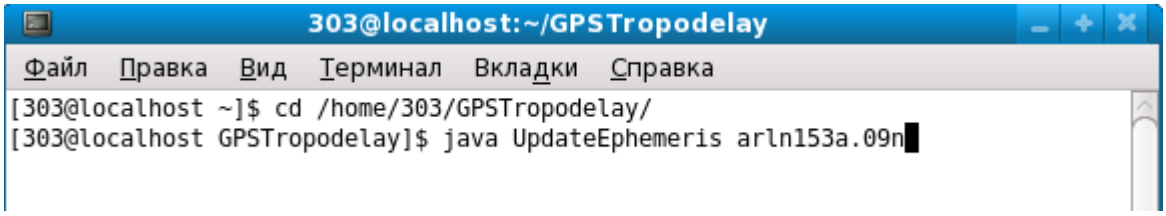
Компонент setLatLonHeight - выполняет перевод координат места наблюдения в географические координаты (широту и долготу).

Компонент getLonlat - выполняет определение азимута и угла места спутника по заданной широте и долготе приемника и координатам спутника.

Компонент psdistance - выполняет вычисления псевдодальности до спутника по известным координатам приемника и вычисленным координатам спутника.

## 4.4 Описание работы

Управляющие модули вызываются и консоли ОС Linux при помощи запуска JVM (см. рис 4.4).



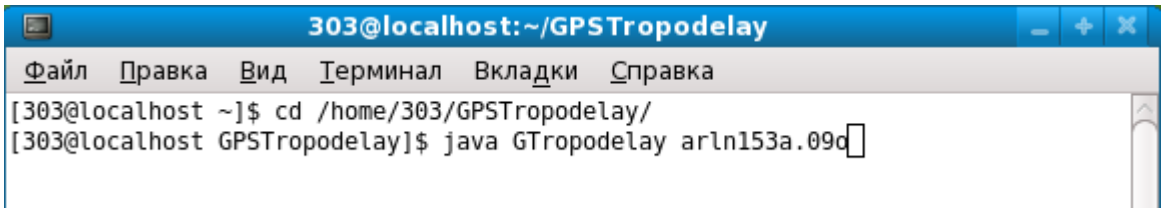



Рисунок 4.4 – Вызов программ из консоли при помощи JVM

Для добавления данных в БД необходимо ввести команду вида:

*java UpdateEphemeris имя_файла_эфемерид*

Для запуска модуля вычисления тропосферной задержки необходимо ввести команду вида:

*java GTropodelay имя_файл_наблюдений*

При запуске приложения UpdateEphemeris происходит проверка ввода пользователем имени файла, при ошибке выдается соответствующее сообщение и приложение закрывается. Если исключительная ситуация не появляется, то происходит соединение с БД, файл навигационного сообщения передается модулю Epheremis, где первым выполняется метод setHeaderAndData. В модуле Epheremis при вызове метода setHeaderAndData происходит построчное чтение заголовка файла навигационного сообщения. Если в строке будет найдена подстрока END OF HEADER это будет означать что заголовок закончился, и далее будет вызван метод setData, с помощью этого метода происходит чтение данных времени спутника и передаваемых параметров орбиты для каждого спутника. Эта процедура завершается записью полученных значений в массив строк в виде SQL запроса, и передаче управления программе UpdateEphemeris. Если в процессе считывания данных произойдет сбой, программа завершит свою работу и сообщит об ошибке. Далее программа UpdateEphemeris выполняет все SQL запросы и производит запись полученных навигационных данных в БД. В таблице 4.1 приведен список функций модуля Epheremis.

При запуске приложения GTropodelay происходит проверка ввода пользователем имени файла, при ошибке выдается соответствующее сообщение и приложение закрывается. Если исключительная ситуация не появляется, то происходит вызов метода setHeaderAndData из модуля Observation. С помощью метода setHeaderAndData происходит чтение заголовка файла наблюдений до того пока не встретится подстрока END OF HEADER, означающая конец заголовка и ничало части с данными наблюдений. Запускается функция setData, который будет выполнять 2 метода:

1) setEpoch – метод выполняет чтение даты и времени и количества видимых спутников, для которых передается псевдодальность;
2) setObservations – метод выполняет чтение блока данных, а также вызывает функции вычисления псевдодальности, определения азимута и угла места и производит расчет задержки в ионосфере и тропосфере.

В методе setObservations после считывания блока данных наблюдений для определенного спутника происходит вычисление поправки на прохождение радиосигнала на 2-х частотах, и удаление из псевдодальности до спутника этой составляющей. Далее совершается соединение с БД при помощи SQL запроса, который возвращает параметры орбиты движения спутника. При помощи функции setTime, расположенной в модуле SatPosition происходит перевод времени, полученного при помощи метода setEpoch в секунды  от начала GPS недели. В setTime передается значение года, месяца, дня, часа, минуты и секунды полученной информации о дальности до спутника. Значение GPS времени передается в виде параметра в функцию getSatelliteXYZ, расположенную в модуле SatPosition, функция решает уравнение Кеплера и определяет положение спутника в декартовой системе координат. В модуле Observation происходит вызов методов setLatLonHeight и getLonlat находящихся в модуле SatPosition. С помощью этих методов

происходит перевод декартовых координат места обсервации в широту и долготу, и последующее вычисление азимута и угла подъема спутника над горизонтом. Последним из класса SatPosition вызывается функция psdistance, которая рассчитывает псевдодальность до ИСЗ на основе координат полученных из фунции getSatelliteXYZ. Далее в модуле Observation производится расчет задержки радиосигнала в тропосфере и вывод результатов на консоль.

В качестве контрольной информации для проверки надежности программы были использованы файлы в формате RINEX, полученные от Департамента транспорта штата Орегон, сети базовых станций Oregon Department of Transportation – GPS Network:

- навигационное сообщение arln153a.09n;
- файл наблюдений arln153a.09o.

Дата наблюдений – 02.06.2009, станция – Арлингтон (США), период измерений – 1 час.

По полученным данным построен график зависимости тросферной задержки от угла места, представленный на рисунке 4.5.

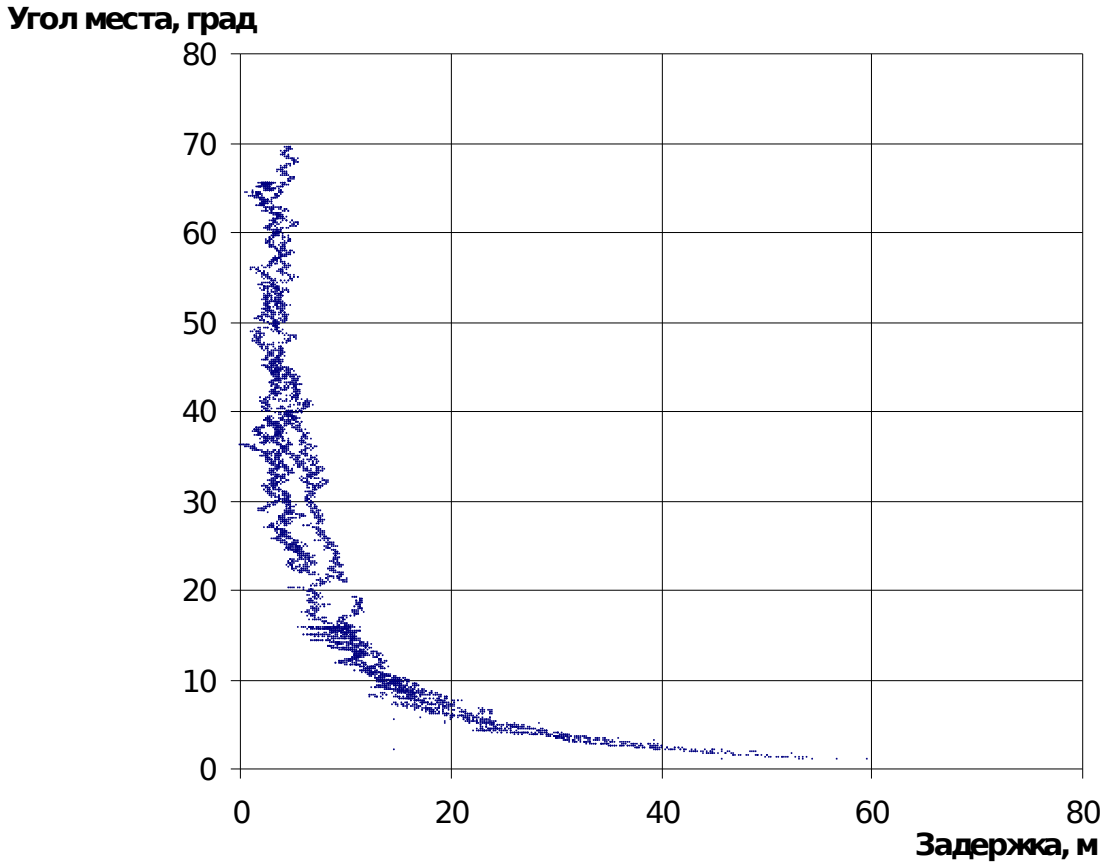



Рисунок 4.5 – График зависимости тропосферной задержки радиосигнала от угла места

Из рисунка видно, что тропосферная задержка в атмосфере увеличивается с уменьшением угла места.

Результаты апробирования показали надежность работоспособности системы.

Фрагмент варианта расчета приведен в приложении В.

Ошибка определения задержки радиосигнала в тропосфере может быть оценена по формуле:

$$\sigma^2 = \sigma_1^2 + \sigma_2^2 + \sigma_3^2, \tag{4.23}$$

где: σ – среднее квадратическое отклонение ошибок определения тропосферной задержки, м;

$\sigma_1$ – среднее квадратическое отклонение ошибок определения

местоположения спутника, м;

$\sigma_2$ – среднее квадратическое отклонение ошибок определения местоположения приемной антенны;

$\sigma_3$ – среднее квадратическое отклонение ошибок измерения псевдодальности, м.

При значениях $\sigma_1$=0.26 м [18] (при использовании оперативных значений эфемерид), $\sigma_2$=0.01 м (при стационарном размещении приемной антенны), $\sigma_3$=1 м (при регистрации сигнала более 4 с), ошибка определения тропосферных задержек составляет 1.1 м. В тоже время из анализа рисунка 4.5 видно, что среднеквадратическое отколонение составляет в среднем 1.5 метра, что может быть объяснено дополнительным вкладом неизвестного нам фактора.

Выводы

В процессе разработки программного комплекса дистанционного зондирования атмосферы сигналами глобальной навигационной спутниковой системы «GPS Tropodelay» были учтены требования, предъявляемые к обработке навигационных данных для целей расчета тропосферной задержки. Для реализации разработанных алгоритмов был выбран язык программирования и среда, позволяющие реализовывать необходимый пользовательский интерфейс. В качестве операционной системы была выбрана *Linux*, СУБД - MySQL, система программирования – *Java*. В ходе работы над дипломным проектом были разработаны алгоритмы вычисления положения ИСЗ и обработки информации по получению тропосферной задержки радиосигнала. В качестве реализации алгоритма разработан программный комплекс, который получает навигационный сигнал глобальной навигационной спутниковой системы Navstar GPS с занесением в базу данных на сервере РГГМУ и проводит расчет тропосферной задержки, проведены контрольные расчеты.

# Заключение

На сегодняшний день разработка и внедрение информационных систем является одной из самых актуальных, интересных и важных задач в области информационных технологий. Появилась потребность получать гидрометеорологическую информацию со станций подсистем спутниковых навигационных систем. С помощью метода дистанционного зондирования атмосферы, основанного на электромагнитном просвечивании можно получить информацию о содержании водяного пара, которая в дальнейшем будет использоваться для целей анализа и прогноза погоды.

В дипломном проекте выполнена разработка программного комплекса дистанционного зондирования атмосферы сигналами глобальной навигационной спутниковой системы «GPS Tropodelay».

В процессе работы изучено большое количество научной литературы и информационных источников о навигационных спутниковых системах, методах положения ИСЗ в пространстве, форматах навигационной информации, методиках расчета тропосферной задержки спутникового навигационного сигнала. Для реализации проекта определялась информационная среда, в которой проводилась подготовка программного комплекса. Получены навыки работы в операционной среде Linux, системе управления базами данных MySQL, был изучен язык программирования Java, на котором проводилось программирование объектов.

Программный комплекс является частью информационной системы, которая будет использоваться для сбора информации о содержании водяного пара с целью анализа и прогноза погоды, так же может использоваться для повышения точности определения координат пользователей дифференциальной подсистемы ГНСС за счет более точного определения тропосферных задержек навигационных радиосигналов.

Практическое внедрение данного проекта позволит проводить расчет задержки радиосигнала, что будет использовано системой оперативного мониторинга атмосферы с высоким временным разрешением. Таким образом, разработанный программный комплекс может использоваться для повышения точности прогнозов погоды.

Список использованных источников


1 Чукин, В.В. Исследование атмосферы методом электромагнитного просвечивания. [Текст] / В.В.Чукин.– СПб.: РГГМУ, 2004. – 107 с.

2 Соловьев, Ю.А. Системы спутниковой навигации. [Текст] / Ю.А. Соловьев.– М.: Эко-Трендз, 2000. – 270 с.

3 Глобальная спутниковая радионавигационная система ГЛОНАСС [Текст] / Под. ред. В.Н. Харисова, А.И. Перова, В.А. Болдина. – М.: ИПРЖР, 1998. – 400 с.

4 Поваляев, Е.А. Системы спутниковой навигации ГЛОНАСС и GPS. / Е.А.Поваляев, С.В. Хуторной // Chip News [Электронный ресурс] . – Электронный журнал. М.: Chip News.–Режим доступа к журналу: http://www.chipinfo.ru/literature/chipnews/200110/9.html

5 Козин, И.Д. Спутниковые радионавигационные системы. [Текст] / И.Д. Козин. – Алматы , 2006. – 75 с.

6 Афрамович, Э.Л. GPS-мониторинг верхней атмосферы Земли [Текст] / Э.Л. Афрамович, Н.П. Перевалова.– Иркутск: ГУ НЦ РВХ ВСНЦ СО РАМН, 2006. – 480 с.

7 Smith, E. K. The constants in the equation for atmospheric refractive index at radio frequencies [Text] / Smith E. K., Weintraub S. // Proceedings of Proc. IRE. – 1953. – Vol.41. – P.1035–1037.

8 Boudouris, G. On the index of refraction of air, the absorption and dispersion of centimeter waves by gases [Text] / Boudouris G. // Journal of Research of the National Bureau of Standards 67D(6). – 1963. – P.631-684.

9 Thayer, G. D. An improved equation for the radio refractive index of air [Text] / Thayer G. D. // Radio Science. – Vol.9(10). – 1974. – P.803–807.

10 Davis, J. L. Geodesy by radio interferometry: Effects of atmospheric modeling errors on estimates of baseline length [Text] / Davis J. L., Herring T. A., Sharipo I. I., Rogers A. E. E., Elgered, G. // Radio Sci.. – 1985. – Vol.20(6). – P.593-1607.

11 Ifadis, I. The Atmospheric Delay of Radio Waves: Modeling the Elevation Dependence on a Global Scale [Text] / Ifadis I. // Technical Report no. 38L, School of Electrical and Computer Engineering, Chalmers University of Technology, Göteborg. – Sweden, 1986.

12 Bevis, M. GPS meteorology: Remote sensing of atmospheric water vapor using the Global Positioning system [Text] / Bevis M., Businger S., Herring T. A., Rocken C., Anthes R. A., Ware R. H. // J. Geophys. Res. – 1992. – Vol.97. – P.15787–15801.

13 Герман, М.А. Космические методы исследования в атмосфере. [Текст] / М.А.Герман, . – Л.: Гидрометеоиздат, 1985. – 352 с.

14 Юркина, М.И. Действующие системы координат в России. / М.И. Юркина, Л.И. Серебрякова // Credo-dialogue [Электронный ресурс] . – Электронный журнал./ ЦНИИГАиК. – М.: Credo-dialogue, 2006 . – Режим доступа к ресурсу: ftp://ftp.credo-dialogue.com/publication/ /geodesy/a2_0_20.pdf

15 Насыров, И. А. Введение в современные спутниковые радионавигационные системы. [Текст] / И. А. Насыров.– Казань: КГУ, 2005. – 43 с.

16 Переход от одной системы координат к другой – методы трансформации / [Электронный ресурс] Режим доступа к ресурсу: http://gis-lab.info/qa/datum-transform-methods.html

17 Глобальная навигационная спутниковая система. ГЛОНАСС. Интерфейсный контрольный документ, Редакция 5 / [Электронный ресурс]. – М, 2002. . – Режим доступа к документу: http://www.glonass-ianc.rsa.ru/i/glonass/ICD-2002r.pdf

18 Гурин, С. Е. Спутниковые радионавигационные системы ГЛОНАСС / GPS на железнодорожном транспорте. [Текст] /. С. Е. Гурин – М.: МИИТ, 2004. – 55 с.

19 Gurtner, W. RINEX: The Receiver Independent Exchange Format Version 2.10 / [Электронный ресурс]. 2002. Режим доступа к документу -ftp://igscb.jpl.nasa.gov/igscb/data/format/rinex210.txt

20 Рубен, А. Эффективная работа с СУБД [Текст] / А. Рубен, А. Горев, С. Макшарипов. – СПб.: Питер, 2001. – 822 с.

21 Флэнаган, Д. Java в примерах. Справочник 2-е издание [Текст] / Д. Флэнаган. – СПб.: Символ-Плюс.2003. – 664 с. – Пер. с англ.

22 Хорстманн, К. Библиотека профессионала. Java 2. Том 2. Тонкости программирования [Текст] / К. Хорстманн, Г. Корнелл. – М.: Издательский дом «Вильямс», 2002. – 1120с. – Пер. с англ.

23 Interface control document. Navstar GPS Space Segment/Navigation User Interfaces. (ICD-GPS-200c) / [Электронный ресурс]. – 2000. – Режим доступа к документу: http://www.navcen.uscg.gov/pubs/gps/icd200/ icd200cw1234.pdf

24 Xu, G. GPS – Theory, Algorithms and Applications. [Text] / G Xu . − Berlin: Springer – Verlag, 2003.

# ПРИЛОЖЕНИЕ А

## Файл исполняемого SQL запроса

```
drop database if exists ephemeris;
create database ephemeris;
use ephemeris;

# Альманах системы GPS

create table gps(
time                datetime,        # время, UTC
prn                 int unsigned,    # номер спутника
af0                 double,          # сдвиг часов спутника, с
af1                 double,          # скорость сдвига часов, с/с
af2                 double,          # ускорение сдвига часов, с/(с*с)
                                     # Раздел 1
iode                double,          #
crs                 double,          # Crc (m)
deln                double,          # Delta n (rad/s)
em0                 double,          # M0 (rad)
                                     # Раздел 2
cuc                 double,          # Cuc (rad)
e                   double,          # экстриситет
cus                 double,          # Cus (rad)
sqrta               double,          # SQRT A полуоси(m^0.5)
                                     # Раздел 3
t0e                 double,          # Время эфемерид (Toe) (секунда
                                     от начала GPS-недели)|
cic                 double,          # Cic (rad)
omg0                double,          # OMEGA наклонение или (rad)
cis                 double,          # CIS (rad)
                                     # Раздел 4
eye0                double,          # i0 (rad)
crc                 double,          # Crc (m)
w                   double,          # Omega (rad)
omgdot              double,          # OMEGA DOT (rad/c)
                                     # Раздел 5
eyedot              double,          # IDOT (rad/s)
l2cd                double,          # Коды в диапазоне L2
wkno                double,          # Номер GPS-недели (используется
                                     с TOE), непрерывная числовая
                                     последовательность
l2pd                double,          # Флаг данных L2 P
                                     # Раздел 6
svac                double,          # Точность положения спутника
                                     (метр)
svhl                double,          # Исправность сп. (биты 17-22
                                     сл.3 кад.1)
tgd                 double,          # TGD (секунда)
iodc                double,          # IODC
                                     # Раздел 7
tot                 double           # время передачи сообщения
) type=MyISAM;
```

# ПРИЛОЖЕНИЕ Б
# Тексты программ модулей «GPS Tropodelay»

## Текст программы UpdateEphemeris.java

```
/**
  * Обновление данных эфемерид навигационных спутниковых систем.
  *@author Григорьев Г.В., Чукин В.В.
  *Российский государственный гидрометеорологический университет (РГГМУ)
  * Лицензия: GPL v.2
  */
import java.util.*;
import java.io.*;
import java.net.*;
import java.sql.*;
import static java.lang.Math.*;
class UpdateEphemeris {
static String dataBaseUrl = "jdbc:mysql://127.0.0.1/ephemeris";
static String dataBaseUser = "root";
static String dataBasePassword = "";

public static void main(String[] args) {
  //Начало создание запроса
  String query  = null;
  String query1 = null;
  int number = -1;
  int i =0;
  String fileName=args[0];
  BufferedReader in;
   // Проверяем, указан ли файл с данными альманаха
   if(args.length>0) {
    fileName = args[0];
    try {
     FileReader fr = new FileReader(fileName);
     in = new BufferedReader(fr);
     }
    catch(Exception e) { System.out.println("Не отрыть файл "+fileName);
    return; }
    }
   else {
    try {
     InputStreamReader br = new InputStreamReader(System.in);
     in = new BufferedReader(br);
     }
    catch(Exception e) {
     System.out.println("Вызов:");
     System.out.println(" java UpdateEphemeris имя_файла_данных");
     return;
     }  }
   // Проверяем наличие драйвера для доступа к MySQL
   try {
    Class.forName("com.mysql.jdbc.Driver");
   }
   catch(Exception ex) {
    ex.printStackTrace();
    return;
   }
   try {
            Connection   c   =   DriverManager.getConnection(dataBaseUrl,
dataBaseUser, dataBasePassword);
```

```
            Statement s = c.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
           Ephemeris ephemeris = new Ephemeris();
           BufferedReader in2 = new BufferedReader(new FileReader(fileName));
           ephemeris.setHeaderAndData(in2);
          while(ephemeris.qstring[i] != null) {
          query=ephemeris.qstring[i];
          s.executeUpdate(query);
          i++;
      }
      System.out.println("Добавлено    "+i+"    записей    в    Базу    данных
Ephemeris/gps");
           s.close();
           c.close();
        }
         catch(SQLException ex) { ex.printStackTrace(); return; }
         catch(Exception e) { e.printStackTrace(); return; }
       }
      }
```

## Текст программы Ephemeris.java

```
      /**
        * Парсер файлов навигационных сообщений GPS в формате RINEX.
        *@author Григорьев Г.В., Чукин В.В.
        *Российский государственный гидрометеорологический университет (РГГМУ)
        * Лицензия: GPL v.2
        */

import java.util.*;
import java.io.*;
import static java.lang.Math.*;
public class Ephemeris {
   private static final double gm    =  3.986005E+14;
   private static final double we    =  7.2921151467E-5;        // Угловая
скорость вращения Земли, рад/с
   private static final double gpspi =  3.1415926535898E+0;      // Число Пи,
принятое в системе GPS, рад
   private static final double bigf  = -4.442807633E-10;

   public String rinexVersion;
   public String fileType;
   public String gnssType;
   public String programName;
   public String programAgency;
   public String programDate;
   public String comment;
   public double A0;
   public double A1;
   public double A2;
   public double A3;
   public double B0;
   public double B1;
   public double B2;
   public double B3;
   public int leapSeconds;
   public int year;
   public int month;
   public int day;
   public int hour;
   public int minute;
```

```
   public double sec;
   public double deltaN;
   public double m0;
   public double OMEGA;
   public double i0;
   public double omega;
   public double omegaDot;
   public double idot;
   public double bandL2Codes;
   public double gpsWeek;
   public double bandL2DataFlag;
   public double svAccuracy;
   public double svHealth;
   public double transmissionTime;
   public double fitInterval;
   public int prn;        // номер спутника PRN
   // Параметры часов
   public double t0c;     // эпоха для коррекции часов, с или опорное время
параметров часов
   public double af0;     // сдвиг часов, с
   public double af1;     // скорость сдвига часов, с/с
   public double af2;     // ускорение сдвига часов, с/(с*с)
   public double a0;
   public double a1;
   public int T;
   public int W;
   // Кеплеровские элементы
   public double t0e;     // эпоха эфемерид, секунды с начала GPS-недели
----опорное время параметров эфемерид
   public double sqrta;   //*  square root of semimajor axis           sqrt(m)
   public double e;       //*  eccentricity
   public double eye0;    //*  inclination at ref. epoch                  (r)
   public double omg0;    //*  right ascension of asc. node at ref. epoch (r)
   public double w;       //*  argument of perigee                        (r)
   public double em0;     //*  mean anomaly at ref. epoch                 (r)
   // Параметры возмущения
   public double deln;    //*  mean motion difference from computed value
   public double eyedot;  //*  rate of change of inclination
   public double omgdot;  //*  rate of change of right ascension      (r/s)
   public double cuc;     //*  cosine harmonic correction to arg. of lat. (r)
   public double cus;     //*  sine harmonic correction to arg. of lat.   (r)
   public double crc;     //*  cosine harmonic correction to radius       (m)
   public double crs;     //*  sine harmonic correction to radius         (m)
   public double cic;     //*  cosine harmonic correction to inclination  (r)
   public double cis;     //*  sine harmonic correction to inclination    (r)
   // другие данные
   public double iode;    //*  issue of data ephemeris
   public double l2cd;    //*  codes on l2 channel двоичные
   public double wkno;    // номер GPS-недели (to go with t0e)           (wk)
   public double l2pd;    // l2 p data flag (binary, 0-norm, 1-set off)
   public double svac;    // URA index (converted)                       (m)
   public double svhl;    // состояние спутника (6 бит, 0-исправен, 1-нет)
   public double tgd;     // est. group delay   (tsv-tgd)                (s)
   public double iodc;    // issue of data clock
   public double tot;     // время передачи сообщения   (sow)
   String[] qstring = new String[100];
   public int q;
   // Конструктор класса
   public Ephemeris() {
    rinexVersion = null;
    fileType = null;
```

```
     gnssType = null;
     programName = null;
     programAgency = null;
     programDate = null;
     comment = "";
     A0 = 0;
     A1 = 0;
     A2 = 0;
     A3 = 0;
     B0 = 0;
     B1 = 0;
     B2 = 0;
     B3 = 0;
     a0 = 0;
     a1 = 0;
     q = 0;
    }
    /** Чтение заголовка и навигационных данных */
    public void setHeaderAndData(BufferedReader in) {
     String line;
     String label;
     try {
      while((line = in.readLine()) != null) {
        label = line.substring(60, 80).trim();
        if(label.equals("RINEX VERSION / TYPE")) setRinexVersionType(line);
        if(label.equals("PGM / RUN BY / DATE"))
setProgramAgencyDateTime(line);
        if(label.equals("COMMENT"))              setComment(line);
        if(label.equals("ION ALPHA"))            setIonAlpha(line);
        if(label.equals("ION BETA"))             setIonBeta(line);
        if(label.equals("DELTA-UTC: A0,A1,T,W")) setDeltaUTC(line);
        if(label.equals("LEAP SECONDS"))         setLeapSeconds(line);
        if(label.equals("END OF HEADER"))        setData(in);
        deltaN=deln;
        transmissionTime=tot;
       }
      } catch(Exception e) { return; }
     }
    /** RINEX VERSION / TYPE */
    private void setRinexVersionType(String line) {
       rinexVersion  = line.substring( 0, 20).trim();
       fileType      = line.substring(20, 40).trim();
       gnssType      = line.substring(40, 60).trim();
    }
    /** PGM / RUN BY / DATE */
    private void setProgramAgencyDateTime(String line) {
       programName   = line.substring(0, 20).trim();
       programAgency = line.substring(20, 40).trim();
       programDate   = line.substring(40, 60).trim();
    }
    /** COMMENT */
    private void setComment(String line) {
       comment = comment + line.substring(0, 60).trim() +"\n";
    }
    /** ION ALPHA */
    private void setIonAlpha(String line) {
     A0 = Double.parseDouble( line.substring(2, 14).trim().replace('D','E') );
    A1 = Double.parseDouble( line.substring(14, 26).trim().replace('D','E') );
    A2 = Double.parseDouble( line.substring(26, 38).trim().replace('D','E') );
    A3 = Double.parseDouble( line.substring(38, 50).trim().replace('D','E') );
    }
```

```
   /** ION BETA */
   private void setIonBeta(String line) {
   B0 = Double.parseDouble( line.substring(2, 14).trim().replace('D','E') );
   B1 = Double.parseDouble( line.substring(14, 26).trim().replace('D','E') );
   B2 = Double.parseDouble( line.substring(26, 38).trim().replace('D','E') );
   B3 = Double.parseDouble( line.substring(38, 50).trim().replace('D','E') );
   }
   /** DELTA-UTC: A0,A1,T,W */
   private void setDeltaUTC(String line) {
   a0 = Double.parseDouble( line.substring( 3, 22).trim().replace('D','E') );
   a1 = Double.parseDouble( line.substring(22, 41).trim().replace('D','E') );
   T  = Integer.parseInt(   line.substring(41, 50).trim() );
   W  = Integer.parseInt(   line.substring(50, 59).trim() );
   }
   /** LEAP SECONDS */
   private void setLeapSeconds(String line) {
     leapSeconds = Integer.parseInt( line.substring(0, 6).trim() );
    }
   /** Чтение навигационных данных */
   public void setData(BufferedReader in) {
    String line;
    try {
     while((line = in.readLine()) != null) {
       setTime(line);
       line = in.readLine(); setBroadcastOrbit1(line);
       line = in.readLine(); setBroadcastOrbit2(line);
       line = in.readLine(); setBroadcastOrbit3(line);
       line = in.readLine(); setBroadcastOrbit4(line);
       line = in.readLine(); setBroadcastOrbit5(line);
       line = in.readLine(); setBroadcastOrbit6(line);
       line = in.readLine(); setBroadcastOrbit7(line);
qstring[q] = "insert into gps values(\""+year+"-"+month+"-"+day+"
"+hour+":"+minute+":"+sec+"\", "+prn+", "+af0+", "+af1+", "+af2+", "+iode+",
"+crs+", "+deln+", "+em0+", "+cuc+", "+e+", "+cus+", "+sqrta+", "+t0e+",
"+cic+", "+omg0+", "+cis+", "+eye0+", "+crc+", "+w+", "+omgdot+", "+eyedot+",
"+l2cd+", "+wkno+", "+l2pd+", "+svac+", "+svhl+", "+tgd+", "+iodc+",
"+tot+");";
 q++;
 }
     } catch(Exception e) { System.out.printf(" Не прочитать блок данных\n");
return; }
    }
   /** SV / EPOCH / SV CLK */
   public void setTime(String line) {
      prn        = Integer.parseInt(   line.substring( 0,  2).trim() );
      year       = Integer.parseInt(   line.substring( 2,  5).trim() );
      if(year<80) year = 2000 + year;
        else year = 1900 + year;
month      = Integer.parseInt(   line.substring( 5,  8).trim() );
day        = Integer.parseInt(   line.substring( 8, 11).trim() );
hour       = Integer.parseInt(   line.substring(11, 14).trim() );
minute     = Integer.parseInt(   line.substring(14, 17).trim() );
sec        = Double.parseDouble( line.substring(17, 22).trim() );
af0 = Double.parseDouble( line.substring(22,41).trim().replace('D','E') );
af1 = Double.parseDouble( line.substring(41,60).trim().replace('D','E') );
af2 = Double.parseDouble( line.substring(60, 79).trim().replace('D','E') );
   }
   /** BROADCAST ORBIT - 1 */
   public void setBroadcastOrbit1(String line) {
iode   = Double.parseDouble( line.substring( 3, 22).trim().replace('D','E') );
crs    = Double.parseDouble( line.substring(22, 41).trim().replace('D','E') );
```

```
deln   = Double.parseDouble( line.substring(41, 60).trim().replace('D','E') );
em0    = Double.parseDouble( line.substring(60, 79).trim().replace('D','E') );
   }
   /** BROADCAST ORBIT - 2 */
   public void setBroadcastOrbit2(String line) {
cuc   = Double.parseDouble( line.substring( 3, 22).trim().replace('D','E') );
e     = Double.parseDouble( line.substring(22, 41).trim().replace('D','E') );
cus   = Double.parseDouble( line.substring(41, 60).trim().replace('D','E') );
sqrta = Double.parseDouble( line.substring(60, 79).trim().replace('D','E') );
   }
   /** BROADCAST ORBIT - 3 */
   public void setBroadcastOrbit3(String line) {
t0e   = Double.parseDouble( line.substring( 3, 22).trim().replace('D','E') );
cic   = Double.parseDouble( line.substring(22, 41).trim().replace('D','E') );
omg0  = Double.parseDouble( line.substring(41, 60).trim().replace('D','E') );
cis   = Double.parseDouble( line.substring(60, 79).trim().replace('D','E') );
   }
   /** BROADCAST ORBIT - 4 */
   public void setBroadcastOrbit4(String line) {
eye0   = Double.parseDouble( line.substring( 3, 22).trim().replace('D','E') );
crc    = Double.parseDouble( line.substring(22, 41).trim().replace('D','E') );
w    = Double.parseDouble( line.substring(41, 60).trim().replace('D','E') );
omgdot = Double.parseDouble( line.substring(60, 79).trim().replace('D','E') );
   }
   /** BROADCAST ORBIT - 5 */
   public void setBroadcastOrbit5(String line) {
eyedot = Double.parseDouble( line.substring( 3, 22).trim().replace('D','E') );
l2cd  = Double.parseDouble( line.substring(22, 41).trim().replace('D','E') );
wkno   = Double.parseDouble( line.substring(41, 60).trim().replace('D','E') );
l2pd   = Double.parseDouble( line.substring(60, 79).trim().replace('D','E') );
     }
   /** BROADCAST ORBIT - 6 */
   public void setBroadcastOrbit6(String line) {
svac  = Double.parseDouble( line.substring( 3, 22).trim().replace('D','E') );
svhl  = Double.parseDouble( line.substring(22, 41).trim().replace('D','E') );
tgd   = Double.parseDouble( line.substring(41, 60).trim().replace('D','E') );
iodc  = Double.parseDouble( line.substring(60, 79).trim().replace('D','E') );
     }
   /** BROADCAST ORBIT - 7 */
   public void setBroadcastOrbit7(String line) {
tot  = Double.parseDouble( line.substring( 3, 22).replace('D','E').trim() );
fitInterval = Double.parseDouble( line.substring(22,
41).replace('D','E').trim() );
     } }
```

## Текст программы GTropodelay.java

```
/**
  * Чтение файлов данных наблюдений в формате RINEX
  *@author Григорьев Г.В.
  *Российский государственный гидрометеорологический университет (РГГМУ)
  * Лицензия: GPL v.2
  */
import java.util.*;
import java.io.*;
import java.net.*;
import java.sql.*;
import static java.lang.Math.*;
class GTropodelay {
 static double kepE;
 static final double gpspi =  3.1415926535898E+0;    // Число Пи для GPS, рад
```

```
 public static void main(String[] args) {
//  Чтение Файла навигационных данных
  String fileName=args[0];
  BufferedReader in;
   if(args.length>0) {
    fileName = args[0];
    try {
     FileReader fr = new FileReader(fileName);
     in = new BufferedReader(fr);
     }
  catch(Exception e) { System.out.println("Не отрыть файл "+fileName);
 return; }
    }
   else {
    try {
     InputStreamReader br = new InputStreamReader(System.in);
     in = new BufferedReader(br);
     }
    catch(Exception e) {
     System.out.println("Вызов:");
     System.out.println(" java GTropodelay имя_файла_данных");
     return;
     } }
   try {
     Observation ob = new Observation();
     ob.setHeaderAndData(in);
    }
   catch(Exception e) { System.out.println("Не отрыть файл"); return; }
} }
```

## Текст программы Observation.java

```
      /**
* Парсер файлов данных наблюдений в формате RINEX, соединение с БД.
*@author Григорьев Г.В., Чукин В.В.
*Российский государственный гидрометеорологический университет (РГГМУ)
* Лицензия: GPL v.2
*/
   import java.util.*;
   import java.io.*;
   import java.net.*;
   import java.sql.*;
   import java.lang.System.*;
   import java.util.Locale.*;
   import static java.lang.Math.*;
   public class Observation {
   public String rinexVersion;
   public String fileType;
   public String gnssType;
   public String programName;
   public String programAgency;
   public String programDate;
   public String markerName;
   public String markerNumber;
   public String observer;
   public String agency;
   public String receiverNumber;
   public String receiverType;
   public String receiverVersion;
   public String antennaNumber;
   public String antennaType;
```

```
  public String comment;
  public double A0;
  public double A1;
  public double T;
  public double W;
  public int year;
  public int month;
  public int day;
  public int hour;
  public int minute;
  public double sec;
  public int flag;
  public int satelliteQuantity;
  public String[] satelliteSystem;
  public int[] satelliteNumber;
  public int observationQuantity;
  public String[] observationType;
  public int[] observationBand;
  public double[][] signalData;
  public int[][] signalLLI;
  public int[][] signalStrength;
  public double[] TEC;
  public double L;
  public double psL;
  static String dataBaseUrl = "jdbc:mysql://127.0.0.1/ephemeris";
  static String dataBaseUser = "root";
  static String dataBasePassword = "";
  String query  = null;
  // Конструктор класса

 public Observation() {
     rinexVersion = null;
     fileType = null;
     gnssType = null;
     programName = null;
     programAgency = null;
     programDate = null;
     comment = "";
     satelliteQuantity = 0;
     satelliteSystem = new String[1000];
     satelliteNumber = new int[1000];
     observationQuantity = 0;
     observationType = new String[30];
     observationBand = new int[30];
     signalData = new double[1000][30];
     signalLLI = new int[100][30];
     signalStrength = new int[100][30];
     TEC = new double[1000];
     }

  public void setHeaderAndData(BufferedReader in) {
     String line;
     String label;
     try {
       while((line = in.readLine()) != null) {
         label = line.substring(60, 80).trim();
         // Чтение заголовка файла данных наблюдений
         if(label.equals("RINEX VERSION / TYPE")) setRinexVersionType(line);
         if(label.equals("PGM / RUN BY / DATE"))
setProgramAgencyDateTime(line);
         if(label.equals("# / TYPES OF OBSERV"))  setObservationNumber(line);
```

```
          if(label.equals("COMMENT"))              setComment(line);
          // Чтение записей данных из файла данных наблюдений
          if(label.equals("END OF HEADER"))        setData(in);
          }
        }
      catch(Exception e) { return; }
      return;
    }
/** RINEX VERSION / TYPE */

  private void setRinexVersionType(String line) {
      rinexVersion  = line.substring( 0, 20).trim();
      fileType      = line.substring(20, 40).trim();
      gnssType      = line.substring(40, 60).trim();
      return;
    }
/** PGM / RUN BY / DATE */

   private void setProgramAgencyDateTime(String line) {
      programName   = line.substring( 0, 20).trim();
      programAgency = line.substring(20, 40).trim();
      programDate   = line.substring(40, 60).trim();
      return;
   }
/** # / TYPES OF OBSERV */

   private void setObservationNumber(String line) {
      observationQuantity = Integer.parseInt(line.substring(0, 6).trim());
      for(int i=0;i<observationQuantity;i++) {
        observationType[i] = line.substring(10+i*6, 11+i*6).trim();
        observationBand[i] = Integer.parseInt(line.substring(11+i*6,
12+i*6).trim());
        }
      return;
   }
/** COMMENT */
   private void setComment(String line) {
      comment  = comment + line.substring(0, 60).trim() +"\n";
      return;
   }
/** Чтение данных наблюдений - эпоха и данные измерений. */
   public void setData(BufferedReader in) {
     try {
       // Чтение эпохи
       setEpoch(in);
       // Чтение данных наблюдений
       setObservations(in);
       }
     catch(Exception e) { return; }
     return;
    }
/** EPOCH / SAT */
   private void setEpoch(BufferedReader in) {
     String line;
     int b;
     try {
     if((line = in.readLine()) != null) {
      year              = Integer.parseInt(   line.substring( 1,  3).trim() );
      month             = Integer.parseInt(   line.substring( 4,  6).trim() );
      day               = Integer.parseInt(   line.substring( 7,  9).trim() );
      hour              = Integer.parseInt(   line.substring(10, 12).trim() );
```

```
      minute            = Integer.parseInt(   line.substring(13, 15).trim() );
      sec               = Double.parseDouble( line.substring(15, 26).trim() );
      flag              = Integer.parseInt(   line.substring(28, 29).trim() );
      satelliteQuantity = Integer.parseInt(   line.substring(29, 32).trim() );
      if(year<80) { year = 2000 + year; }
             else { year = 1900 + year; }
      for(int i=0;i<satelliteQuantity;i++) {
        b = i%12;
        if(b==0 && i!=0)
          line = in.readLine();
        satelliteSystem[i] = line.substring(32+3*b, 33+3*b).trim();
satelliteNumber[i]=Integer.parseInt( line.substring(33+3*b, 35+3*b).trim() );
        }}}
     catch(Exception e) { return; }
      return;
   }
/** OBSERVATIONS */
   public void setObservations(BufferedReader in) {
     String line;
     String str;
     int b;
     double f1, f2;
     double L1=0, L2=0;
     double dL;
     double k4 = 40.308;
     try {
         for(int i=0;i<satelliteQuantity;i++) {
         line = in.readLine();
            for(int j=0;j<observationQuantity;j++) {
            b = j%5;
            if(b==0 && j!=0)
            line = in.readLine();
signalData[i][j] = Double.parseDouble(line.substring(0+16*b, 14+16*b).trim());
            str = line.substring(14+16*b, 15+16*b).trim();
            if(str.equals("")==true) signalLLI[i][j] = 0;
            else signalLLI[i][j] = Integer.parseInt(str);
            str = line.substring(15+16*b, 16+16*b).trim();
           if(str.equals("")==true) signalStrength[i][j] = 0;
                        else signalStrength[i][j] = Integer.parseInt(str);
/*
           if(observationType[j].equals("L") && observationBand[j]==1)
             L1 = signalData[i][j] * getWaveLength(satelliteSystem[i],
satelliteNumber[i], observationBand[j]);
           if(observationType[j].equals("L") && observationBand[j]==2)
             L2 = signalData[i][j] * getWaveLength(satelliteSystem[i],
satelliteNumber[i], observationBand[j]);
*/
           if(observationType[j].equals("C") && observationBand[j]==1)
             L1 = signalData[i][j];
           if(observationType[j].equals("P") && observationBand[j]==2)
             L2 = signalData[i][j];
      }
        f1 = getFrequency(satelliteSystem[i], satelliteNumber[i], 1);
        f2 = getFrequency(satelliteSystem[i], satelliteNumber[i], 2);
        dL = f2*f2*(L1-L2)/(f1*f1-f2*f2);
      TEC[i] = f1*f1*f2*f2*(L1-L2)/(f1*f1-f2*f2)/k4;
        L = L1 - dL;
//Начало создания запроса к БД
      int number = -1;
      int prn0 = satelliteNumber[i];
System.out.println("Номер спутника "+prn0);
```

```
      double tsec1, t0test=0;
// Проверяем наличие драйвера для доступа к MySQL
      try {
      Class.forName("com.mysql.jdbc.Driver");
      }
      catch(Exception ex1) {
      ex1.printStackTrace();
      return;
      }
      try {
Connection c = DriverManager.getConnection(dataBaseUrl, dataBaseUser,
dataBasePassword);
Statement s = c.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
SatPosition XYZ = new SatPosition();
query = "select year(time), month(time), day(time), hour(time), minute(time),
second(time), af0, af1, af2, crs, deln, em0, cuc, e, cus, sqrta, t0e, cic,
omg0, cis, eye0, crc, w, omgdot, eyedot, prn from gps where prn="+prn0+" order
by abs(UNIX_TIMESTAMP(time)-UNIX_TIMESTAMP(\""+year+"-"+month+"-"+day+"
"+hour+":"+minute+":"+sec+"\")) limit 1;";
            ResultSet rs = s.executeQuery(query);
            int i2= 0;
    // Если такая запись есть в базе данных
            while(rs.next()) {
            XYZ.yeart = Integer.parseInt( rs.getString(1).trim() );
            XYZ.mont = Integer.parseInt( rs.getString(2).trim() );
            XYZ.dayt = Integer.parseInt( rs.getString(3).trim() );
            XYZ.hourt = Integer.parseInt( rs.getString(4).trim() );
            XYZ.mint = Integer.parseInt( rs.getString(5).trim() );
            XYZ.sect = Double.parseDouble( rs.getString(6).trim() );
            XYZ.af0 = Double.parseDouble( rs.getString(7).trim() );
            XYZ.af1 = Double.parseDouble( rs.getString(8).trim() );
            XYZ.af2 = Double.parseDouble( rs.getString(19).trim() );
            XYZ.crs = Double.parseDouble( rs.getString(10).trim() );
            XYZ.deln = Double.parseDouble( rs.getString(11).trim() );
            XYZ.em0 = Double.parseDouble( rs.getString(12).trim() );
            XYZ.cuc = Double.parseDouble( rs.getString(13).trim() );
            XYZ.e = Double.parseDouble( rs.getString(14).trim() );
            XYZ.cus = Double.parseDouble( rs.getString(15).trim() );
            XYZ.sqrta = Double.parseDouble( rs.getString(16).trim() );
            XYZ.t0e = Double.parseDouble( rs.getString(17).trim() );
            XYZ.cic = Double.parseDouble( rs.getString(18).trim() );
            XYZ.omg0 = Double.parseDouble( rs.getString(19).trim() );
            XYZ.cis = Double.parseDouble( rs.getString(20).trim() );
            XYZ.eye0 = Double.parseDouble( rs.getString(21).trim() );
            XYZ.crc = Double.parseDouble( rs.getString(22).trim() );
            XYZ.w = Double.parseDouble( rs.getString(23).trim() );
            XYZ.omgdot = Double.parseDouble( rs.getString(24).trim() );
            XYZ.eyedot = Double.parseDouble( rs.getString(25).trim() );
            XYZ.prn = Integer.parseInt( rs.getString(26).trim() );
            i2++;
               }
//Подставляем время эпохи и получаем секунды от GPS недели
      XYZ.t=XYZ.setTime(year,month,day,hour,minute,sec);
       // XYZ.t0c=XYZ.t0e;
XYZ.t0c=XYZ.setTime(XYZ.yeart,XYZ.mont,XYZ.dayt,XYZ.hourt,XYZ.mint,XYZ.sect);
//Вычисление положения спутника
      XYZ.getSatelliteXYZ(XYZ.t);
//перевод координат места в широту и долготу
      XYZ.setLatLonHeight(XYZ.xo,XYZ.yo,XYZ.zo);
//Вычисление широты долготы по координатам
```

```
      XYZ.getLonlat(XYZ.xk,XYZ.yk,XYZ.zk,XYZ.latitude,XYZ.longitude,XYZ.eht);
//Вычисление псевдодальности
      psL=XYZ.psdistance(XYZ.t);
//..Вывод на печать
System.out.printf(Locale.US, "%2.6f\t%6.6f\n",XYZ.elevation,(L-psL));
     rs.close();
     s.close();
     c.close();
   }
   catch(SQLException ex) { ex.printStackTrace(); return;
}
  catch(Exception e) { e.printStackTrace(); return;
}   }   }
     catch(Exception e) { return; }

    return;
   }

/** Частота радиосигнала */

   public double getFrequency(String satelliteSystem, int satelliteNumber, int
observationBand) {
     // Номера частотных литер спутников ГЛОНАСС
int[] glonass = {100, 1, 5, 6, 100, 1, 5, 6, -2, 4, 0, 100, -2, 4, 0, 100, -1,
-3, 3, 2, -1, -3, 3, 2};
     int k;
     // ГЛОНАСС
     if(satelliteSystem.equals("R")) {
       // Частотная литера спутника
       k = glonass[satelliteNumber-1];
       if(k==100) return 0.0;
       switch(observationBand) {
         case 1:  return (1602.0+k*0.5625)*1E+6;
         case 2:  return (1246.0+k*0.4375)*1E+6;
         default: return 0.0;
         }
       }
     // GPS
     if(satelliteSystem.equals("G")) {
       switch(observationBand) {
         case 1:  return 1575420000.0;
         case 2:  return 1227600000.0;
         case 5:  return 1176450000.0;
         default: return          0.0;
         }
       }
     return 0.0;
   }
/** Длина волны радиосигнала */

  public double getWaveLength(String satelliteSystem, int satelliteNumber, int
observationBand) {
    return 299792458.0/getFrequency(satelliteSystem, satelliteNumber,
observationBand);
   }
}
```

## Текст программы SatPosition.java

```
      /**
        * Расчет координат НКА, псевдодальности, угла места.
```

```
*@author Григорьев Г.В., Чукин В.В.
*Российский государственный гидрометеорологический университет (РГГМУ)
* Лицензия: GPL v.2
*/

   import java.util.*;
   import java.io.*;
   import static java.lang.Math.*;

public class SatPosition {
   private static final double gm    =  3.986005E+14;
   private static final double we = 7.2921151467E-5; // Угловая скорость З.
   private static final double gpspi =  3.1415926535898E+0; // Число Пи GPS
   private static final double bigf  = -4.442807633E-10;
   private static final int tunix = 31595400;  //*** начало GPS в секундах, с
начала UNIX
   private static final int mjd6jan80 = 44244;  //*** mjd for begin GPS---mjd
для начинать GPS
   private static int    mjd0 = -1; //*** ref part for times----ref часть в
течение времени
   private static final double re = 6378.135;
   // Параметры часов
   public double t0c;     // эпоха для коррекции часов, с
   public int yeart;
   public int mont;
   public int dayt;
   public int hourt;
   public int mint;
   public double sect;
   public double t;
   public double af0;     // сдвиг часов, с
   public double af1;     // скорость сдвига часов, с/с
   public double af2;     // ускорение сдвига часов, с/(с*с)
   public double dt;
   public double ek;
   public double dtr;
   public double sqrta;   //*  square root of semimajor axis           sqrt(m)
   public double e;       //*  eccentricity
   public double a;
   public double en0;
   public double en;
   public double tk;
   public double em0;     //*  mean anomaly at ref. epoch                 (r)
   public double t0e;
   public double deln;   //*  mean motion difference from computed value (r/s)
   public double sinek;
   public double cosek;
   public double f;
   public double u;
   public double w;
   public double cu2;
   public double su2;
   public double duk;
   public double drk;
   public double dik;
   // Параметры возмущения
   public double cuc;     //*  cosine harmonic correction to arg. of lat. (r)
   public double cus;     //*  sine harmonic correction to arg. of lat.   (r)
   public double crc;     //*  cosine harmonic correction to radius       (m)
   public double crs;     //*  sine harmonic correction to radius         (m)
   public double cic;     //*  cosine harmonic correction to inclination  (r)
```

```
   public double cis;     //*  sine harmonic correction to inclination    (r)
   public double uk;
   public double rk;
   public double eyek;
   public double ceyek;
   public double eye0;
   public double eyedot;
   public int prn;
   public double xpk;
   public double ypk;
   public double omgk;
   public double comgk;
   public double somgk;
   public double omgdot;
   public double omg0;
   public double xk;
   public double yk;
   public double zk;
   public double em;
   public double ecca0;
   public double ecca;
   public double eccano;
   public double emk;
   public double latitude;
   public double longitude;
   public double eht;
   public double azimuth;
   public double elevation;
   //Положение антенны
   public final double xo=-2243185.5144;
   public final double yo=-3856771.1329;
   public final double zo=4542744.8698;
public SatPosition() {
}
/** Расчет декартовых геоцентрических координат XYZ спутника.*/

public static double setTime(int iyr, int imo, int idy,
                        int ihr, int imn, double sec) {
// Получаем текущую дату
      int tsec = (int)Math.floor(sec);
     TimeZone.getDefault().setID("Europe/Moscow") ;
      // Формируем календарь на основе текущей даты
      Calendar currentCalendar = Calendar.getInstance();
      // Устанавливаем значение календаря в соответствии с датой
      currentCalendar.set(Calendar.YEAR, 1980);
      currentCalendar.set(Calendar.MONTH, 0);
      currentCalendar.set(Calendar.DAY_OF_MONTH, 6);
      currentCalendar.set(Calendar.HOUR, 0);
      currentCalendar.set(Calendar.MINUTE, 0);
      currentCalendar.set(Calendar.SECOND, 0);
      currentCalendar.set(Calendar.MILLISECOND, 0);
      currentCalendar.set(Calendar.AM_PM, 0);
      Calendar calendar = Calendar.getInstance();
      calendar.set(Calendar.DAY_OF_MONTH, idy);
      calendar.set(Calendar.HOUR, ihr);
      calendar.set(Calendar.MINUTE, imn);
      calendar.set(Calendar.SECOND, tsec);
      calendar.set(Calendar.MILLISECOND, 0);
      calendar.set(Calendar.MONTH, imo-1);
      calendar.set(Calendar.YEAR, iyr);
      calendar.set(Calendar.AM_PM, 0);
```

```
// Находим разницу между двумя календарями в милисекундах
   long diff = calendar.getTimeInMillis() - currentCalendar.getTimeInMillis();
        long seconds = diff / 1000;        // в секундах
        long minutes = seconds / 60;        // в минутах
        long hours = minutes / 60; // в часах
        long days = hours / 24; // в днях
        double    ngweek = seconds/604800;
        double gsec = seconds%604800;
        return gsec;
    }

public void getSatelliteXYZ(double t) {
      // Если началась другая GPS-неделя
      if((t-t0e) > 302400.0) t = t - 604800.0;
      if((t-t0e) <-302400.0) t = t + 604800.0;
      // Корректировка спутникового времени
      double dt = af0 + af1*(t-t0c) + af2*(t-t0c)*(t-t0c);
      t = t - dt;
      // Если началась другая GPS-неделя
      if((t-t0e) > 302400.0) t = t - 604800.0;
      if((t-t0e) <-302400.0) t = t + 604800.0;
      double ek = eccano(t);
      double dtr = bigf * e * sqrta*sin(ek);
      t = t - dtr;
      // Если началась другая GPS-неделя
      if((t-t0e) > 302400.0) t = t - 604800.0;
      if((t-t0e) <-302400.0) t = t + 604800.0;
      double a   = sqrta*sqrta;
      double en0 = sqrt(gm/(a*a*a));
      // Время, прошедшее от исходной эпохи
      double tk = t - t0e;
      // Скорректированное среднее движение
      double en = en0 + deln;
      // Средняя аномалия (M)
      double emk = em0 + en*tk;
      // Решение уравнения Кеплера для эксцентрической аномалии (E)
      ek    = kepEQ(emk);
      double sinek = sin(ek);
      double cosek = cos(ek);
      // Истинная аномалия (f)
      double f = atan2(sqrt(1.0-e*e)*sinek, cosek-e);
      // Аргумент широты
      double u   = f + w;
      double cu2 = cos(u+u);
      double su2 = sin(u+u);
      // Поправки к радиусу-вектору
      double duk = cuc*cu2 + cus*su2;
      double drk = crc*cu2 + crs*su2;
      double dik = cic*cu2 + cis*su2;
      // Исправленный аргумент широты
      double uk    = u + duk;
      // Исправленный радиус-вектор
      double rk    = a*(1.0-e*cosek) + drk;
      // Исправленный угол наклона орбиты
      double eyek  = eye0 + eyedot*tk  + dik;
      double ceyek = cos(eyek);
      // Позиция в орбитальной плоскости
      double xpk = rk*cos(uk);
      double ypk = rk*sin(uk);
      // Исправленная долгота восходящего узла орбиты
      double omgk  = omg0+(omgdot-we)*tk-we*t0e;
```

```
      double comgk = cos(omgk);
      double somgk = sin(omgk);

      // Декартовы геоцентрические координаты спутника
      xk = xpk*comgk-ypk*somgk*ceyek;
      yk = xpk*somgk+ypk*comgk*ceyek;
      zk =           ypk      *sin(eyek);
      return;
   }
/** Решение уравнения Кеплера для эксцентрической аномалии.*/

 public double kepEQ(double em) {
      double ecca0;
      double ecca = em + e*sin(em);
      do {
         ecca0 = ecca;
         ecca = em + e*sin(ecca0);
      }
      while(abs((ecca-ecca0)/ecca) > 1.0E-14);     // выход из цикла только
при достижении заданной точности
      return ecca;
   }
/** Вычисление эксцентрической аномалии в момент времени t.*/

  public double eccano(double t) {
      // Если началась другая GPS-неделя
      if((t-t0e) > 302400.0) t = t - 604800.0;
      if((t-t0e) <-302400.0) t = t + 604800.0;
      double a = sqrta*sqrta;            // большая ось
      double en0 = sqrt(gm/(a*a*a));     // среднее движение
      double tk = t - t0e;               // время с исходной эпохи
      double en = en0 + deln;            // скорректированное среднее движение
      double emk = em0 + en*tk;          // средняя аномалия (M)
      return kepEQ(emk);                 // решение уравнения Кеплера для
эксцентрической аномалии (E)
   }
//Вычисление псевдодальности до спутника по заданной эпохе

public double psdistance(double t){
double dist;
dist=sqrt(pow((xk-xo),2)+pow((yk-yo),2)+pow((zk-zo),2));
System.out.printf(Locale.US,"Псевдодальность =%.4f\n",dist);
return dist;
   }
//Перевод координат

public void setLatLonHeight(double X,double Y,double Z)
   {
      final double TOL = 1.0E-14;
      final double A = 6378137.0;              // эллипсоид grs-80
      final double E2 = 0.006694380022903416;  // (geod.hand.1992)
      final double AE2 = A*E2;                 // (66(2), p. 191)
      double p;
      double tgla;
      double tglax;
      double sinLat;
      double w;
      double n;
      int i = 0;
      final int N = 10;
      // Вычисление начальной оценки широты (eht=0)
```

```
      p = Math.sqrt(X*X + Y*Y);
      tgla = Z/p/(1.0-E2);
      // Итерационный процесс продолжается либо до достижения заданной
точности,
      // либо до максимального числа итераций
      while(i < N)
       {
         tglax = tgla;
         tgla = Z/(p-(AE2/sqrt(1.0+(1.0-E2)*tgla*tgla)));
         i++;
         // Проверка достижения заданной точности
         if(abs(tgla-tglax) <= TOL)
          {
            latitude = atan(tgla);
            sinLat = sin(latitude);
            longitude = atan2(Y, X);
            w = sqrt(1.0-E2*sinLat*sinLat);
            n = A/w;
            if(abs(latitude) < 0.7854) eht = p/cos(latitude) - n;
            else                       eht = Z/sinLat - n + E2*n;
         return;
          }
       }
// Если за заданное максимальное количество итераций не достигнута сходимость,
// то возможно координаты слишком близки к полюсу
System.err.println("Fail to converge in xyzgeo()");
System.exit(1);

   }

  public void getLonlat(double xsat,double ysat,double zsat,double lat,double
lon,double alt) {
double twopi=2*gpspi;
double rx,ry,rz,theta,top_s,top_e,top_z;
double az,el,rg;
lat=lat*gpspi/180;//перевод градусов (Grad) в радианы (Rad)
lon=lon*gpspi/180;//перевод градусов (Grad) в радианы (Rad)
  theta=lon;
  rx = xsat - xo;
  ry = ysat - yo;
  rz = zsat - zo;
  top_s = sin(lat)* cos(theta)*rx + sin(lat)* sin(theta)*ry - cos(lat)*rz;
  top_e = - sin(theta)*rx + cos(theta)*ry;
  top_z = cos(lat)* cos(theta)*rx + cos(lat)* sin(theta)*ry + sin(lat)*rz;
  az = atan(-top_e/top_s);
  if (top_s > 0)
    {az = az + gpspi;}
  if (az < 0)
    {az = az + (2*gpspi);}
  rg = sqrt(rx*rx + ry*ry + rz*rz);
  el = asin(top_z/rg);
azimuth=az*180/gpspi;//обратный перевод
elevation=el*180/gpspi;//обратный перевод
}}
```

# ПРИЛОЖЕНИЕ В

## Фрагмент расчета тропосферной задержки программой «GPS Tropodelay»

| Угол места НКА, град | Тропосферная задержка, м |
|---|---|
| 1.2992 | 50.9872 |
| 53.8073 | 3.3654 |
| 32.5117 | 4.0472 |
| 14.5999 | 9.9557 |
| 62.3881 | 2.6402 |
| 35.4969 | 5.8692 |
| 44.8027 | 2.7246 |
| 33.3776 | 3.3277 |
| 16.5322 | 7.6308 |
| 5.6178 | 23.8736 |
| 1.2847 | 51.6747 |
| 53.8444 | 3.0399 |
| 32.5416 | 4.2966 |
| 14.5914 | 10.0125 |
| 62.3673 | 3.0769 |
| 35.4634 | 6.1759 |
| 44.7705 | 2.9607 |
| 33.3537 | 2.9636 |
| 16.5031 | 7.8757 |
| 5.6475 | 20.0126 |
| 1.2703 | 51.5172 |
| 53.8814 | 2.8915 |
| 32.5715 | 3.8803 |
| 14.5829 | 10.1067 |
| 62.3465 | 2.4202 |
| 35.4300 | 5.9596 |
| 44.7383 | 2.7070 |
| 33.3298 | 2.8948 |
| 16.4739 | 8.1635 |
| 5.6772 | 17.1962 |
| 1.2558 | 51.6184 |
| 53.9185 | 2.6490 |
| 32.6013 | 3.4174 |
| 14.5744 | 10.4633 |
| 62.3255 | 2.2171 |
| 35.3966 | 5.8290 |
| 44.7061 | 2.6813 |
| 33.3058 | 2.9966 |
| 1.2992 | 50.9872 |
| 53.8073 | 3.3654 |
| 32.5117 | 4.0472 |
| 14.5999 | 9.9557 |
| 62.3881 | 2.6402 |
| 35.4969 | 5.8692 |
| 44.8027 | 2.7246 |