

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное
учреждение
высшего профессионального образования
«Московский физико-технический институт
(государственный университет)»
Факультет радиотехники и кибернетики
Кафедра информационных систем

**ВОССТАНОВЛЕНИЕ ОРБИТЫ КО ПО
РАДИОЛОКАЦИОННЫМ НАБЛЮДЕНИЯМ С
УЧЁТОМ ВЛИЯНИЯ ИОНОСФЕРЫ**

Выпускная квалификационная работа
(магистерская диссертация)

Направление подготовки: 03.04.01 Прикладные математика и физика

Выполнил:

студент 115 группы Морген Матвей Владиславович

Научный руководитель:

д.т.н. Лагуткин Владимир Николаевич

Москва 2017

Содержание

Введение	3
1 Влияние ионосферы на прохождение радиосигналов	5
1.1 Строение ионосферы	5
1.2 Модель ионосферы	7
2 Алгоритмы и модели ПАК	10
2.1 Постановка задачи	11
2.2 Модуль иммитации измерений	11
2.2.1 Модель радиолокационных измерений	11
2.2.2 Модель влияния ионосферы	13
2.2.3 Модель движения Космических Объектов	14
2.2.4 Модель ошибок	16
2.2.5 Общая схема иммитатора	17
2.3 Модуль восстановления орбиты	18
2.4 Модуль чтения TLE файлов	20
2.5 Модуль выгрузки и визуализации данных	23
3 Программная реализация	25
3.1 Общая информация	25
3.2 Парсинг TLE-файлов	27
3.3 Иммитатор радиоизмерений	28
3.4 Восстановитель орбит	33
Список литературы	35

Введение

РЛС наблюдают цели через околоземную среду, которую можно рассматривать как канал связи между РЛС и целями. Поэтому РЛС в большей или меньшей степени подвержены разнообразному влиянию околоземной среды. Характер и величина этого влияния зависит от состояния околоземной среды, типа и параметров ее возмущений, от используемого в средстве диапазона длин электромагнитных волн и координат РЛС.

Компоненты околоземной среды имеют различные электромагнитные характеристики. Для нейтральной и ионизированной компонент среды основной электромагнитной характеристикой является комплексная диэлектрическая проницаемость ϵ , которая определяет показатели преломления и поглощения электромагнитных волн в зависимости от частоты. Для конденсированной компоненты, образованной облачными и аэрозольными частицами, основными электромагнитными характеристиками являются объёмные поперечники рассеяния и поглощения.

Для определения диэлектрической проницаемости нейтральной компоненты (атмосферы) необходимо задавать давление, температуру, влажность, для ионизированной компоненты (ионосферы) – плотность и температуру электронов, ионов и нейтральных частиц, для конденсированной компоненты – распределение частиц (облачных и аэрозольных) по составу и размерам, полную концентрацию частиц.

Строго говоря, указанные параметры являются случайными функциями пространственных координат и времени. В нормальных условиях средние значения этих параметров зависят от географических координат (долготы и широты), высоты, времени суток, сезона, а также специфических индексов активности влияния внешних факторов, к числу которых относятся индексы солнечной, геомагнитной и вулканической активности. Параметры облачности и облачных частиц зависят, к тому же, и от метеорологических условий. В настоящее вре-

мя существуют достаточно надежные эмпирические модели пространственно-временных изменений средних значений параметров компонент околоземной среды в естественных нормальных условиях, в частности, для ионизированной компоненты – модель ионосферы (IRI, Nequick).

Проведённые к настоящему времени исследования эффектов взаимодействия электромагнитных волн радиолокационных диапазонов с пространственно неоднородной околоземной средой сделали возможным получение надежных количественных моделей эффектов прохождения сигналов в околоземной среде. Важнейшей частью работы современных радиолокационных устройств является задача компенсации искажений вносимых ионосферой.

Целью данной работы является доработка программного иммитатора влияния околоземной среды, разработка и реализация моделей восстановления орбиты КО¹ с учётом и без влияния околоземной среды, тестирование и сравнение работы алгоритмов реализующих модели.

¹Космических Объектов

Глава 1

Влияние ионосферы на прохождение радиосигналов

Для описания и количественной оценки эффектов, возникающих при распространении электромагнитных волн в земной атмосфере, необходимо знать её электрические свойства. В атмосфере Земли выделяются две существенно отличающиеся по своим свойствам области: тропосфера (нижняя часть атмосферы, ниже 60км) и ионосфера (соответственно – часть атмосферы выше 60км) [1].

Тропосфера представляет из себя смесь нейтральных газов и её можно рассматривать как недиспергирующий изотропный диэлектрик. В ионосфере же, существенную роль играет ионизация воздуха, обусловленная действием ультрафиолетового излучения Солнца. Ионосферу в диапазоне УКВ¹ волн можно рассматривать как холодную и анизотропную электронную плазму [1].

1.1 Строение ионосферы

Ионосферой называют ионизированную область верхних слоев атмосферы Земли. Ионизация возникает главным образом под действием ультрафиолетово-

¹Ультракороткие волны – радиоволны дапазона от метровых до децимиллиметровых, т.е. от 30 МГц до 3000 МГц

го излучения Солнца, в результате чего образуются положительно заряженные ионы и свободные электроны. Кроме того, в процессе ионизации участвуют рентгеновские лучи, излучаемые солнечной короной и корпускулярные потоки Солнца. Вследствие низкой плотности атмосферы на большой высоте ионы и свободные электроны рекомбинируют сравнительно медленно, образуется ионизированный слой газа, находящийся в состоянии динамического равновесия [1].

Ионосфера расположена на высотах от 60 до 1000 км. Высота и плотность определяются интенсивностью УФ² излучения, которое убывает по мере прохождения атмосферы, от разряженных её слоев к более плотным. Ионосфера состоит из нескольких основных слоёв, плавно переходящих один в другой [1].

В дневные часы возникают четыре основных максимума ионизации:

- Область F_2 , высота 250-450 километров.
- Область F_1 , высота 180-200 километров.
- Область E , высота 100-200 километров.
- Область D , высота 60-80 километров.

Слои F_1 , E и D достаточно устойчивы [1].

После захода Солнца прекращается ионизация атмосферы и начинается процесс рекомбинации электронов, который более активно проходит в плотных слоях атмосферы [1]. Этим объясняется быстрое исчезновение наиболее низкого слоя D . Уменьшаются и сливаются слои F_1 и F_2 . В ночные часы ионосфера состоит только из двух слоёв: слоя E , с пониженной концентрацией свободных электронов, и слоя F_2 , который в ночные часы обозначается символом F без индекса.

По своим свойствам ионосфера эквивалентна полупроводнику [1,2]. Поэтому ионосфере свойственны отражающие, преломляющие и ослабляющие свойства прохождению сквозь неё радиоволн.

²Ультрафиолетовое

Было установлено, что при частоте сигнала $F_{кр} = \sqrt{80,8 \cdot N_e}$ кГц, где N_e – удельная электронная концентрация в ионосфере, радиоволна перестаёт взаимодействовать с ионосферой. При $F > F_{кр}$ волна, пришедшая из космического пространства, преломляется и уходит к поверхности нашей планеты. А при $F < F_{text}$ волна отклоняется обратно или поглощается ионосферой [1].

Поглощающие свойства ионосферы зависят от колебаний свободных электронов. После возбуждения волной электрон сталкивается с нейтральной молекулой или ионом газа и отдаёт полученную им энергию. Энергия радиоволны превращается в энергию движения частиц газа, т.е. в тепловую. Исходя из этих постулатов, было произведена градация на радиоволны КВ³ и УКВ диапазонов. КВ диапазон используется для дальней радиосвязи между объектами, находящимися на поверхности нашей планеты, УКВ для связи с ИСЗ⁴, локации планет и радиоастрономических наблюдений.

1.2 Модель ионосферы

Характеристикой среды, в которой распространяются электромагнитные волны, служит тензор диэлектрической проницаемости $\hat{\epsilon}(r, t)$ [1].

На частотах радиолокационного диапазона и в условиях статического поля Земли можно воспользоваться высокочастотным приближением для тензора диэлектрической проницаемости холодной плазмы, в котором учтён только вклад электронов и пренебрегается вклад ионов и влияние теплового движения. В данном приближении, компоненты тензора

$$\hat{\epsilon} = \begin{pmatrix} \epsilon_{xx} & \epsilon_{xy} & \epsilon_{xz} \\ \epsilon_{yx} & \epsilon_{yy} & \epsilon_{yz} \\ \epsilon_{zx} & \epsilon_{zy} & \epsilon_{zz} \end{pmatrix}$$

³Короткие волны, до 30МГц

⁴Искусственный спутник земли

определяющего линейную связь $D = \hat{\varepsilon}E$ между индукцией и напряжённостью электрического поля, имеют вид:

$$\begin{aligned}\varepsilon_{xx} &= 1 - \frac{v(1+is)}{(1+is)^2 - u}; & \varepsilon_{yy} &= 1 - \frac{v[(1+is)^2 - u \sin^2 \alpha]}{(1+is)[(1+is)^2 - u]}; \\ \varepsilon_{yy} &= -\varepsilon_{yx} = \frac{iv\sqrt{u} \cos \alpha}{(1+is)^2 - u}; & \varepsilon_{yz} &= \varepsilon_{zy} = \frac{uv \cos \alpha \sin \alpha}{(1+is)[(1+is)^2 - u]}; \\ \varepsilon_{xz} &= -\varepsilon_{zx} = -\frac{iv\sqrt{u} \sin \alpha}{(1+is)^2 - u}; & \varepsilon_{zz} &= 1 - \frac{v[(1+is)^2 - u \cos^2 \alpha]}{(1+is)[(1+is)^2 - u]};\end{aligned}$$

Здесь α – угол между направлением распространения волны и статическим магнитным полем H_0, v, us – стандартные плазменные параметры:

$$v = \frac{\omega_p^2}{\omega^2}; \quad u = \frac{\omega_H^2}{\omega^2}; \quad s = \frac{v}{\omega}$$

где $\omega_p^2 = 4\pi N e^2 / m$ – квадрат плазменной частоты; $\omega_H = eH_0 / mc$ – гирочастота; v – эффективная частота соударений электронов.

Численно значения параметров $v\sqrt{u} = \frac{\omega_h}{\omega}$ удобно определять по формулам:

$$\begin{aligned}v &= 3,18 \cdot 10^9 \cdot \left(\frac{N}{\omega^2}\right) = 8,06 \cdot 10^7 \cdot \left(\frac{N}{f^2}\right); \\ \sqrt{u} &= 0,22 \cdot 10^6 \cdot \left(\frac{H_0}{\omega}\right) = 3,52 \cdot 10^4 \cdot \left(\frac{H_0}{f}\right)\end{aligned}$$

где $f = \frac{\omega}{2\pi}$ – частота радиоволн в герцах; H_0 – напряжённость магнитного поля, А/м. Для УКВ параметры v, us малы по сравнению с единицей [1]. Поэтому для моделей, связанных с УКВ, часто используются приближения основанные на малости этих параметров.

Пренебрегая всеми этими параметрами (в том числе влиянием геомагнитного поля), получаем, что тензор $\hat{\varepsilon}$ сводится к скаляру:

$$\begin{aligned}\varepsilon_\omega(r, t) &= 1 + \mu_\omega(r, t) \\ \mu_\omega(r, t) &= -3,187 \cdot 10^3 \frac{N_e(r, t)}{\omega^2}\end{aligned}$$

$\mu_\omega(r, t)$ – приведённая диэлектрическая проницаемость, N_e – электронная концентрация, $\omega = 2\pi f$, f – частота.

В условиях малости $\mu_\omega(r, t)$ влияние ионосферы на фазовый и групповой пути, углы прихода и доплеровское смещение частоты электромагнитных волн описывается поправками первого порядка по приближениям геометрической оптики [1–3].

Поправка к групповому пути волны:

$$\begin{aligned} D_l(R) &= -\frac{1}{2} \int_0^R \mu(\vec{r}_0(s), t) ds = \frac{E(R)}{\omega_c^2} \\ E(R) &= 1,593 \cdot 10^3 \int_0^R N_e(\vec{l}_0, t) ds \\ \vec{r}_0 &= \vec{l}_0 s, \quad \vec{l}_0 = \frac{\vec{R}}{R}, \quad R = |\vec{R}|, \quad \vec{R} = \vec{R}(t) \end{aligned} \quad (1.1)$$

где $\vec{R} = \vec{R}(t)$ – координаты объекта.

Поправка к фазовому пути:

$$L_l(R) = -D_l(R) = -\frac{E(R)}{\omega^2}$$

Поправка к доплеровской скорости:

$$V_l(R) = \frac{\partial L_l(R)}{\partial t} = \frac{1}{2} [\mu(\vec{r}(R), t) V_r + \frac{V_t}{R} \int_0^R s \nabla \mu(\vec{r}_0(s), t) ds],$$

где V_r – радиальная скорость объекта, V_t – поперечная, по отношению к \vec{l}_0 , скорость объекта. Поправка к единичному вектору направления прихода волны:

$$\vec{l}_l(R) = \frac{1,584 \cdot 10^3}{\omega_c^2} \int_0^R (1 - \frac{s}{R} \nabla_\perp N_e(\vec{l}_0 s)) ds, \quad (1.2)$$

где $\nabla_\perp \cdot \mu$ – поперечный по отношению к \vec{l}_0 градиент функции $\mu(\vec{r})$

Глава 2

Алгоритмы и модели ПАК

Для упрощения восприятия внутреннего устройства программы, её можно разделить на два глобальных модуля: модуль моделирования и имитации измерений координат КО и модуль восстановления орбиты по координатам, полученным в результате радиолокационных измерений. Или иммитатор и восстановитель соответственно.

Для получения иммитированных количественных значений измерений, получаемых в результате работы РЛС¹, будет проводиться моделирование этого процесса. Предполагается, что пользователь вводит в алгоритм параметры орбит КО и параметры одной или нескольких РЛС. Для расчётов электронной концентрации ионосферы будет использована модель NeQuick [2], а для расчёта пространственных координат и скоростей КО набор моделей SDP/SGP [4]. В результате своей работы иммитатор сгенерирует набор точек в которых был замечен КО, заданных координатами, временем наблюдения и РЛС (и её параметры) производившей измерение.

Модуль восстановления орбиты в качестве входных параметров принимает результаты измерений проведённые РЛС – т.е. те данные которые являются выходными для модуля имитации, а на выходе выдаёт параметры орбиты

¹Радиолокационная Станция

которые считает наиболее приближенными к истинным (т.е. тем которые были заданы на вход иммитатору).

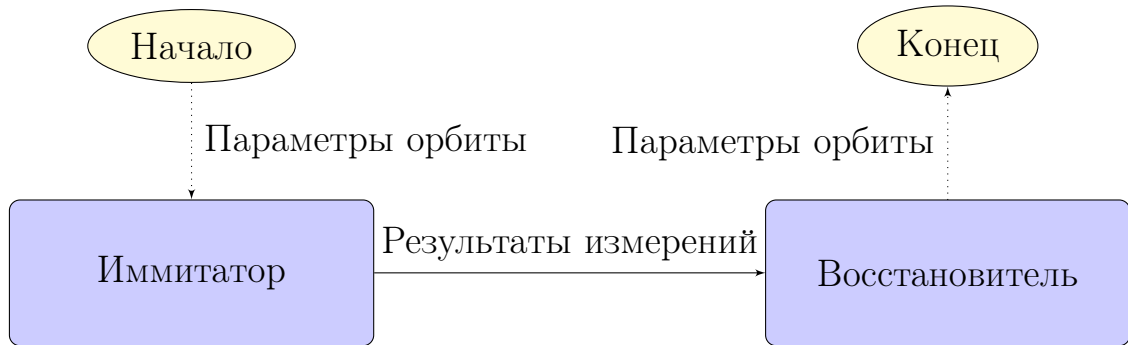


Рисунок 2.1: Общая схема работы программы

2.1 Постановка задачи

Задача сравнить истинные параметры орбиты и параметры орбиты получаемые в результате работы второго модуля при различных параметрах среды, РЛС и орбитах КО.

2.2 Модуль иммитации измерений

Как уже говорилось ранее модуль иммитации производит моделирование процесса измерений координат спутников производимый РЛС. Результатом работы этого модуля должно быть множество координатных точек КО с меткой времени момента измерения. В идеальном случае в результате иммитации должен получиться такой набор точек, который получился бы при работе реальной РЛС в случае совпадения всех параметров.

2.2.1 Модель радиолокационных измерений

Одни и те же результаты радиолокационных измерений можно представить во множестве разных форматов и в разных системах координат. Одним из популярных способов записи и передачи информации об обнаруженных объектах

является запись в локальной для РЛС системе координат таких параметров как Азимут², Угол Места³ и Расстояние до объекта. Такой формат удобен в первую очередь операторам РЛС, т.к. фактически не требует от них никакой обработки поступающих данных. Однако при обработке информации поступающей с нескольких РЛС или при сведении результатов с картами или другими средствами которые используют глобальную для Земли систему координат такой формат становится неудобным.

Несмотря на то, что при суммуляции рассчитываются фактически снимаемые радаром данные (азимут, угол места и расстояние до цели), при обработке информации более удобным бдует записывать результаты записанные в географических координатах. В этом же формате удобно задавать местоположение РЛС. Географические координаты представляют из себя точку на координатной сфере задаваемую широтой высотой и долготой $\vec{r}(\varphi, \lambda, h)$, где

широта - это угол φ между направлением зенита и плоскостью экватора, а долгота - угол λ между плоскостью нулевого меридиана и меридиана, проходящего через задаваемую точку. Высота h задаётся в метрах над уровнем моря.

Так же для работы модели радиолокационных измерений требуются параметры характеризующие саму РЛС:

- Координаты РЛС
- Рабочая частота ω

²Горизонтальный угол между направлением на север и направлением на объект с вершиной в точке установки РЛС

³Вертикальный угол между касательной проведённой к поверхности земли, где расположена РЛС к направлению на объект

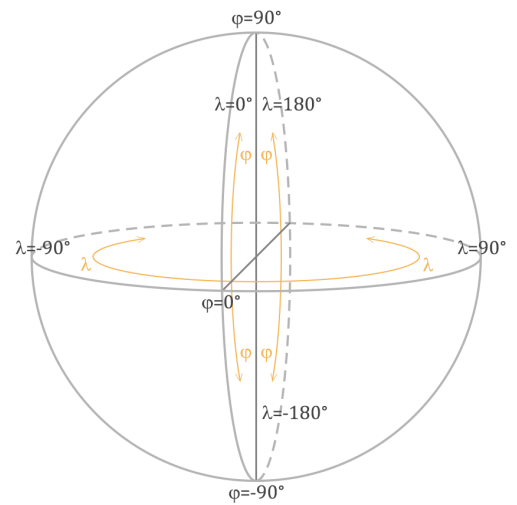


Рисунок 2.2: Координатная сфера

- Наблюдаемый сектор (биссектриса угла наблюдения и углы обзора).

Стоит отметить, что в генерируемые иммитатором точки попадают только те точки траектории спутников, которые попали в наблюдаемый сектор.

Таким образом результатом работы модели радиолокационных измерений будет набор точек, заданных географическими координатами. А входными параметрами будут характеристики РЛС. Однако для полноценной симуляции измерений этого недостаточно, полная модель измерений требует учесть влияние ионосферы, расчёт которой вынесен в отдельную модель.

2.2.2 Модель влияния ионосферы

Для вычисления систематической ошибки, которую получит радар, наблюдая за указанным спутником, в определённой точке его траектории нужно вычислить поправки к групповому пути радиоволны при помощи выражений 1.1 (для вычисления видимого расстояния) и 1.2 (для угла).

Величины ω – частота волны, равняющаяся рабочей частоте РЛС, начальные и конечные пределы интегрирования являются параметрами модели.

Для расчёта электронной концентрации N_e между радаром и спутником используется модель NeQuick [2], которая принимает следующие параметры:

- Географические координаты начальной и конечной точек (точка-РЛС и точка-спутник).
- Время измерения.
- Солнечная активность⁴

Так как наша модель использует модель NeQuick для своих расчётов, то все входные параметры требуемые ею так же требуются и для работы нашей модели. Соответственно параметрами нашей модели будут:

⁴Интенсивность потока энергии радиоизлучения Солнца на частоте 2.8 ГГц (длина волны – 10.7 см).

- Координаты РЛС
- Частота работы РЛС
- Координаты спутника
- Время измерений
- Солнечная активность

На выходе модель даст нам поправку к измеренным углам и расстоянию, которые необходимо учесть, чтобы получить те же результаты которые получились бы при реальных измерениях.

2.2.3 Модель движения Космических Объектов

Для расчёта местоположения КО, моделями SDP/SGP требуются параметры орбиты задаются следующими величинами:

- Время эпохи спутника ⁵
- Коэффициент торможения B^*
- Наклонение орбиты спутника к плоскости экватора Земли.
- Долгота ⁶ восходящего узла ⁷.
- Эксцентриситет.
- Аргумент перицентра⁸

⁵Момент времени, в который были определены параметры орбиты.

⁶Один из основных элементов орбиты, используемый для математического описания ориентации плоскости орбиты относительно базовой плоскости (плоскость экватора в нашем случае). Определяет угол в базовой плоскости, образуемый между базовым направлением на нулевую точку и направлением на точку восходящего узла орбиты. Нулевая точка для Земли – первая точка Овна (точка весеннего равноденствия), угол измеряется от направления на нулевую точку против часовой стрелки.

⁷Точка, в которой движущееся по орбите тело пересекает условную плоскость в северном направлении (то есть переходит из южного полушария небесной сферы в северное).

⁸Угол между направлениями из притягивающего центра на восходящий узел орбиты и на перицентр.

- Частота обращения (среднее движение).
- Средняя аномалия⁹

В модели SDP/SGP в качестве результата вычислений являются координаты спутника в декартовой системе отсчёта, не вращающейся вместе с Землёй. Нам такие координаты не подходят, т.к. в остальных моделях используются координаты в системе отсчёта привязанной к Земле и вращающейся вместе с ней. Поэтому потребуются выражения для пересчёта координат из одной системы в другую и обратно.

Для перевода из сферических координат в прямоугольные, можно воспользоваться общеизвестными выражениями:

$$\begin{aligned}x &= r \cos(\phi) \cos(\lambda) \\y &= r \cos(\phi) \sin(\lambda) \\z &= r \sin(\phi)\end{aligned}\tag{2.1}$$

Соответственно, находя обратные значения для выражений 2.1, получим:

$$\begin{aligned}\phi &= \arcsin\left(\frac{z}{\sqrt{x^2 + y^2 + z^2}}\right) \\ \lambda &= \arctan(\phi) \sin(\lambda) \\ r &= \sqrt{x^2 + y^2 + z^2}\end{aligned}\tag{2.2}$$

Выражения 2.1 и 2.2 позволяют переводить координаты из сферической (географической) в прямоугольную систему координат и обратно, если обе они вращаются вместе с Землёй. Для перевода из сферической системы, которая вращается, в невращающуюся прямоугольную систему координат, нужно учесть угол на который повернулась земля, за время от начала дня до момента, когда

⁹Средняя аномалия для тела, движущегося по невозмущённой орбите – произведение его среднего движения и интервала времени после прохождения перицентра. Другими словами, средняя аномалия ни что иное, как угловое расстояние от перицентра гипотетического тела, движущегося с постоянной угловой скоростью, равной среднему движению.

спутник проходил расчётную точку:

$$\begin{aligned}\lambda' &= \lambda - \theta_{\text{Земли}} \\ \theta_{\text{Земли}} &= \omega_{\text{Земли}} \cdot t_{\text{суток}}\end{aligned}\tag{2.3}$$

где $\omega_{\text{Земли}} = 7,29 \cdot 10^{-5} \frac{\text{рад}}{\text{с}}$ – скорость вращения Земли, $t_{\text{суток}}$ – время прошедшее от начала суток.

В виде функции работу модели расчёта движения КО можно описать так:

$$\begin{aligned}\vec{x}(t_{\text{наб}}) &= f_{\text{SDP/SGP}}(p_1, \dots, p_8, t_{\text{наб}}) \\ \vec{x} &= (x, y, z)\end{aligned}\tag{2.4}$$

2.2.4 Модель ошибок

Ещё одной важной особенностью работы РЛС является присутствие случайных ошибок в её измерениях. В целом ошибки измерений РЛС можно разделить на следующие виды [5]

- Грубые ошибки – возникающие в результате просчётов операторов.
- Систематические ошибки – вызываются длительно действующими факторами, например, задержкой сигнала или влиянием ионосферы.
- Случайные ошибки – зависят от случайных обстоятельств, которые не поддаются предварительному учёту.

В нашей модели грубые ошибки и систематические ошибки (кроме ошибок возникающих в результате влияния ионосферы) отсутствуют, а случайные ошибки будут генерироваться генератором случайных чисел в пределах заданных пользователем. В измерениях РЛС есть три величины и, соответственно, три величины погрешности для азимута, угла места и расстояния. Обозначим их буквами α , β и γ , а пределы погрешности теми же буквами, но с индексом *ошибк*. Функцию, генерирующую случайные числа обозначим как $f_{\text{слч}}$, таким

образом:

$$\begin{aligned}
 \alpha &= \alpha_{\text{изм}} + f_{\text{слч}}(\alpha_{\text{ошбк}}) \\
 \alpha &= \beta_{\text{изм}} + f_{\text{слч}}(\beta_{\text{ошбк}}) \\
 \alpha &= \gamma_{\text{изм}} + f_{\text{слч}}(\gamma_{\text{ошбк}}), \\
 \text{где } f_{\text{слч}}(x) &\in [-x, +x]
 \end{aligned}
 \tag{2.5}$$

2.2.5 Общая схема иммитатора

Для симуляции наблюдений спутника радаром, будет производиться расчёт положения всех заданных пользователем КО в указанный период времени. Затем, зная наблюдаемые и истинные параметры, добавив случайную погрешность к наблюдаемым данным, имитатор вычислит ошибку и запишет её. Наглядно схему работы программы можно увидеть на рисунке 2.3.

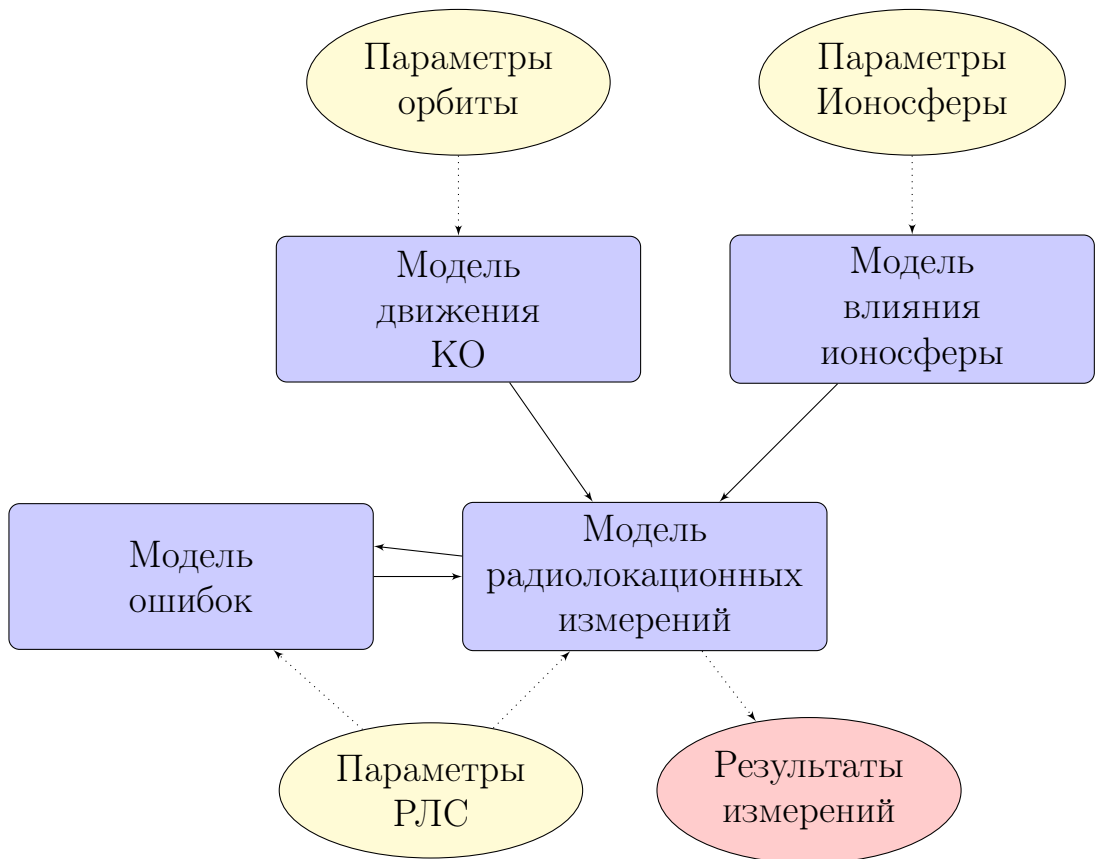


Рисунок 2.3: Общая схема работы модуля иммитации

2.3 Модуль восстановления орбиты

Модуль восстановления орбиты необходим для получения параметров орбиты наблюдаемого спутника по результатам измерений его траектории. Входными параметрами модуля являются массив точек (координаты и время) в которых РЛС наблюдала космический объект.

Алгоритм путём перебора возможных параметров орбиты и ионосферы должен найти такой набор этих параметров, которые при подачи в качестве входных в модуль иммитации (при неизменных других неорбитальных параметров) даст наиболее приближенную измеренную траекторию к измеренной изначально, в целом, схему работы модуля можно представить в виде рисунка 2.4.

Стоит отметить, что для достижения целей работы (сравнение качества восстановления орбиты с учётом ионосферы и без) алгоритм восстановления может работать в двух режимах, в первом режиме подбираются все параметры орбиты и параметры ионосферы (солнечная активность p_{flux}), во втором же режиме, солнечная активность приравнивается к нулю т.е. ионосферное влияние не учитывается.

Точным решением этого модуля могут быть только истинные параметры орбиты (те которые были изначально поданы на вход иммитатору), но из-за присутствия случайных ошибок в измерениях параметры орбиты наилучшим образом подходящие как решение всегда будут отличаться от изначальной орбиты.

Лучший набор параметров орбиты будет искать при помощи метода наименьших квадратов, т.е. лучшим набором найденных параметров будем считать такой набор, который даёт минимум суммы квадратов разностей входных точек и результатов полученных при подборе орбиты:

$$\sum_i (x_i - f_i(p_1, \dots, p_8, p_{\text{flux}}))^2 \rightarrow \min_{p_1, \dots, p_8, p_{\text{flux}}} \quad (2.6)$$

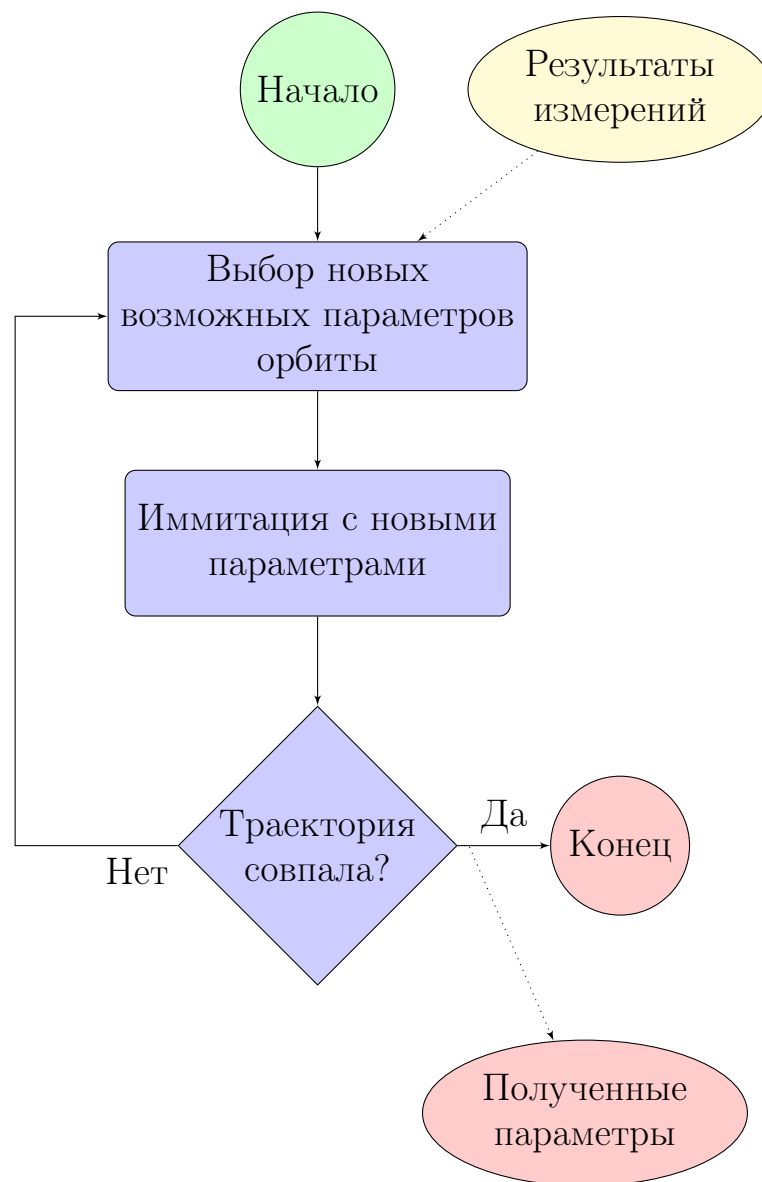


Рисунок 2.4: Общая схема работы модуля восстановления орбиты

Фактически задача сведена к поиску локального минимума выражения 2.6.

Полный перебор всех вариантов параметров слишком сложный подход в плане вычислений, каждая итерация подбора вызывает полную симуляцию процесса измерений, соответственно даже небольшое уменьшение количества итераций может значительно сократить время поиска решения. Для оптимизации поиска решения вместо полного перебора разумно воспользоваться одним из известных методов нахождения локального экстремума – градиентным спуском.

Так же, зная что размер случайной погрешности измерений мал относительно измеряемых величин, начальную точку поиска решения (параметры орбиты с которых начнётся подбор) можно задавать в точке истинного решения, т.е. те параметры орбиты, которые были поданы на вход иммитатору. Это позволяет алгоритму поиска минимума сразу же оказаться около искомой точки.

Для иммитации измерений производимых в процессе подбора параметров симулирование случайных погрешностей должно быть отключено, поскольку их наличие будет мешать алгоритму остановиться на последних шагах, где изменение параметров достаточно мало по сравнению с размерами погрешностей.

2.4 Модуль чтения TLE файлов

Для записи параметров орбит космических объектов часто используется простой двухстрочный формат TLE¹⁰, который, как может быть ясно из названия, представляет запись орбитальных параметров в виде двух строк текста длиной 69 символов. Модель SDP/SGP [4] так же использует данные из TLE формата для вычисления положения спутника в определённое время. Орбитальные элементы определяются для многих тысяч космических объектов из базы данных NORAD и свободно распространяются для дальнейшего использования в Интернете. TLE всегда состоит из двух строк форматированного текста. Кроме того, им может предшествовать строка с названием объекта.

Подробное описание этих строк, на примере записи (Листинг 2.1) для одного из модулей Международной Космической Станции, можно увидеть в таблицах 2.1 и 2.2, а побитовое разбиение можно увидеть на рисунке 2.5.

Контрольная сумма строк вычисляется путём сложения значащих цифр в строке. Для каждого знака "минус" в строке к сумме добавляется единица. Все остальные символы игнорируются. В получившейся сумме берется последняя цифра.

¹⁰Two-line Element Set, двухстрочный набор элементов

Следует также помнить, что приведённые NORAD в двустрочных элементах аргумент перигея, наклонение и другие классические Кеплеровы элементы вычисляются при помощи усреднения в рамках модели SDP/SGP и не являются Кеплеровыми элементами оскулирующей орбиты.

Листинг 2.1: TLE-Запись Для Модуля Заря Международной Космической Станции

```
ISS (ZARYA)
1 25544U 98067A    08264.51782528  -.00002182   00000-0  -11606-4  0   2927
2 25544   51.6416 247.4627 0006703 130.5360 325.0288 15.72125391563537
```

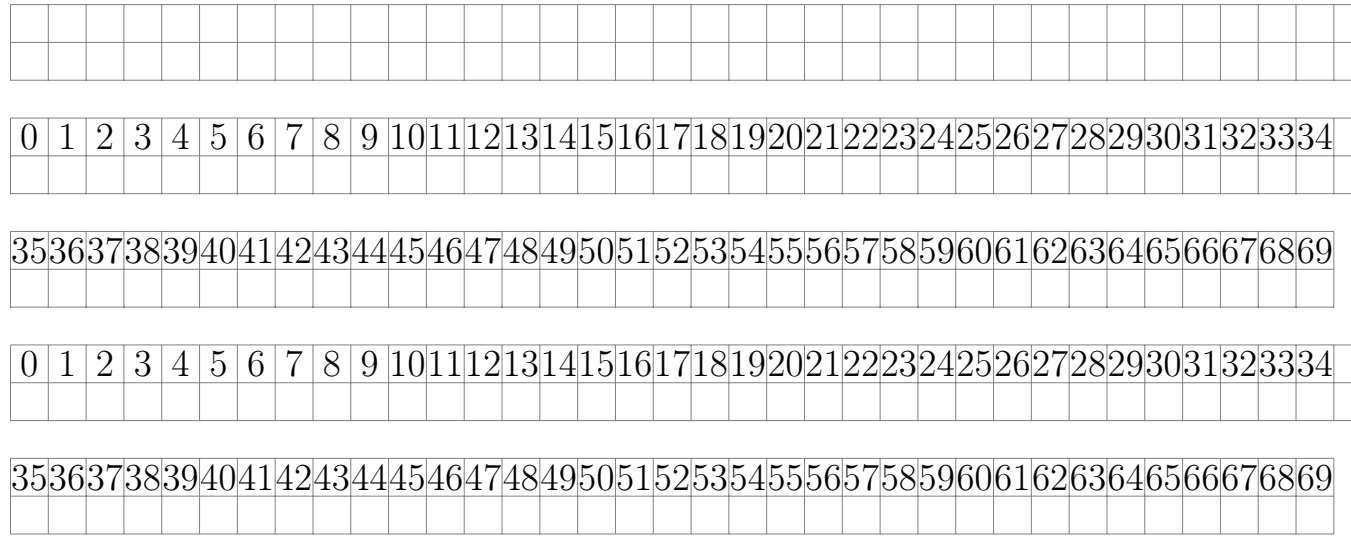


Рисунок 2.5: Формат TLE побитово

Номер	Положение	Содержание	Пример
1	01-01	Номер строки	1
2	03-07	Номер спутника в базе данных NORAD	25544
3	08-08	Классификация (U=Unclassified)	U
4	10-11	Международное обозначение (год запуска)	98
5	12-14	Международное обозначение (номер запуска)	067
6	15-17	Международное обозначение (часть запуска)	A
7	19-20	Год эпохи (последние две цифры)	08
8	21-32	Время эпохи	264.51782528
9	34-43	$\frac{\text{виток}}{\text{день}^2}$	-.00002182
10	45-52	$\frac{\text{виток}}{\text{день}^3}$	00000-0
11	54-61	Коэффициент торможения B^*	-11606-4
12	63-63	Число 0	0
13	65-68	Номер (версия) элемента	292
14	69-69	Контрольная сумма по модулю 10	7

Таблица 2.1: Описание первой строки TLE

Номер	Положение	Содержание	Пример
1	01-01	Номер строки	2
2	03-07	Номер спутника в базе данных NORAD	25544
3	09-16	Наклонение в градусах	51.6416
4	18-25	Долгота восходящего узла в градусах	247.4627
5	27-33	Эксцентриситет	0006703
6	35-42	Аргумент перицентра в градусах	130.5360
7	44-51	Средняя аномалия в градусах	325.0288
8	53-63	Частота обращения $\frac{\text{виток}}{\text{день}}$	15.72125391
9	64-68	Номер витка на момент эпохи	56353
10	69-69	Контрольная сумма по модулю 10	7

Таблица 2.2: Описание второй строки TLE

2.5 Модуль выгрузки и визуализации данных

После проведения всех расчётов для дальнейшей обработки данные должны быть выгружены в формате удобном для чтения и обработки в модуле визуализации, а также, быть может, загрузки в другие программы для более сложного анализа или визуализации.

Поскольку данные о параметрах вычисленных орбит хорошо записываются в виде таблиц, то для выгрузки этих данных идеально подойдёт формат CSV¹¹ – текстовый формат созданный для представления табличных данных.

Заголовок генерируемого текстового вывода представлен в листинге 2.2:

Листинг 2.2: Пример заголовка выходных данных (Одна строка)

```
DragCoefficient , InclinationAngle , AscendingNode , Eccentricity ,
MeanDistance , noise , flux , type
```

Где соответственно величины:

- Коэффициент торможения.
- Угол наклона.
- Долгота восходящего узла.
- Эксцентриситет.
- Аргумент перицентра.
- Средняя аномалия.
- Среднее движение.
- Средняя величина отклонения полученной траектории от заданной.
- Уровень шума.

¹¹Comma-Separated Values – значения, разделённые запятыми

- Коэффициент солнечной активности (всегда ноль для случая без ионосферы).
- Тип запуска (с ионосферой или без).

Все угловые величины договоримся выводить в радианах, расстояния в метрах, время в секундах, а скорости в метрах в секунду.

Глава 3

Программная реализация

В предыдущей главе была составлена общая абстрактная модель имитации, в этой же главе будет описана её реализация. Для написания программы был выбран язык C++, в качестве библиотеки графического интерфейса – Qt. Объектноориентированный стиль программы хорошо подошёл для поставленной задачи. Во-первых, в программе сами собой напрашиваются многие объекты: РЛС, спутник и пр., во-вторых, между ними хорошо строятся наследственные связи (это будет показано далее по тексту), а в-третьих, объектноориентированный подход позволил спроектировать программу таким образом, чтобы любой её модуль или даже часть модуля можно было легко заменить или сделать несколько реализаций.

3.1 Общая информация

Программа разделена на четыре основные части:

- Библиотека для имитации и восстановления параметров орбиты по результатам наблюдений, содержащую в себе всю логику относящуюся к симулированию радионаблюдений, реализацию всех необходимых для этого моделей из предыдущей главы и прикладные модули, необходимые для нормальной работы.

- Библиотека для работы с методом градиентного спуска и методом наименьших квадратов.
- Графический интерфейс, позволяющий удобно использовать написанную программу.
- Текстовый постпроцессор, позволяющий выгружать данные для обработки в других программах.

Читая текст работы может создаться впечатление, что архитектура программы была спроектирована заранее (т.к. изложение идёт от постановки задачи к её реализации модуль за модулем), но это совсем не так. Программа множество раз меняла свою структуру, реализацию, архитектуру и пр., по мере улучшения понимания поставленной задачи и её теоретических аспектов. Однако не все части программы переписывались, для идеального вписывания в новую архитектуру, из-за этого некоторые функции могут не использоваться, классы могли получиться перегруженными или наоборот – слишком разряженными, API местами неудобным и т.д. Всё это результат многократного рефакторинга. Конечно, в будущем все такие недостатки можно исправить.

3.2 Парсинг TLE-файлов

Для парсинга TLE-строк описанных в параграфе 2.4 на языке C++ можно воспользоваться функцией *sscanf* (см. листинг 3.1), предварительно заменив все пробелы на нули на позициях, предшествующих десятичным разделителям т.к. в TLE-формате принято не писать 0 перед десятичной частью.

Листинг 3.1: Чтение TLE строк

```

if (longstr1[44] != '\0')
    longstr1[43] = longstr1[44];
longstr1[44] = '.';
if (longstr1[7] == '\0')
    longstr1[7] = 'U';
if (longstr1[9] == '\0')
    longstr1[9] = '.';
for (j = 45; j <= 49; j++)
    if (longstr1[j] == '\0')
        longstr1[j] = '0';
if (longstr1[51] == '\0')
    longstr1[51] = '0';
if (longstr1[53] != '\0')
    longstr1[52] = longstr1[53];
longstr1[53] = '.';
longstr2[25] = '.';
for (j = 26; j <= 32; j++)
    if (longstr2[j] == '\0')
        longstr2[j] = '0';
if (longstr1[62] == '\0')
    longstr1[62] = '0';
if (longstr1[68] == '\0')
    longstr1[68] = '0';
// First tle-line
sscanf(longstr1, "%2d%5ld%1c%10s%2d%12lf%11lf%7lf%2d%7lf%2d%2d%6ld",
        &cardnumb, &satrec.satnum, &classification, intldesg,
        &satrec.epochyr, &satrec.epochdays, &satrec.ndot,
        &satrec.nddot, &nexp, &satrec.bstar, &ibexp, &numb, &elnum );
// Second tle-line
if (longstr2[52] == '\0')
    sscanf(longstr2, "%2d%5ld%9lf%9lf%8lf%9lf%9lf%10lf%6ld\n",
            &cardnumb, &satrec.satnum, &satrec.inclo,
            &satrec.nodeo, &satrec.ecco, &satrec.argpo,
            &satrec.mo, &satrec.no, &revnum );
else
    sscanf(longstr2, "%2d%5ld%9lf%9lf%8lf%9lf%9lf%11lf%6ld\n",
            &cardnumb, &satrec.satnum, &satrec.inclo,
            &satrec.nodeo, &satrec.ecco, &satrec.argpo,
            &satrec.mo, &satrec.no, &revnum );

```

3.3 Иммитатор радиоизмерений

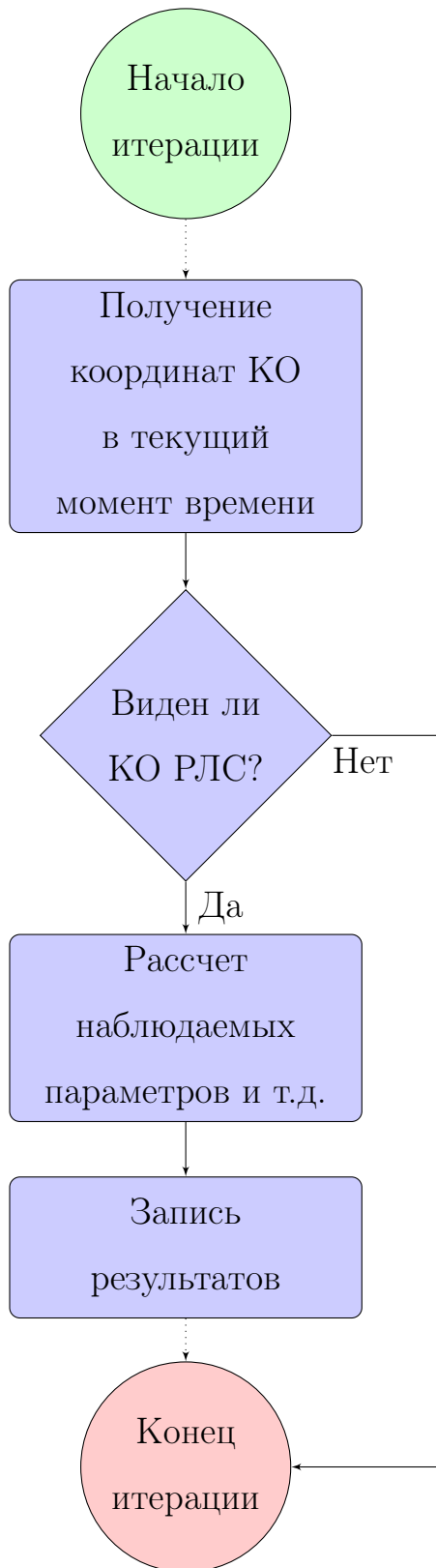


Рисунок 3.1: Граф потока управления иммитатора

Иммитатор разделён на следующие подмодули реализующие абстрактные модели (см. рисунок 2.3):

- OrbitCalculator, модуль расчёта орбит спутников – реализация модели движения КО.
- NeQuick, модуль расчёта влияния ионосферы – реализация модели влияния ионосферы.
- RadarStation, модуль отвечающий за логику самой РЛС и проводимые её измерения – реализация модели выбора КО и модели радиолокационных измерений, здесь же будет вноситься случайная ошибка измерений.

Так же в иммитатор входит *ImitationDriver* управляющий модуль, запускает и управляет циклами имитации, направляет и именует потоки данных, может взаимодействовать с пользователем или внешними модулями и т.д

Определив все модули программы, можно определить то, как будет выглядеть один цикл вычислений, т.е. проведение одного измерения радиолокационной станцией в определённый

момент времени. Блок-схемы потока управления и потока данных представлены на рисунке 3.1 и 3.2.

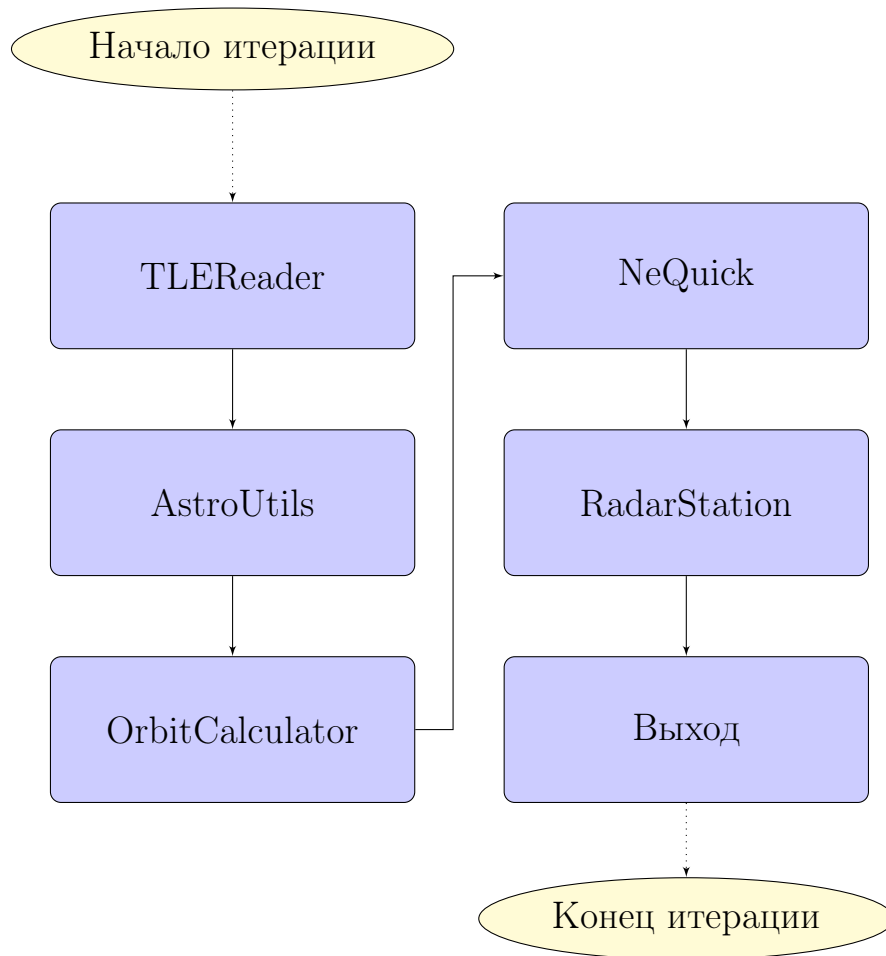


Рисунок 3.2: Граф потока данных иммитатора

ImitationDriver занимается управлением циклами и слежением за тем, чтобы во время каждого цикла модулям были доступны соответствующие этому циклу данные. *TLEReader* в практической реализации не участвует в каждом цикле – *OrbitCalculator* обращается к нему только на первой итерации, для получения списка спутников и их параметров.

Как уже говорилось ранее, основные объекты, участвующие в имитации, хорошо и удобно объявляются в виде классов. В модуле имитации определены следующие классы:

Определение 3.1. Тип **GeoPoint** – Класс для представления координат любых объектов в географической системе координат. Имеет методы *GetLatitude*,

GetLongitude, *GetAltitude*, *DistanceTo* для получения соответственно широты, долготы, высоты и расстояния до любой другой заданной точки.

Определение 3.2. Тип **OrbitPoint** – Класс для представления координат любых объектов находящихся на орбите Земли в прямоугольной не вращающейся вместе с Землёй системе координат. Является наследником класса *GeoPoint* (см. Определение 3.1) Имеет методы *GetIntertialX*, *GetIntertialY*, *GetIntertialZ* для получения координат x, y и z. А также метод *GetTime* для получения момента времени в который спутник находился в указанных координатах.

Определение 3.3. Тип **RadarStation** – Класс для представления координат РЛС и её параметров. Является наследником класса *GeoPoint* (см. Определение 3.1) Имеет методы *GetWorkFrequency* для получения рабочей частоты радара, *IsInSigh* – для получения информации о том, попадает, переданная в качестве аргумента метода, точка в зону обнаружения РЛС (т.е. видит ли РЛС эту точку), *ObservedDistanceTo* для вычисления измеряемого расстояния до точки (т.е. с учётом ионосферы) координат x, y и z. А также методы *ZenithAngleTo* и *AzimuthAngleTo* для получения углов наблюдения точки с учётом влияния ионосферы.

Вызов любого из методов: *ObservedDistanceTo*, *ZenithAngleTo* или *AzimuthAngleTo*, фактически приводит к запуску одной итерации цикла имитации (см. рисунок 3.1).

Кроме этого в модуле есть классы для работы с орбитами, которые могут содержать в себе множество точек типа *OrbitPoint* (3.2):

Определение 3.4. Тип **Orbit** – Класс для представления орбиты спутника и её параметров. Позволяет вычислить и сохранить местоположения спутника в указанные моменты времени. Вызов любого из его методов – *GetTrajectoryPoint* или *GetTrajectoryPoints* запустит имитацию по модели SDP/SGP (см. параграф 2.2.3).

Для сохранения результатов имитации создан специальный класс *SightReport*

Определение 3.5. Тип **SightReport** – Класс представляющий из себя отчёт о наблюдении одной РЛС.в один момент времени Сохраняет координаты наблюдения, время и параметры наблюдавшей РЛС, а так же истинные значения всех величин.

Из определений видно, что *RadarStation* и *OrbitPoint* наследуются от *GeoPoint*, таким образом, удаётся добиться простоты в расчётах расстояния и координат для них. Для точек орбит функции *GetLatitude*, *GetLongitude* будут возвращать координаты подспутниковой точки.

Схем классов слегка отличаются от тех, что были составлены для модулей иммитатора, главным образом в целях оптимизации производительности и читаемости кода.

Размер случайной погрешности будет задаваться через сигму и рассчитываться при помощи встроенной в C++ математической библиотеки (см. листинг 3.3).

Листинг 3.2: Добавление случайной величины к измеренной

```
double astroutils::RandomizeValue(double expected, double sigma)
{
    assert(sigma < 1);
    double normalized_sigma = fabs(expected * sigma);
    std::normal_distribution<double> distribution(expected, normalized_sigma);

    return distribution(astroutils::generator);
}
```

Полностью описать работу иммитатора можно блок-схемой изображённой на рисунке 3.3



Рисунок 3.3: Полная схема работы иммитатора

3.4 Восстановитель орбит

Для реализации метода градиентного спуска была использована библиотека GSL¹. Для сравнения качества восстановления орбиты с учётом и без учёта ионосферы необходимо реализовать два метода для каждого из них соответственно, методы различаются лишь частью связанной с получением параметров т.к. для одного из них отсутствует ионосферный параметр – солнечная активность (см

Листинг 3.3: Добавление случайной величины к измеренной

```
int orbitresolver::CalculationFunctionWithFlux(const gsl_vector * x, void *data,
                                              gsl_vector * f)
{
    restore_function_parameters *parameters =
        (restore_function_parameters *) data;
    const size_t result_size = parameters->objects_observations->size();
    time_t start_time =
        parameters->objects_observations->begin()->GetObservationTime();
    time_t end_time =
        parameters->objects_observations->back().GetObservationTime();
    time_t epoch =
        parameters->real_orbit->GetEpochTime();
    std::list<SightObject> *objects_observations =
        parameters->objects_observations;
    RadarStation *radar = parameters->observer;

    double drag_coefficient = gsl_vector_get(x, 0);
    double inclination_angle = gsl_vector_get(x, 1);
    double ascending_node = gsl_vector_get(x, 2);
    double eccentricity = gsl_vector_get(x, 3);
    double apsis_argument = gsl_vector_get(x, 4);
    double mean_anomaly = gsl_vector_get(x, 5);
    double mean_motion = gsl_vector_get(x, 6);

    Orbit orbit(0, *parameters->real_orbit,
                drag_coefficient, inclination_angle, ascending_node,
                eccentricity, apsis_argument, mean_anomaly, mean_motion);
    orbit.SetEpoch(parameters->real_orbit->GetEpochTime());

    radar->SetSigma(0);
    // radar->SetLocalFlux(gsl_vector_get(x, 7));
    radar->SetLocalFlux(parameters->observer->GetLocalFlux());

    ImitationDriver driver;
    driver.addRadarStation(*radar);
    driver.addSatellite(orbit);
}
```

¹Scientific Library

```

    auto simulation_report = driver.RunImitation(start_time, end_time);
    std::list<SightObject> result_objects;
    for(auto it = simulation_report.begin(); it != simulation_report.end();
        ++it)
    {
        auto observation_report = it->second;
        for(auto observation_iterator = observation_report.begin();
            observation_iterator != observation_report.end();
            ++observation_iterator) {
            if(observation_iterator->GetObservedObjects().size() < 1)
                result_objects.push_back(SightObject(0, 0, 0, 0, 0));
            else
            {
                auto observe = *observation_iterator;
                result_objects.push_back(observe.GetObservedObjects()[0]);
            }
        }
    }

    size_t i = 0;
    for (auto it = result_objects.begin(),
        it2 = objects_observations->begin();
        it != result_objects.end() && it2 != objects_observations->end()
        && i < result_size; ++it, ++it2, ++i)
    {
        double delta = pow(it->GetDistanceTo()
            - it2->GetDistanceTo(), 2);
        //std::cout << "Delta : " << delta << std::endl;
        gsl_vector_set(f, i, delta);
    }

    return GSL_SUCCESS;
}

```

Список литературы

1. *О.А.Кравцов З.И.Фейзулин А.Г.Виноградов.* Прохождение радиоволн через атмосферу Земли. — Москва: Радио и связь, 1983. — 224 с.
2. *S. B.Nava P.Coisson.* A new version of the Nequick ionosphere electron density model // *J. Atmos.* — 2008. — Vol. Solar-Terr.
3. *В.Д Гацковский А.В. Курикша А.А. Лагуткин В.Н. Лукьянов А.П. Хуторовский З.Н. Шилин.* Анализ методов компенсации ионосферных ошибок определения параметров движения КО по данным радиолокационных наблюдений. — Москва: МАК Вымпел, 2015. — 8 с.
4. *Felix R. Hoots Ronald L. Roehrich.* MODELS FOR PROPAGATION OF NO-RAD ELEMENT SETS // *Aerospace Defense Command United States Air Force.* — 1980. — December. — Vol. 3.
5. *П.А. Бакулев.* Радиолокационные системы: Учебник для вузов. — Москва: Радиотехника, 2004.