

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Национальный исследовательский университет «МЭИ»

Институт: *ИРЭ*

Кафедра: *Радиотехнических систем*

Специальность:

*11.05.01 Радиоэлектронные системы и
комплексы*

ОТЧЕТ по практике

**Наименование
практики:**

Производственная практика: научно-
исследовательская работа

СТУДЕНТ

 / Тасканов В.Е. /
(подпись) (Фамилия и инициалы)

Группа ЭР-15-16
(номер учебной группы)

**ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ
ПО ПРАКТИКЕ**

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

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

/ Корогодин И.В. /
(подпись) (Фамилия и инициалы члена комиссии)

**Москва
2021**

СПИСОК ИСПОЛНИТЕЛЕЙ

студент

Тасканов В.Е.

Содержание

1	КРАТКИЙ ОБЗОР 4-Х ГНСС СИСТЕМ	6
1.1	ОПИСАНИЕ ГНСС ГЛОНАСС	8
1.2	ОПИСАНИЕ ГНСС GPS	9
1.3	ОПИСАНИЕ ГНСС GALILEO	11
1.4	ОПИСАНИЕ ГНСС BEIDOU	13
2	ОПРЕДЕЛЕНИЯ ДАЛЬНОСТИ ОТ ПОТРЕБИТЕЛЯ ДО СПУТНИКА ИЛИ КООРДИНАТ ПОТРЕБИТЕЛЯ	15
2.1	ПСЕВДОДАЛЬНОМЕРНЫЙ МЕТОД	15
2.2	ИСТОЧНИКИ ОШИБОК, ВЛИЯЮЩИХ НА ТОЧНОСТЬ НАВИГАЦИОННО-ВРЕМЕННЫХ ОПРЕДЕЛЕНИЙ	17
2.3	ОЦЕНКА КООРДИНАТ ПОТРЕБИТЕЛЯ С УЧЕТОМ ОШИБОК SISRE	19
3	ПОСТАНОВКА ЗАДАЧИ НАХОЖДЕНИЯ ОЦЕНКИ КООРДИНАТЫ ПОТРЕБИТЕЛЯ С УЧЕТОМ ОШИБОК SISRE	20
4	РЕШЕНИЯ ЗАДАЧИ НАХОЖДЕНИЯ ОЦЕНКИ КООРДИНАТЫ ПОТРЕБИТЕЛЯ С УЧЕТОМ ОШИБОК SISRE	21
5	АЛГОРИТМЫ ОПРЕДЕЛЕНИЯ КООРДИНАТ	22
5.1	АЛГОРИТМ ДЛЯ GPS	22
5.2	АЛГОРИТМ РАСЧЕТА КООРДИНАТ ГЛОНАСС.....	24
5.3	АЛГОРИТМ РАСЧЕТА КООРДИНАТ GALILEO	27
6	РЕАЛИЗАЦИЯ НОВЫХ АЛГОРИТМОВ	30
7	ПОСТРОЕНИЯ ГРАФИКОВ СО СЛУЧАЙНО ЗАДАНЫМИ КООРДИНАТАМИ И ВРЕМЕНЕМ	31
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	37

ВВЕДЕНИЕ

В настоящее время спутниковые радионавигационные системы (СРНС) являются самыми точными по определению пространственно-временных характеристик потребителя, благодаря чему они широко используются как в гражданских, так и военных целях. На данный момент имеется четыре крупных СРНС: ГЛОНАСС, GPS, Galileo, Beidou, все они различаются своим расположением на орбите, количеством выведенных спутников и другими значимыми элементами.

Многие задаются вопросами какая из этих систем все-таки точнее, но дальше экспериментальных значений, с помощью обычных приложений сравнения обычно не происходят, для этого мы разработает ПО и алгоритм к нему, для произведения оценки предсказания точности в определенное время и в определенное место, а потом сравним полученные результаты

Цель работы – произвести разработку алгоритма предсказания точности навигационного приемника по широте, долготе и высоте для четырёх СРНС и произвести статистическое сравнение по точности в плане и по высоте, даваемой систем ГЛОНАСС, GPS, Galileo, Beidou.

Для достижения данной цели были выполнены следующие **задачи**:

1. Обзор 4-х ГНСС систем
2. Разработка алгоритма предсказания точности навигационного приемника по широте, долготе и высоте при работе по системам ГЛОНАСС или GPS
3. Разработка алгоритма предсказания точности навигационного приемника по широте, долготе и высоте при работе по системе Galileo

4. Разработка алгоритма предсказания точности навигационного приемника по широте, долготе и высоте при работе по системе Beidou

5. Проведение статистического эксперимента по сравнению точности в плане и по высоте, даваемой системами ГЛОНАСС, GPS, Galileo, Beidou.

1 КРАТКИЙ ОБЗОР 4-Х ГНСС СИСТЕМ

Для решения навигационных уравнений методом МНК, и соответственно, определения координат, требуется как минимум 4 видимых спутника.

Алгоритм итогового определения и решений уравнений во всех системах схож, но системы отличаются техническими характеристиками и орбитальными группировками, которые влияют на зону покрытия, а так же на точность определения координат, которая вытекает через определения геометрического фактора.

Так же следует учесть, что во всех 4-ех системах разная геоцентрическая система координат, но они примерно одинаковые и дают различия с точностью 2-3 см.

Все навигационные сигналы СРНС лежат в диапазоне частот 1100-1610 МГц.

Во всех сигналах СРНС с точки зрения потребителя используются только два вида модуляции – BPSK и BOC.

Расположение спутников на орбитальной группировке

Глонасс

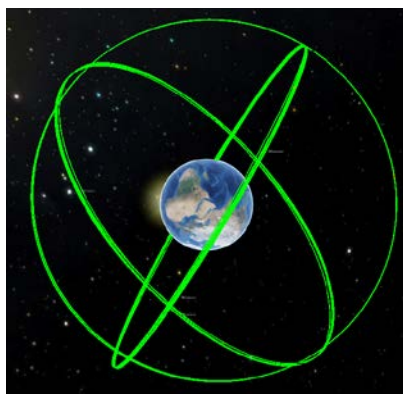


Рисунок 1.1 –
Расположение
спутников Глонасс

GPS

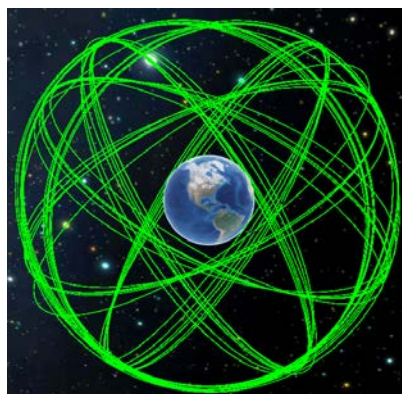


Рисунок 1.2 –
Расположение
спутников GPS

Galileo

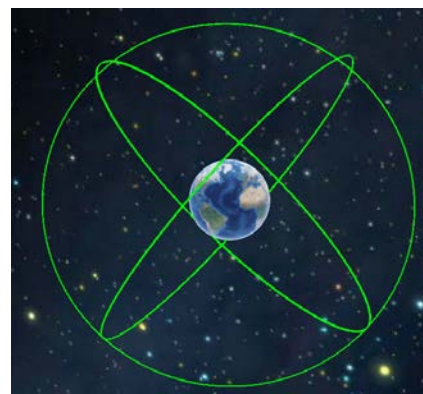


Рисунок 1.3 –
Расположение
спутников Galileo

Beidou

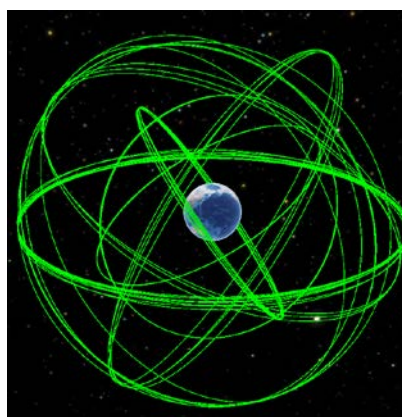


Рисунок 1.4 –
Расположение
спутников Beidou

Как видим, на рисунках 1.1-1.4, орбитальные группировки категорически отличаются у всех ГНСС. Более подробное описание орбит ниже.

1.1 ОПИСАНИЕ ГНСС ГЛОНАСС

Глобальная навигационная спутниковая система (ГЛОНАСС) – российская спутниковая радионавигационная система, которая передает как гражданские, так и военные сигналы, которые доступны в любой точке земного шара, в любое время, так как сигналы передаются непрерывно.

Орбита устроена таким образом, что наибольший коэффициент геометрического фактора у земного полюса, собственного, наименьший на экваторе, это происходит за счет того, что орбита имеет наклонение 64.8 гр., и где у остальных систем геометрический фактор, за счет количества видимых спутников, увеличивается, в системе Глонасс, наоборот уменьшается, таким образом точность там выше.

Спутниковая группировка разбита на три плоскости, в каждой из которых по 8 равномерно распределённых спутников.

Созвездие спутников теоретических должно состоять из 24 спутников, но на данный момент находится 25 спутников, 23 из которых используются.

Высота орбиты составляет 19 100 км.

Характерное отличие от других СРНС является частотное разделение сигналов, за счёт чего улучшается помехоустойчивость.

Геоцентрическая система координат ПЗ-90.11

Одно из главных отличий от других глобальных навигационных спутниковых систем – это шкала времени

Навигационно-временные определения в системе ГЛОНАСС осуществляются с использованием следующих шкал времени (ШВ):

- шкала всемирного времени (UT1);
- шкала универсального координированного времени государственного первичного эталона Российской Федерации UTC(SU);

- шкала времени системы (ШВС) ГЛОНАСС;
- шкала московского декретного времени (МДВ);
- бортовая шкала времени (БШВ);
- шкалы времени навигационных сигналов (ШВ {обозначение сигнала});
- шкала времени аппаратуры потребителя (ШВП);
- Международная шкала атомного времени (TAI)

начало отсчёта 1996 г 01.01.1996 00ч 00 мин

Формат времени: $N_4:N_T:t$

N_4 – номер четырехлетнего периода;

N_T – текущая дата, календарный номер суток, внутри четырёхлетнего интервала (1-1461)

t – количество секунд от начала текущих суток

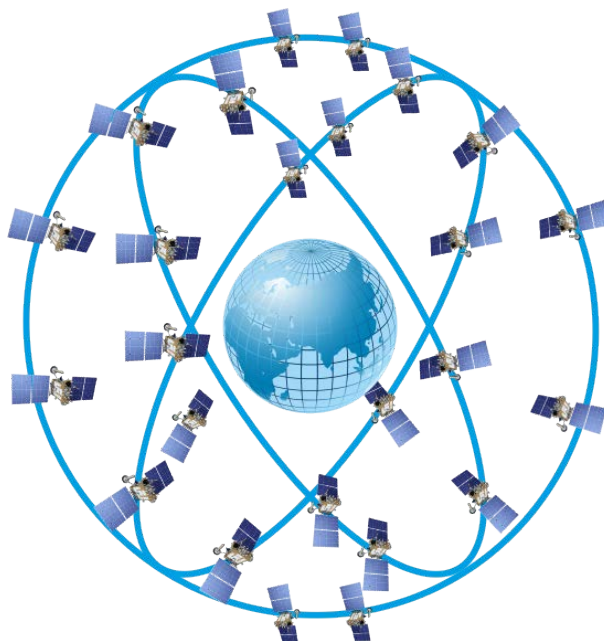


Рисунок 1.1.1 – Расположение спутников Глонасс

1.2 ОПИСАНИЕ ГНСС GPS

GPS(Global Positioning System) - американская спутниковая радионавигационная система, которая передает как гражданские, так и

военные сигналы, которые доступны в любой точке земного шара, в любое время, так как сигналы передаются непрерывно.

Высота орбиты составляет 20 180 км, орбиты являются синхронными, так как период обращения равен 11 ч. 58 мин..

Спутниковая группировка разбита на шесть плоскостей, на которых расположено 32 основных КА, дополнительно на некоторых орбитах может находиться резервный КА

Геоцентрическая система координат WGS 84.

Системное время связано со шкалой времени UTC и оно разрывное.

Наклоны орбитальных плоскостей 55 гр.

начало отсчёта – 00 ч 00 мин 06.01.1980

Формат времени: WN: TOW

WN – номер недели, который начинается с 6 января 1980 г., по модулю 1024

TOW – количество секунд от начала текущей недели

Система имеет кодовое разделение сигналов, то есть каждый спутник излучает сигналы на тех же частотах, что и остальные

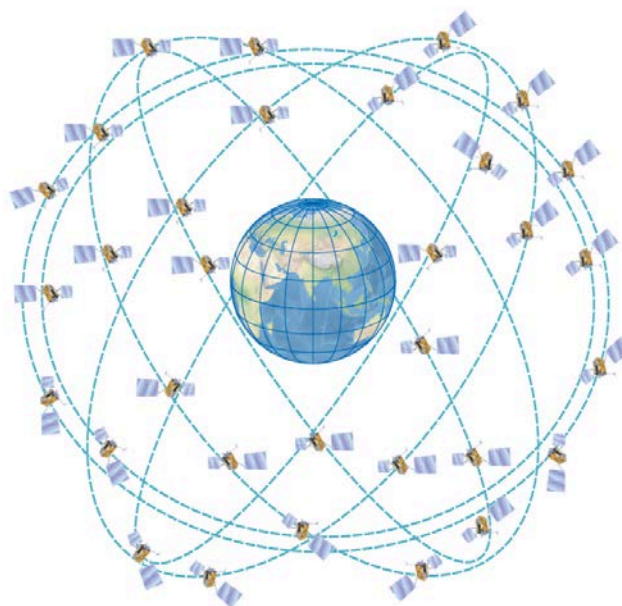


Рисунок 1.2.1 – Расположение спутников GPS

1.3 ОПИСАНИЕ ГНСС GALILEO

Galileo - созданная Европейским Союзом спутниковая радионавигационная система, которая передает гражданские сигналы, которые доступны в любой точке земного шара, в любое время, так как сигналы передаются непрерывно, также может являться взаимодоплнением с системой GPS третьего поколения.

Высота орбиты составляет 23 222 км

Наклонение орбитальных плоскостей 56 гр., период 14 ч 4 мин 45 с.

Спутниковая группировка разбита на три плоскости, на которых расположено 27 основных КА, дополнительно на некоторых орбитах может находиться резервный КА, на данный момент всего 3 резервных КА.

Геоцентрическая система координат GTRF.

Шкала времени системы Галилео (Galileo System Time – GST) – непрерывная атомная шкала времени с постоянным смещением на целое количество секунд относительно международного атомного времени TAI. Со

шкалой времени UTC шкала GST имеет переменное расхождение на целое количество секунд.

Информация о величине расхождения шкалы времени GST относительно шкал TAI и UTC включена в навигационное сообщение для передачи потребителям.

Начало отсчета 22 августа 1999 года - время, когда неделя GPS достигла значения 1 024.

Формат времени: WN: TOW

WN – номер недели, который начинается с 22 августа 1999 г., по модулю 4096

TOW – количество секунд от начала текущей недели

Система имеет кодовое разделение сигналов, то есть каждый спутник излучает сигналы на тех же частотах, что и остальные

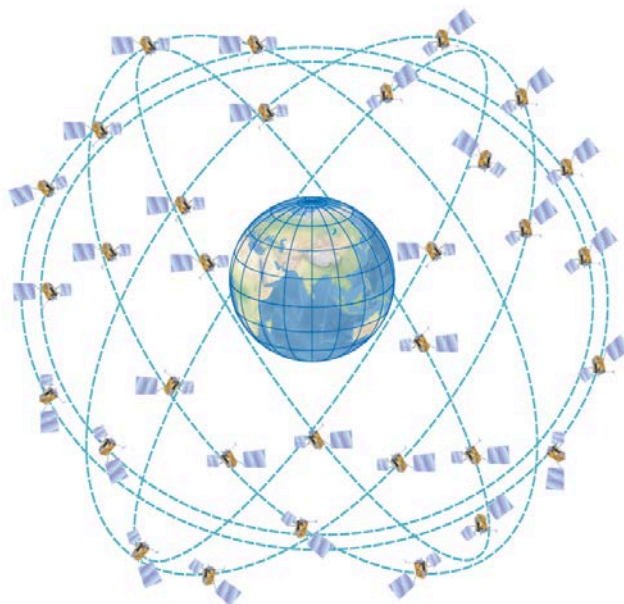


Рисунок 1.3.1 – Расположение спутников Galileo

1.4 ОПИСАНИЕ ГНСС BEIDOU

Beidou- китайская спутниковая радионавигационная система, которая передает как гражданские, так и военные сигналы, которые доступны в любой точке земного шара, в любое время, так как сигналы передаются непрерывно,

Орбитальная группировка разбита на 3 части:

- 1 На геостационарной орбите 3 КА, высота орбиты составляет 35 786 км
- 2 На средних круговых орбитах 24 КА, разбита на 3 плоскости, высота орбиты составляет 21 528 км, наклонение 55 гр..
- 3 На наклонной геосинхронной орбите 3 КА, высота орбиты 35 786 км. наклонение 55 гр..

Геоцентрическая система координат BDCS

Параметры движения КА БЭЙДОУ передаются в китайской геодезической системе координат 2000 года.

В качестве шкалы времени системы определено китайское универсальное глобальное время, которое поддерживается атомными стандартами частоты, установленными в центре управления в Пекине. Время системы БЭЙДОУ (BDT) связано с координированным всемирным временем (UTC) и синхронизировано с ним с точностью 100 нс. Началом отсчёта BDT является 0 ч 0 мин 1 января 2006 года (UTC). По заявлению создателей системы, предусмотрена совместимость BDT со временем GPS/Galileo. Расхождение между шкалами времени GPST/GST измеряется и передаётся.

Формат времени: WN : SOW

WN – номер недели, который начинается с 1 января 2006 г., по модулю 8192

SOW – количество секунд от начала текущей недели

Система имеет кодовое разделение сигналов, то есть каждый спутник излучает сигналы на тех же частотах, что и остальные

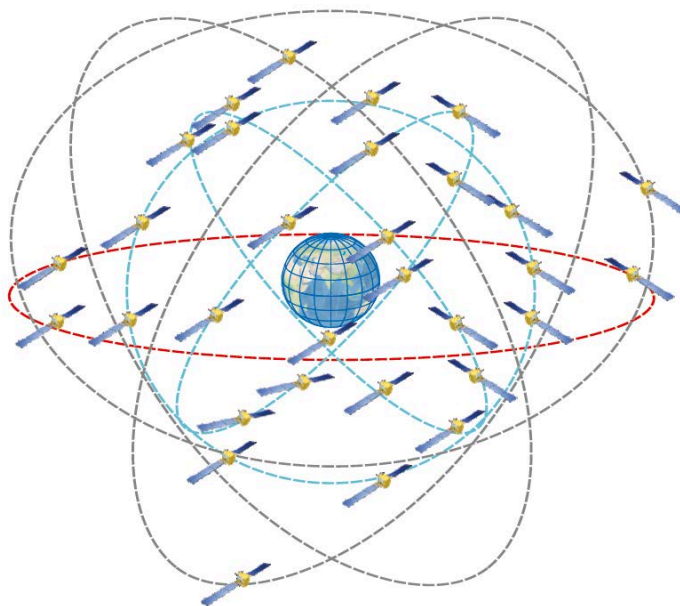


Рисунок 1.4.1 – Расположение спутников Beidou

2 ОПРЕДЕЛЕНИЯ ДАЛЬНОСТИ ОТ ПОТРЕБИТЕЛЯ ДО СПУТНИКА ИЛИ КООРДИНАТ ПОТРЕБИТЕЛЯ

Определения координат потребителя складывается в решение задачи из как минимум 4-х псевдодальномерных уравнений, при решении которых получаются нужные оценки координат.

2.1 ПСЕВДОДАЛЬНОМЕРНЫЙ МЕТОД

Псевдодальномерный метод или метод определения координат объекта по измерениям квазидальности рассмотрен в работе.

Под псевдодальностью от i -го НС до потребителя понимают измеренную дальность $D_{изм\ i}$ до этого НС, отличающуюся от истинной дальности D_i на неизвестную, но постоянную за время определения навигационных параметров величину D' . Таким образом, псевдо дальность до i -го НС

$$D_{изм\ i} = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} + D' \quad (1.1)$$

В псевдодальномерных методах, основанных на измерениях псевдодальностей, в качестве навигационного параметра выступает $D_{изм\ i}$. Поверхностью положения ПО по-прежнему является сфера с центром в точке центра масс НС, но радиус этой сферы изменен на неизвестную величину D' . Измерение псевдодальностей до трех НС приводит к системе трех уравнений с четырьмя неизвестными (x, y, z, D'). В решении этой системы уравнений возникает неопределенный параметр, и для устранения возникшей неопределенности необходимо провести дополнительное измерение, т. е. измерить псевдодальность до четвертого спутника. Полученная таким образом система четырех уравнений имеет точное решение, и следовательно, местоположение потребителя при измерениях псевдодальностей определяется как точка пересечения четырех поверхностей положения.

Необходимость нахождения в зоне видимости четырех НС предъявляет достаточно жесткие требования к структуре сети НС, которые выполняются только в среднеорбитальных СРНС.

Важно отметить, что если геоцентрическая высота объекта априорно известна, то число минимально необходимых для решения навигационной задачи НС сокращается на один (с четырех до трех). Используя изложенные в [2] приемы, нетрудно получить алгоритмы решения навигационных задач в конечном виде для подвижного объекта с известной высотой. Однако следует подчеркнуть, что для априорного вычисления геоцентрической высоты требуется знать, в частности, земной радиус-вектор, который является функцией широты места. В этом случае навигационную задачу можно решить с высокой точностью лишь путем последовательных приближений.

Псевдодальномерный метод не накладывает жестких ограничений на значение погрешности $D' = ct'$ (погрешности временной шкалы) и позволяет одновременно с определением местоположения вычислять отклонение шкалы времени потребителя, однако для его реализации необходимо принимать сигналы от 4 НС. Но данный метод обладает более высокой точностью, по сравнению с дальномерным. [1]

2.2 ИСТОЧНИКИ ОШИБОК, ВЛИЯЮЩИХ НА ТОЧНОСТЬ НАВИГАЦИОННО-ВРЕМЕННЫХ ОПРЕДЕЛЕНИЙ

Навигационно-временные определения принимают формирования оценок координат, составляющих скорости потребителя и текущего времени в результате обработки радиосигналов, принимаемых от навигационных систем.

Точность определяется двумя типами погрешностей:

- Погрешности, возникающие на этапе первичной обработки
- Погрешности, возникающие на этапе вторичной обработки.

На этапе первичной обработки формируются оценки псевдодальностей и псевдоскоростей, соответственно, погрешности, возникающие на этом этапе – погрешности определения псевдодальностей и псевдоскоростей.

На этапе вторичной обработки оценки псевдодальностей и псевдоскоростей пересчитываются в оценки псевдодальностей и псевдоскоростей пересчитываются в оценки координат потребителя, поэтому погрешности этого этапа определяются факторами, влияющими на эффективности такого пересчета.

Приведем таблицу отдельных составляющих общей погрешности, которую называют бюджетом погрешностей. В таблице 1 предполагается, что все составляющие погрешности являются некоррелированными между собой случайными величинами с гауссовским законом распределения, имеющими нулевое значение математического ожидания и дисперсию $\sigma_{\delta Д}^2$. Поэтому дисперсия общей ошибки определяется как сумма дисперсий отдельных составляющих. Приводимые в таблице значения отдельных составляющих соответствуют остаточным значения погрешностей, то есть предполагается, что в приемнике приняты меры по снижению погрешностей определения псевдодальности и псевдоскорости, погрешность, вносимая приемником,

дается для автономно работающего с одностационарного (неподвижного) приемника с двухэтапной обработкой сигналов, при приеме сигнала от НС, находящегося в зените. Таким образом, учитывая данные замечания, величина итоговой погрешности имеет больше смысл как потенциальная, то есть предельно достижимая в реальной аппаратуре.

Бюджет погрешностей определения псевдодальности

Источник погрешности	$\sigma_{\delta d}, м$
Часы спутника	2
Ионосфера	4
Тропосфера	0.5
Многолучевость	2.5
Приемник	1.5
Прочие	2
Суммарная погрешность	5.7

Бюджет погрешностей определения псевдоскорости

Источник погрешности	$\sigma_{\delta v}, м / с$
Бортовой ОГ	0.04
Релятивистский эффект	0.001
Многолучевость	0.01
Приемник	0.08
Прочие	0.01
Суммарная погрешность	0.09

2.3 ОЦЕНКА КООРДИНАТ ПОТРЕБИТЕЛЯ С УЧЕТОМ ОШИБОК SISRE

SISRE - эквивалентная погрешность псевдодальности за счёт космического сегмента (signal-in-space range error). Эта величина характеризует погрешность измерения дальности до космического аппарата, обусловленную влиянием так называемых «системных» факторов, не зависящих от внешней среды и характеристик оборудования потребителей.

Для системы ГЛОНАСС эту величину можно наблюдать графически (рис.1.1) на официальном сайте [3].

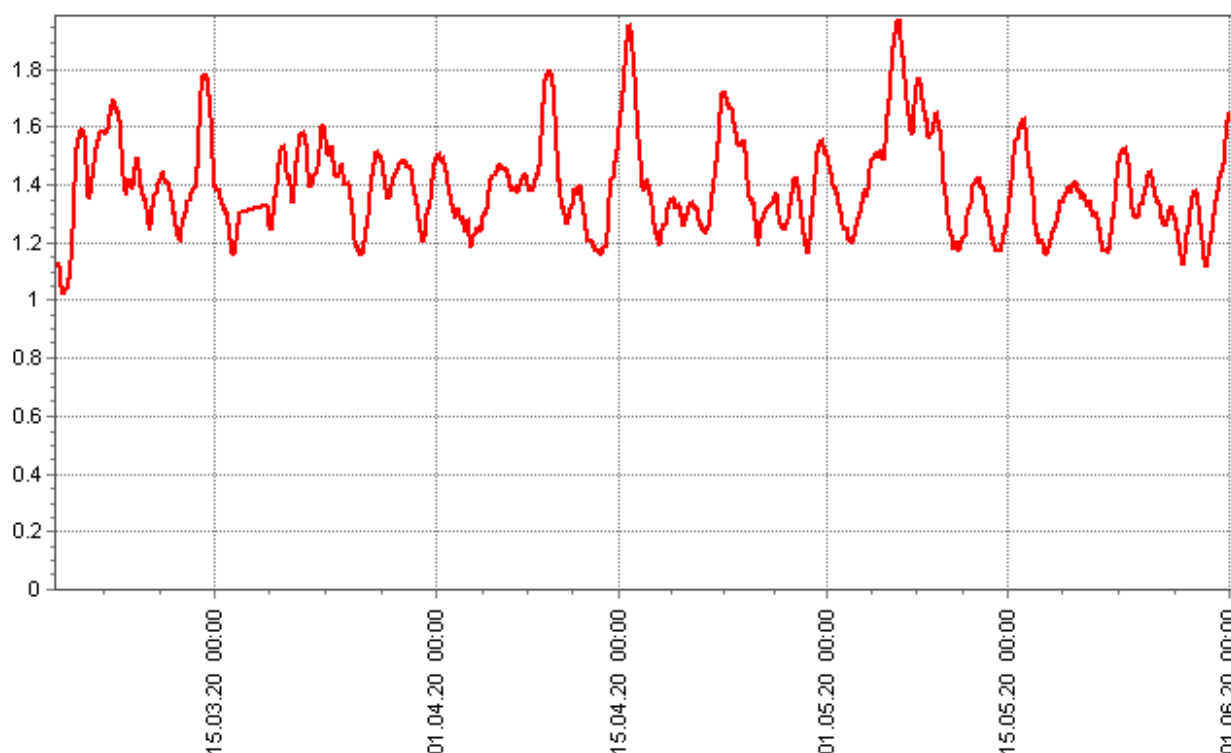


Рисунок 2.1 - Суточные СКП SISRE ГЛОНАСС, м

Оценка функциональных характеристик ГЛОНАСС и GPS проводится в ИАЦ КВНО в режиме регулярной службы на основе измерений глобальной сети станций с помощью собственных программных средств.

Так же данные SISRE можно получить на сервере информационно-аналитического центра координатно-временного и навигационного обеспечения [4].

3 ПОСТАНОВКА ЗАДАЧИ НАХОЖДЕНИЯ ОЦЕНКИ КООРДИНАТЫ ПОТРЕБИТЕЛЯ С УЧЕТОМ ОШИБОК SISRE

Постановка задачи: Оценить координаты потребителя $\{x, y, z\}$ с учетом ошибок SISRE

Введем вектор состояния $\mathbf{x} = \begin{bmatrix} x & y & z & \mathcal{D} \end{bmatrix}^T$, вектор вторичных наблюдений

$$\mathbf{y}_{\tilde{\mathbf{R}},k} = \begin{bmatrix} y_{\tilde{R}_1,k} & y_{\tilde{R}_2,k} & \dots & y_{\tilde{R}_N,k} \end{bmatrix}^T \quad \text{и} \quad \text{вектор погрешностей}$$

$$\mathbf{n}_{\tilde{\mathbf{R}},k} = \begin{bmatrix} n_{\tilde{R}_1,k} & n_{\tilde{R}_2,k} & \dots & n_{\tilde{R}_N,k} \end{bmatrix}^T$$

Линеаризуем вторичные наблюдения $y_{\tilde{R},k} = R_i(x_k, y_k, z_k) + \mathcal{D}' + n_{\tilde{R},k}$ относительно некоторой априорной оценки $\tilde{\mathbf{x}}_k = \begin{bmatrix} \tilde{x}_k & \tilde{y}_k & \tilde{z}_k & \tilde{\mathcal{D}}_k \end{bmatrix}^T$, тогда:

$$y_{\tilde{R},k} = h_{i,k}(\tilde{\mathbf{x}}_k) + \tilde{H}(\tilde{\mathbf{x}}_k)(\mathbf{x}_k - \tilde{\mathbf{x}}_k) + \mathbf{n}_{\tilde{R},k} \quad (1.2)$$

Выполним преобразования [1, 204 стр.] тогда получим выражение:

$$\Delta \hat{\mathbf{x}}_k = (\tilde{\mathbf{H}}^T \mathbf{D}_{\mathbf{n}_R}^{-1} \tilde{\mathbf{H}})^{-1} \tilde{\mathbf{H}}^T \mathbf{D}_{\mathbf{n}_R}^{-1} \Delta \mathbf{y}_{\tilde{\mathbf{R}},k} \quad (1.3)$$

Где $\mathbf{D}_{\mathbf{n}_R} = M[\mathbf{n}_{\tilde{\mathbf{R}},k} \mathbf{n}_{\tilde{\mathbf{R}},k}^T]$ введенная матрица дисперсий погрешностей вторичных наблюдений.

На данный этап приближенно будем считать $\sigma_n^2 = SISRE_n^2$

Тогда преобразовав матрицу дисперсий и рассмотрим для случая при $k=1$, она примет вид:

$$\mathbf{D}_{\mathbf{n}_R} = M[\mathbf{n}_{\tilde{\mathbf{R}},1} \mathbf{n}_{\tilde{\mathbf{R}},1}^T] = \begin{bmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \sigma_n^2 \end{bmatrix}$$

Необходимо найти матрицу дисперсий вектора состояния \mathbf{x} , которая по определению равна $\mathbf{D}_x = M[\Delta \hat{\mathbf{x}} \cdot (\Delta \hat{\mathbf{x}})^T]$, подставим формулу (1.3), получим:

$$\mathbf{D}_x = M[\Delta \hat{\mathbf{x}} \cdot (\Delta \hat{\mathbf{x}})^T] = (\tilde{\mathbf{H}}^T \mathbf{D}_{n_R}^{-1} \tilde{\mathbf{H}})^{-1} \tilde{\mathbf{H}}^T \mathbf{D}_{n_R}^{-1} \Delta \mathbf{y}_R \cdot \left((\tilde{\mathbf{H}}^T \mathbf{D}_{n_R}^{-1} \tilde{\mathbf{H}})^{-1} \right)^T \tilde{\mathbf{H}} (\mathbf{D}_{n_R}^{-1})^T \Delta \mathbf{y}_R^T$$

Учитывая, что $\mathbf{D}_{n_R} = M[\mathbf{n}_R \mathbf{n}_R^T] = M[\Delta \mathbf{y}_R \cdot (\Delta \mathbf{y}_R)^T]$, тогда матрица дисперсий вектора состояния \mathbf{x} равна:

$$\begin{aligned} \mathbf{D}_x &= (\tilde{\mathbf{H}}^T \mathbf{D}_{n_R}^{-1} \tilde{\mathbf{H}})^{-1} \tilde{\mathbf{H}}^T \mathbf{D}_{n_R}^{-1} \cdot \left((\tilde{\mathbf{H}}^T \mathbf{D}_{n_R}^{-1} \tilde{\mathbf{H}})^{-1} \right)^T \tilde{\mathbf{H}} (\mathbf{D}_{n_R}^{-1})^T \cdot \mathbf{D}_{n_R} = \\ &= \left((\tilde{\mathbf{H}}^T \mathbf{D}_{n_R}^{-1} \tilde{\mathbf{H}})^{-1} \right)^T (\mathbf{D}_{n_R}^{-1})^T \cdot \mathbf{D}_{n_R} = \left((\tilde{\mathbf{H}}^T \mathbf{D}_{n_R}^{-1} \tilde{\mathbf{H}})^{-1} \right)^T \end{aligned}$$

Получаем, что на диагонали \mathbf{D}_x будут лежать дисперсии компонент вектора состояний \mathbf{x}

4 РЕШЕНИЯ ЗАДАЧИ НАХОЖДЕНИЯ ОЦЕНКИ КООРДИНАТЫ ПОТРЕБИТЕЛЯ С УЧЕТОМ ОШИБОК SISRE

Из пункта 3.1 следует, что решением данной задачи является ковариационная матрица \mathbf{D}_x , на диагонали которой будут лежать дисперсии компонент вектора состояний \mathbf{x} .

И матрица принимает вид: $\mathbf{D}_x = \left((\tilde{\mathbf{H}}^T \mathbf{D}_{n_R}^{-1} \tilde{\mathbf{H}})^{-1} \right)^T$

Для решение данной задачи необходимо найти матрицы \mathbf{H} и \mathbf{D}_n :

- Матрица \mathbf{H} принимает вид:

$$\mathbf{H}_{n,1...4} = \left| \begin{array}{ccc} \frac{(x_n - x)}{R_n} & \frac{(y_n - y)}{R_n} & \frac{(z_n - z)}{R_n} \\ & & 1 \end{array} \right| \quad (1.4)$$

где:

$$R_n = \sqrt{(x_n - x)^2 + (y_n - y)^2 + (z_n - z)^2} \quad (1.5)$$

- Матрица \mathbf{D}_n принимает вид:

$$\mathbf{D}_n = \begin{vmatrix} D_1 & 0 & 0 & 0 \\ 0 & D_2 & 0 & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & D_n \end{vmatrix} \quad (1.6)$$

где n зависит от количества видимых спутников

После полученных значений матрица \mathbf{D}_x примет вид:

$$\mathbf{D}_x = \left((\tilde{\mathbf{H}}^T \mathbf{D}_{n_r}^{-1} \tilde{\mathbf{H}})^{-1} \right)^T = \begin{vmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} & \sigma_{14} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} & \sigma_{24} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} & \sigma_{34} \\ \sigma_{41} & \sigma_{42} & \sigma_{43} & \sigma_{44} \end{vmatrix} \quad (1.7)$$

где:

$$\begin{aligned} \sigma_{11}^{-1} &= \frac{H_{1,1}^2}{D_1} + \frac{H_{2,1}^2}{D_2} + \dots + \frac{H_{n,1}^2}{D_n} \\ \sigma_{22}^{-1} &= \frac{H_{1,2}^2}{D_1} + \frac{H_{2,2}^2}{D_2} + \dots + \frac{H_{n,2}^2}{D_n} \\ \sigma_{33}^{-1} &= \frac{H_{1,3}^2}{D_1} + \frac{H_{2,3}^2}{D_2} + \dots + \frac{H_{n,3}^2}{D_n} \\ \sigma_{44}^{-1} &= \frac{H_{1,4}^2}{D_1} + \frac{H_{2,4}^2}{D_2} + \dots + \frac{H_{n,4}^2}{D_n} \end{aligned}$$

Развернутое вычисление матрицы \mathbf{D}_x в приложении 2.

На диагонали лежат дисперсии компонент вектора состояний \mathbf{x} .

5 АЛГОРИТМЫ ОПРЕДЕЛЕНИЯ КООРДИНАТ

5.1 АЛГОРИТМ ДЛЯ GPS

Алгоритм для расчета значений, представленных в ИКД GPS:

5.1.1 Определим время, отсчитываемое от опорной эпохи эфемерид:

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

5.1.2 Определим среднее движение:

$$n_0 = \sqrt{\frac{\mu}{A_0^3}}$$

5.1.3 Определим скорректированное среднее движение:

$$n_A = n_0 + \Delta n$$

5.1.4 Определим среднюю аномалию:

$$M_k = M_0 + n_A \cdot t_k$$

5.1.5 Решим уравнение Кеплера минимум 3-мя итерациями и определим

E_k :

$$M_k = E_k - e_n \cdot \sin(E_k) \Rightarrow E_k = M_k + e_n \cdot \sin(E_k)$$

5.1.6 Определим истинную аномалию:

$$v_k = \arctg \left(\frac{\sqrt{1-e_n^2} \sin(E_k)}{(\cos(E_k) - e_n)} \right)$$

5.1.7 Определим скорректированный радиус орбиты спутника:

$$A_k = A_0 + \left(\dot{A} \right) t_k$$

$$r_k = A_k (1 - e_n \cos(E_k)) + \delta r_k$$

5.1.8 Определим аргумент широты:

$$\Phi_k = v_k + \omega$$

$$u_k = \Phi_k + \delta u_k$$

5.1.9 Определим координаты НС в орбитальной плоскости:

$$\begin{cases} x'_k = r_k \cdot \cos(u_k) \\ y'_k = r_k \cdot \sin(u_k) \end{cases}$$

5.1.10 Определим скорректированную долготу восходящего узла Ω_k определяется из соотношения:

$$\dot{\Omega} = \dot{\Omega}_{REF} + \cancel{\Delta\dot{\Omega}}$$

$$\Omega_k = \Omega_{0-n} + \left(\dot{\Omega} - \dot{\Omega}_e \right) t_{oe} - \dot{\Omega}_e t_{oe}$$

5.1.11 Определим скорректированное наклонение орбиты спутника

$$i_k = i_{0-n} + \cancel{(i_{0-n} - DOT)t_k} + \cancel{\delta u_k}$$

5.1.12 Определим координаты НС в геоцентрической системе координат:

$$\begin{cases} x_k = x'_k \cos \Omega_k - y'_k \cos i_k \sin \Omega_k \\ y_k = x'_k \sin \Omega_k + y'_k \cos i_k \cos \Omega_k \\ z_k = y'_k \sin i_k \end{cases}$$

5.2 АЛГОРИТМ РАСЧЕТА КООРДИНАТ ГЛОНАСС

Алгоритм для расчета значений, представленных в ИКД ГЛОНАСС:

5.1.1 Определяется интервал прогноза в секундах:

$$\Delta t_{\text{пр}} = \Delta N_A \cdot 86400 + (t_i - t_{\lambda_A}),$$

$$\Delta N_A = \begin{cases} N - N_A - \left\langle \left\langle \frac{N - N_A}{1461} \right\rangle \right\rangle \cdot 1461 & \text{если } N_4 \neq 27, \\ N - N_A - \left\langle \left\langle \frac{N - N_A}{1460} \right\rangle \right\rangle \cdot 1460 & \text{если } N_4 = 27; \end{cases}$$

Где:

N – календарный номер суток внутри четырехлетнего периода, начиная с високосного года, на которых находится заданный момент времени t_i в секундах по шкале МДВ;

N_A – календарный номер суток по шкале МДВ внутри четырехлетнего интервала, передаваемый НКА в составе неоперативной информации;

$\left\langle \left\langle x \right\rangle \right\rangle$ – вычисление целого, ближайшего к x .

5.1.2 Рассчитывается количество целых витков W на интервале прогноза:

$$W = \left\langle \frac{\Delta t_{\text{пр}}}{T_{\text{ср}} + \Delta T_A} \right\rangle,$$

где $\langle x \rangle$ выделение целой части x ;

5.1.3 Определяется текущее наклонение:

$$i = \left(\frac{i_{\text{ср}}}{180^\circ} + \Delta i_A \right) \cdot \pi \text{ рад},$$

5.1.4 Определяются средний драконический период на витке $W+1$ и среднее движение:

$$T_{\text{др}} = T_{\text{ср}} + \Delta T_A + (2W + 1) \cdot \Delta \dot{T}_A,$$

$$n = 2\pi / T_{\text{др}},$$

5.1.5 Методом последовательных приближений $m = 0, 1, 2 \dots$ рассчитывается большая полуось орбиты a :

$$a^{(m+1)} = \sqrt[3]{\left(\frac{T_{\text{оск}}^{(m)}}{2\pi}\right)^2 \cdot GM};$$

$$p^{(m+1)} = a^{(m+1)} (1 - (\varepsilon_A)^2);$$

$$T_{\text{оск}}^{(m+1)} = \frac{T_{\text{др}}}{1 - \frac{3}{2} \cdot J_2^0 \left(\frac{a_e}{p^{(m+1)}}\right)^2 \left[\left(2 - \frac{5}{2} \cdot \sin^2 i\right) \cdot \frac{(1 - (\varepsilon_A)^2)^{3/2}}{(1 + \varepsilon_A \cdot \cos(\omega_A \pi))^2} + \frac{(1 + \varepsilon_A \cdot \cos(\omega_A \pi))^3}{1 - (\varepsilon_A)^2} \right]}},$$

5.1.6 Определяются текущие значения долготы восходящего узла орбиты и аргумента перигея с учетом их векового движения под влиянием сжатия Земли:

$$\lambda = \lambda_A \cdot \pi - \left\{ \omega_3 + \frac{3}{2} J_2^0 \cdot n \cdot \left(\frac{a_e}{p}\right)^2 \cos i \right\} \Delta t_{\text{пр}};$$

$$\omega = \omega_A \cdot \pi - \frac{3}{4} J_2^0 n \left(\frac{a_e}{p}\right)^2 (1 - 5 \cos^2 i) \cdot \Delta t_{\text{пр}},$$

5.1.7 Рассчитывается значение средней долготы на момент прохождения текущего восходящего узла:

$$L_1 = \omega + E_0 - \varepsilon_A \sin E_0,$$

$$E_0 = -2 \cdot a \tan \left(\sqrt{\frac{1 - \varepsilon_A}{1 + \varepsilon_A}} \cdot \tan \frac{\omega}{2} \right).$$

Где

5.1.8 Определяется текущее значение средней долготы НКА:

$$L = L_1 + n(\Delta t_{\text{np}} - (T_{\text{cp}} + \Delta T_A)W - \Delta \dot{T}_A W^2).$$

5.1.9 Определяется эксцентрическая аномалия путем решения уравнения Кеплера

$$L - \omega = E - \varepsilon \cdot \sin E.$$

Как правило, используется схема последовательных приближений $m = 0, 1, 2$, и т.д.:

$$E^{(m+1)} = L - \omega + \varepsilon \cdot \sin E^{(m)},$$

5.1.10 Вычисляются истинная аномалия и аргумент широты НКА u :

$$v = 2 \arctan \left(\sqrt{\frac{1 + \varepsilon_A}{1 - \varepsilon_A}} \tan \frac{E}{2} \right);$$

$$u = v + \omega.$$

5.1.11 Рассчитываются координаты центра масс НКА в геоцентрической прямоугольной пространственной системе координат:

$$p = a(1 - (\varepsilon_A)^2);$$

$$r = \frac{p}{1 + \varepsilon_A \cos v};$$

$$x(t_i) = r(\cos \lambda \cos u - \sin \lambda \sin u \cos i);$$

$$y(t_i) = r(\sin \lambda \cos u + \cos \lambda \sin u \cos i);$$

$$z(t_i) = r \sin u \sin i.$$

5.3 АЛГОРИТМ РАСЧЕТА КООРДИНАТ GALILEO

Алгоритм для расчета значений, представленных в ИКД GALILEO, за основу расчета берется алгоритм GPS с поправками:

Следует учесть, что в альманахе Galileo, дано значение разницы A между квадратным корнем из большой полуоси и квадратным корнем из номинальной большой полуоси, поэтому:

Большая полуось определяется, как:

$$\Delta(A^{1/2}) = A^{1/2} - A_{ном}^{1/2}$$

Где: $A_{ном} = 29600000 \text{ м}$ - номинальное значение полуоси орбиты

А также следует учесть, что значение номера недели задан по модулю 4,
Определим время, отсчитываемое от опорной эпохи эфемерид:

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

5.1.13 Определим среднее движение:

$$n_0 = \sqrt{\frac{\mu}{A_0^3}}$$

5.1.14 Определим скорректированное среднее движение:

$$n_A = n_0 + \Delta n$$

5.1.15 Определим среднюю аномалию:

$$M_k = M_0 + n_A \cdot t_k$$

5.1.16 Решим уравнение Кеплера минимум 3-мя итерациями и определим E_k :

$$M_k = E_k - e_n \cdot \sin(E_k) \Rightarrow E_k = M_k + e_n \cdot \sin(E_k)$$

5.1.17 Определим истинную аномалию:

$$v_k = \arctg\left(\frac{\sqrt{1-e_n^2} \sin(E_k)}{(\cos(E_k) - e_n)}\right)$$

5.1.18 Определим скорректированный радиус орбиты спутника:

$$A_k = A_0 + \left(\dot{A} \right) t_k$$

$$r_k = A_k (1 - e_n \cos(E_k)) + \delta r_k$$

5.1.19 Определим аргумент широты:

$$\Phi_k = v_k + \omega$$

$$u_k = \Phi_k + \delta u_k$$

5.1.20 Определим координаты НС в орбитальной плоскости:

$$\begin{cases} x'_k = r_k \cdot \cos(u_k) \\ y'_k = r_k \cdot \sin(u_k) \end{cases}$$

5.1.21 Определим скорректированную долготу восходящего узла Ω_k определяется из соотношения:

$$\dot{\Omega} = \dot{\Omega}_{REF} + \Delta \dot{\Omega}$$

$$\Omega_k = \Omega_{0-n} + \left(\dot{\Omega} - \dot{\Omega}_e \right) t_{oe}$$

5.1.22 Определим скорректированное наклонение орбиты спутника

$$i_k = i_{0-n} + (\dot{i}_{0-n} - DOT) t_k + \delta i_k$$

5.1.23 Определим координаты НС в геоцентрической системе координат:

$$\begin{cases} x_k = x'_k \cos \Omega_k - y'_k \cos i_k \sin \Omega_k \\ y_k = x'_k \sin \Omega_k + y'_k \cos i_k \cos \Omega_k \\ z_k = y'_k \sin i_k \end{cases}$$

6 РЕАЛИЗАЦИЯ НОВЫХ АЛГОРИТМОВ

Добавление новой формы, для добавления новых функций в программу (код в приложении 1):

- добавление сетки значений, для определения ско с шагом по высоте, широте и долготе;
- добавление случайно заданных значений координат и времени.

Интерфейс ПО

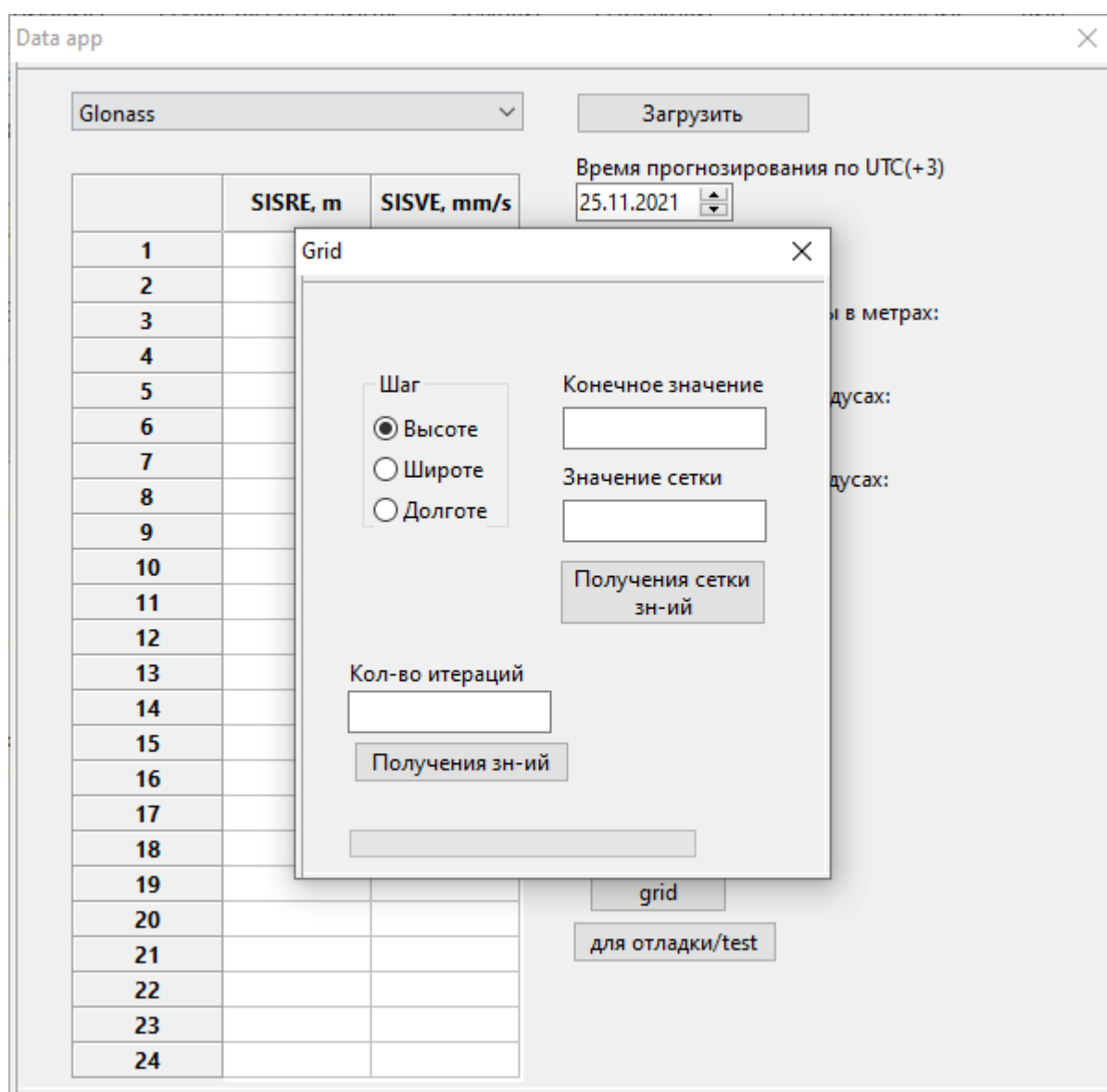


Рисунок 2.1 – Интерфейс новых функций

7 ПОСТРОЕНИЯ ГРАФИКОВ СО СЛУЧАЙНО ЗАДАННЫМИ КООРДИНАТАМИ И ВРЕМЕНЕМ

Добавим в алгоритм программы случайно заданные значения координат и времени, для этого воспользуемся функция «rand», в случае если при решение навигационной задачи матрица получится сингулярной, приравняем эти значения к -1, фактически это будет соответствовать значению СКО бесконечности

Графики для Глонасс

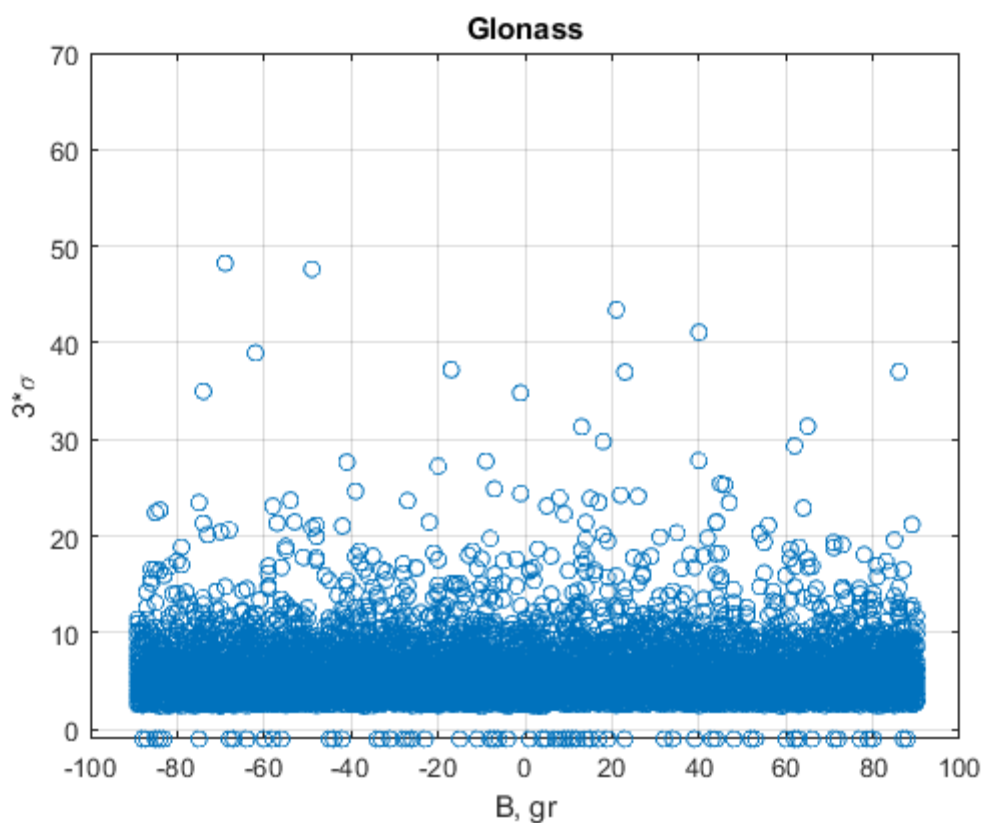


Рисунок 5.1.1 – График 3х ско от широты

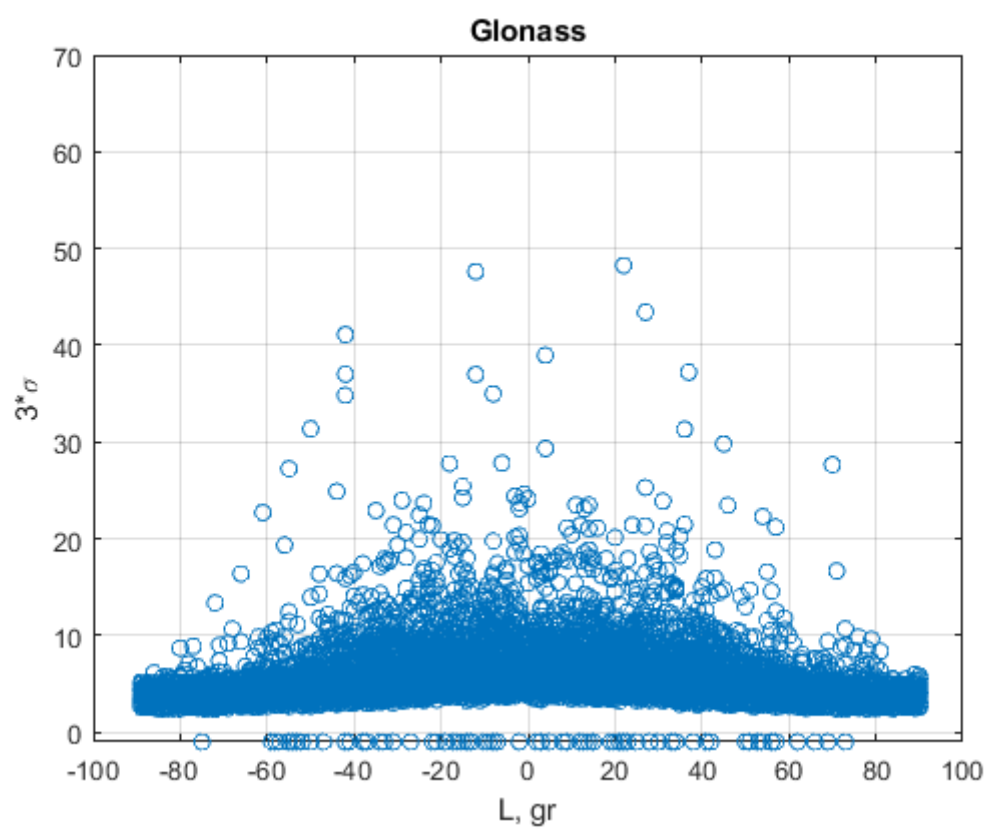


Рисунок 5.1.2 – График $3 \cdot 0$ от долготы

Для GPS

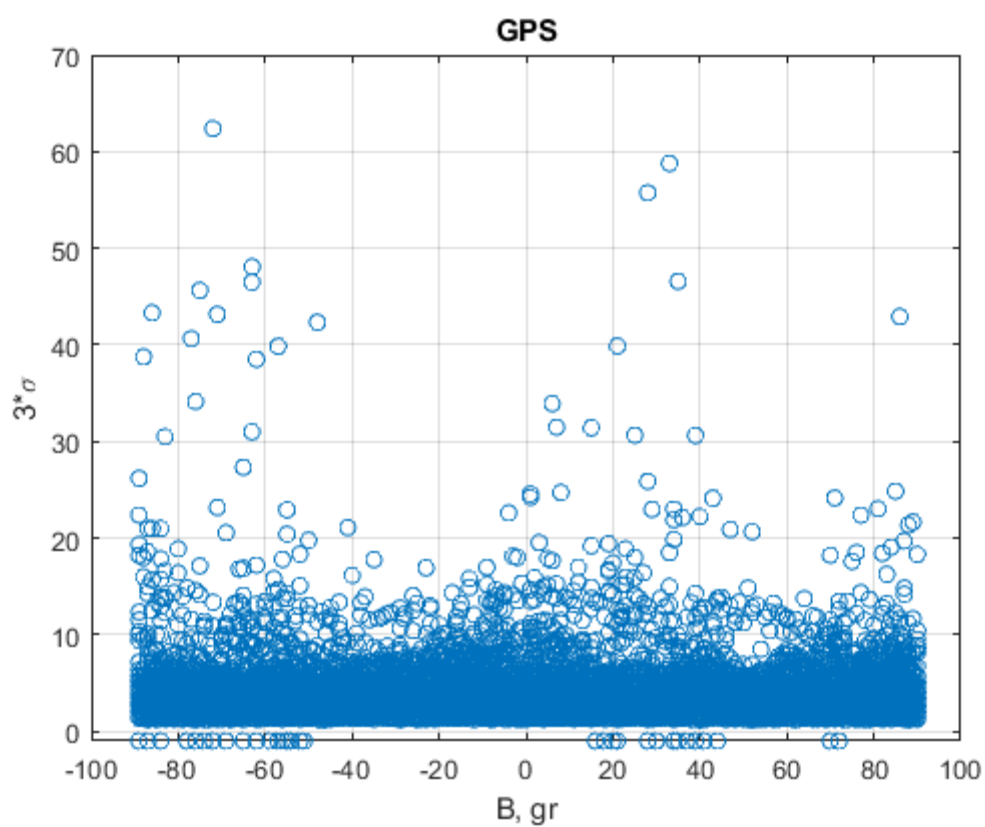


Рисунок 5.2.1 – График Z^*_0 ско от широты

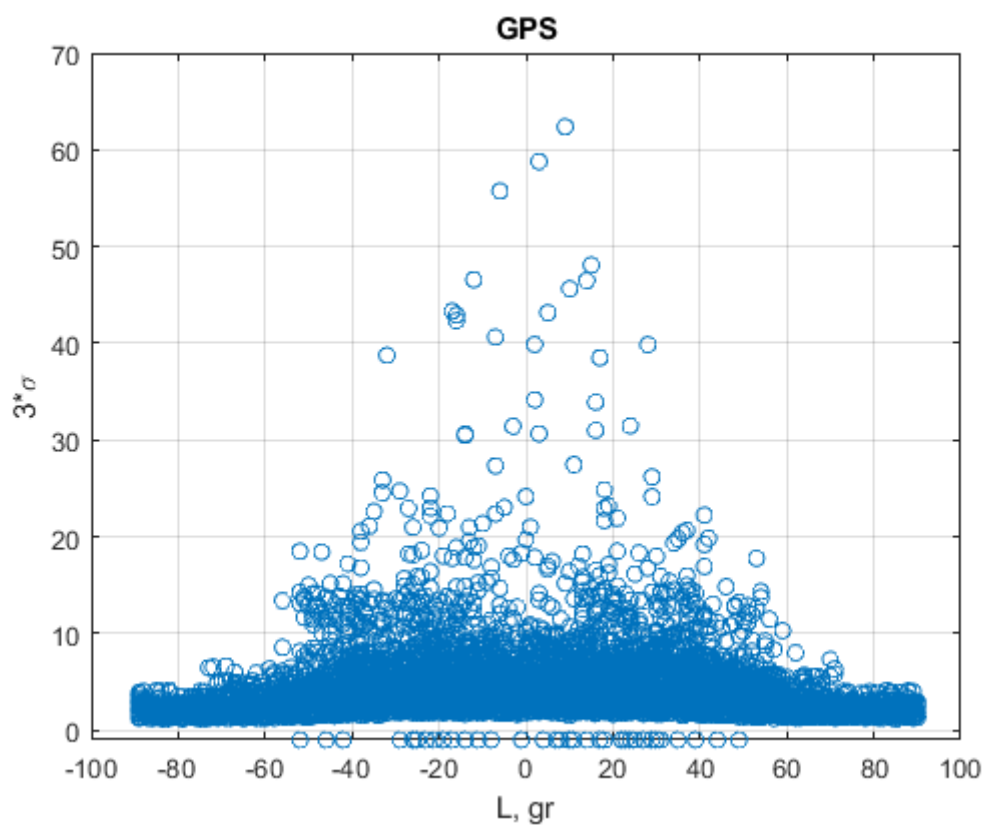


Рисунок 5.2.2 – График 3σ ско от долготы

Для Галилео

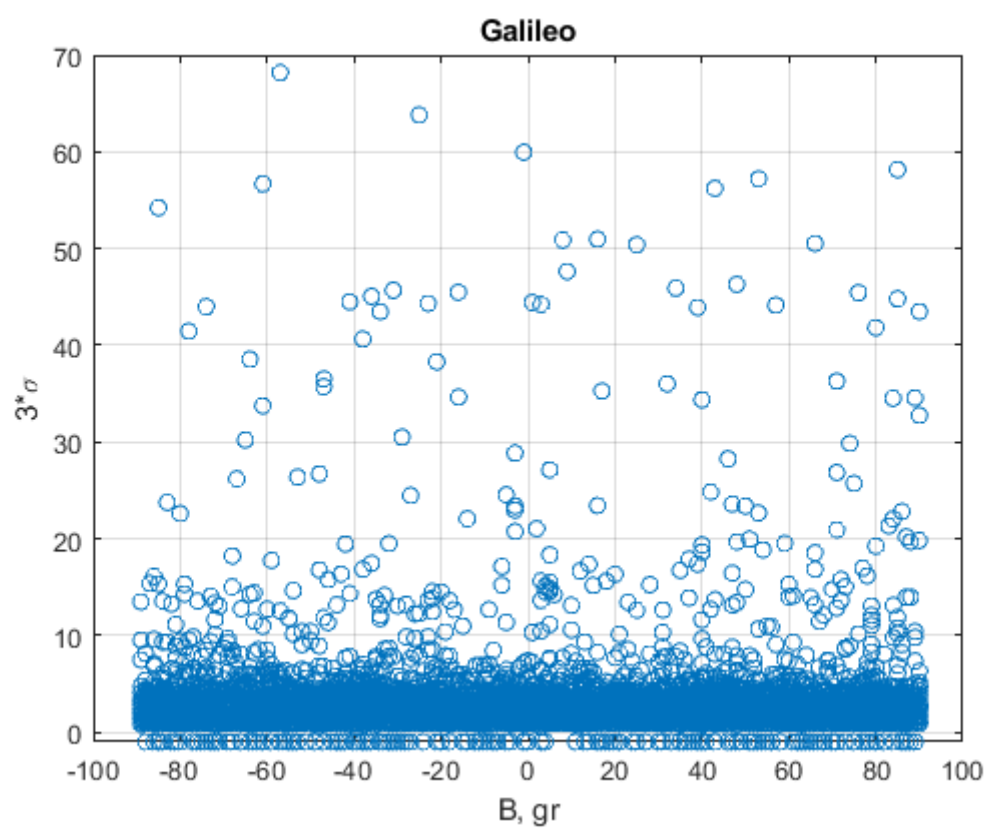


Рисунок 5.3.1 – График 3σ ско от широты

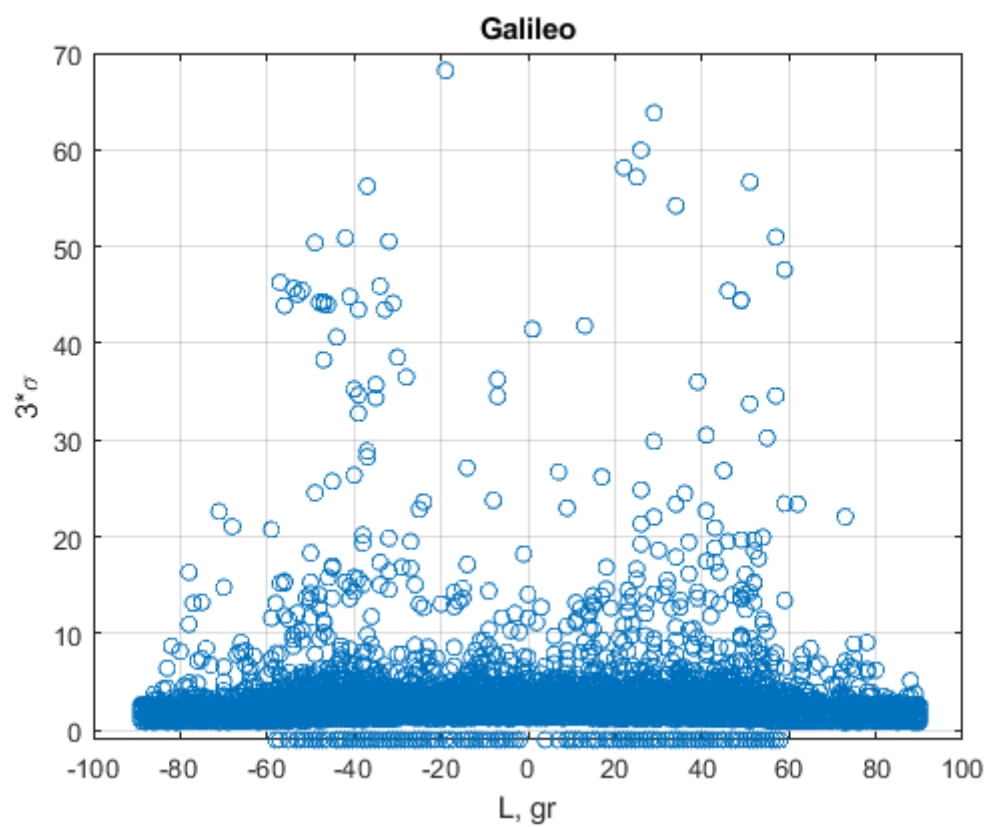


Рисунок 5.3.2 – График 3σ ско от долготы

На полученных графиках можем наблюдать распределение трех ско от широты и долготы

ЗАКЛЮЧЕНИЕ

Все поставленные задачи выполнены, добавленные новые функции в ПО такие как, получения СКО с шагом по высоте, долготе и широте, а также со случайно заданными значениями координат и времени.

Добавлены в алгоритм новые проверки во избежание ошибок при решении навигационной задачи – проверка на сингулярность матрицы.

Получил графики распределения СКО от широты и долготы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1]. Сервер «информационно-аналитического центра координатно-временного и навигационного обеспечения «<ftp://glonass-iac.ru>» »

[2]. Отчет НИР за 9 семестр

ПРИЛОЖЕНИЕ

Приложение 1

GridDialog.CPP

```
#include "GridDialog.h"
#include <array>
#include <iostream>
#include <string>
#include <stdio.h>
#include <fstream>
#include <wx/gdicmn.h>
#include <dataMain.h>
#include "timeCalc.h"
#include "calcGPS.h"
#include <calcGlonass.h>
#include <downlGalileo.h>
#include <calcGalileo.h>

#include "geoc2geod.h"

using namespace std;
//(*InternalHeaders(GridDialog)
#include <wx/intl.h>
#include <wx/string.h>
//*)

//(*IdInit(GridDialog)
const long GridDialog::ID_TEXTCTRL1 = wxNewId();
const long GridDialog::ID_STATICTEXT3 = wxNewId();
const long GridDialog::ID_TEXTCTRL7 = wxNewId();
```

```

const long GridDialog::ID_STATICTEXT1 = wxNewId();
const long GridDialog::ID_RADIOBOX1 = wxNewId();
const long GridDialog::ID_STATICTEXT2 = wxNewId();
const long GridDialog::ID_BUTTON1 = wxNewId();
const long GridDialog::ID_GAUGE1 = wxNewId();
const long GridDialog::ID_TEXTCTRL2 = wxNewId();
const long GridDialog::ID_STATICTEXT4 = wxNewId();
const long GridDialog::ID_BUTTON2 = wxNewId();
const long GridDialog::ID_SASHWINDOW1 = wxNewId();
//*)

```

```

BEGIN_EVENT_TABLE(GridDialog,wxDialog)
    //(*EventTable(GridDialog)
    //*)Np3mq9PeEx7
END_EVENT_TABLE()

```

```

GridDialog::GridDialog(wxWindow* parent,wxWindowID id,const wxPoint&
pos,const wxSize& size)
{
    //(*Initialize(GridDialog)
    wxSashWindow* SashWindow1;

    Create(parent, wxID_ANY, _("Grid"), wxDefaultPosition, wxDefaultSize,
wxDEFAULT_DIALOG_STYLE, _T("wxID_ANY"));
    SetClientSize(wxSize(288,325));
    SetMinSize(wxSize(-1,-1));
    SetMaxSize(wxSize(-1,-1));
    SashWindow1 = new wxSashWindow(this, ID_SASHWINDOW1,
wxPoint(0,0), wxSize(280,328), wxSW_3D, _T("ID_SASHWINDOW1"));

```

```

    TextCtrl1 = new wxTextCtrl(SashWindow1, ID_TEXTCTRL1,
wxEmptyString, wxPoint(144,121), wxDefaultSize, 0, wxDefaultValidator,
_T("ID_TEXTCTRL1"));

    StaticText3 = new wxStaticText(SashWindow1, ID_STATICTEXT3,
wxEmptyString, wxPoint(176,188), wxDefaultSize, 0,
_T("ID_STATICTEXT3"));

    TextCtrl7 = new wxTextCtrl(SashWindow1, ID_TEXTCTRL7,
wxEmptyString, wxPoint(144,71), wxDefaultSize, 0, wxDefaultValidator,
_T("ID_TEXTCTRL7"));

    StaticText1 = new wxStaticText(SashWindow1, ID_STATICTEXT1,
_T("Конечное значение"), wxPoint(144,51), wxDefaultSize, 0,
_T("ID_STATICTEXT1"));

    wxString __wxRadioBoxChoices_1[3] =
    {
        _T("Высоте"),
        _T("Широте"),
        _T("Долготе")
    };

    RadioBox1 = new wxRadioBox(SashWindow1, ID_RADIOBOX1, _T("Шар"),
wxPoint(36,51), wxSize(79,86), 3, __wxRadioBoxChoices_1, 1, 0,
wxDefaultValidator, _T("ID_RADIOBOX1"));

    StaticText2 = new wxStaticText(SashWindow1, ID_STATICTEXT2,
_T("Значение сетки"), wxPoint(144,101), wxDefaultSize, 0,
_T("ID_STATICTEXT2"));

    Button1 = new wxButton(SashWindow1, ID_BUTTON1, _T("Получения
сетки\nзн-ий"), wxPoint(142,153), wxSize(112,36), 0, wxDefaultValidator,
_T("ID_BUTTON1"));

    Gauge1 = new wxGauge(SashWindow1, ID_GAUGE1, 100, wxPoint(29,299),
wxSize(-1,-1), 0, wxDefaultValidator, _T("ID_GAUGE1"));

```



```

    TextCtrl2 = new wxTextCtrl(SashWindow1, ID_TEXTCTRL2,
wxEmptyString, wxPoint(28,224), wxDefaultSize, 0, wxDefaultValidator,
_T("ID_TEXTCTRL2"));

    StaticText4 = new wxStaticText(SashWindow1, ID_STATICTEXT4, _("Кол-
во итераций"), wxPoint(29,207), wxDefaultSize, 0, _T("ID_STATICTEXT4"));

    Button2 = new wxButton(SashWindow1, ID_BUTTON2, _("Получения зн-
ий"), wxPoint(31,251), wxDefaultSize, 0, wxDefaultValidator,
_T("ID_BUTTON2"));

    SashWindow1->SetSashVisible(wxSASH_TOP, true);
    SashWindow1->SetSashVisible(wxSASH_BOTTOM, true);
    SashWindow1->SetSashVisible(wxSASH_LEFT, true);
    SashWindow1->SetSashVisible(wxSASH_RIGHT, true);
    Center();

Connect(ID_RADIOBOX1,wxEVT_COMMAND_RADIOBOX_SELECTED,(
wxObjectEventFunction)&GridDialog::OnRadioBox1Select);

Connect(ID_BUTTON1,wxEVT_COMMAND_BUTTON_CLICKED,(wxObjec
tEventFunction)&GridDialog::OnButton1Click2);

Connect(ID_TEXTCTRL2,wxEVT_COMMAND_TEXT_UPDATED,(wxObjec
tEventFunction)&GridDialog::OnTextCtrl2Text);

    //*)

}

GridDialog::~GridDialog()
{

```

```

//(*Destroy(GridDialog)
//*)
}

```

```

void GridDialog::OnButton1Click(wxCommandEvent& event)
{
}

```

```

void GridDialog::OnRadioButton1Select(wxCommandEvent& event)
{
}

```

```

void GridDialog::OnRadioBox1Select(wxCommandEvent& event)
{
}

```

```

void GridDialog::OnButton1Click2(wxCommandEvent& event)
{
    ofstream f;
    f.open("test\\testRadioBox1.txt");
    int k;
    // aa.year = 228;
    int year_down = gData.year ;
    int month_down = gData.month;
    int day_down = gData.day ;
    int hour_predsk;
    int min_predsk;
    int sec_predsk;
    double grid;
}

```

```

double endd;
TextCtrl1->GetValue().ToDouble(&grid);
TextCtrl7->GetValue().ToDouble(&endd);
timeCalc
calc(gData.day,gData.month,gData.year,gData.hour,gData.minutes,gData.sec,00)
;
timeCalc
calcGln(gData.day,gData.month,gData.year,gData.hour,gData.minutes,gData.sec
,00);
int zz = 2021;
string text1="MCCJ_211024.agp";//gps
const char* file; // для файла с алм
file = (text1).c_str();
timeCalc calcGalileo(gData.day, gData.month, gData.year, 0,0,0,0);
string text2 = "MCCJ_211108.agl";//gl
const char* file2; // для файла с алм
file2 = (text2).c_str();
const char* file3;
string text3 = "2021-10-22.xml";//gal
file3 = (text3).c_str();
double sko[5];
double Bg,Lg,Hg;
k = RadioBox1->GetSelection(); // получение высоты/широты/долготы
double h = gData.H;
double b = gData.B;
double l = gData.L;
int range=0;
if (k == 0) // высота
{

```

```

range = endd-gData.H;
Gauge1->SetRange(range);
ofstream fgps;
fgps.open("test\\Gps_H.txt");
ofstream fgl;
fgl.open("test\\Glonass_H.txt");
ofstream fgal;
fgal.open("test\\Galileo_H.txt");
for (int i=0; i<=(endd-gData.H); i+=grid)
{
    Gauge1->SetValue(i);
    calccGPS( file, sko, calc,
              (gData.B*M_PI/180.0),(gData.L*M_PI/180.0),h);
    geoc2geod(sko[0], sko[1], sko[2],&Bg,&Lg,&Hg);
    fgps<<h<<"\t"<<
sko[0]<<"\t"<<sko[1]<<"\t"<<sko[2]<<"\t"<<sko[3]<<"\t"<<sko[4]<<Bg<<"\t"
<<Lg<<"\t"<<Hg<<"\n";
    geoc2geod(sko[0], sko[1], sko[2],&Bg,&Lg,&Hg);
    calccGlonass( file2, sko,
calcGln,(gData.B*M_PI/180.0),(gData.L*M_PI/180.0),h);
    fgl<<h<<"\t"<<
sko[0]<<"\t"<<sko[1]<<"\t"<<sko[2]<<"\t"<<sko[3]<<"\t"<<sko[4]<<Bg<<"\t"
<<Lg<<"\t"<<Hg<<"\n";
    calccGalileo(file3,
                  sko,
                  calc,
                  calcGalileo,
                  (gData.B*M_PI/180.0),(gData.L*M_PI/180.0),h);
    geoc2geod(sko[0], sko[1], sko[2],&Bg,&Lg,&Hg);

```

```

    fgal<<h<<"\t"<<
sko[0]<<"\t"<<sko[1]<<"\t"<<sko[2]<<"\t"<<sko[3]<<"\t"<<sko[4]<<Bg<<"\t"
<<Lg<<"\t"<<Hg<<"\n";
    h +=grid;
}
fgps.close();
fgl.close();
fgal.close();
}
if (k == 1) //широта B
{
    range = endd-gData.B;
    Gauge1->SetRange(range);
    ofstream fgps;
    fgps.open("test\\Gps_B.txt");
    ofstream fgl;
    fgl.open("test\\Glonass_B.txt");
    ofstream fgal;
    fgal.open("test\\Galileo_B.txt");

    for (int i=0; i<=(endd-gData.B); i+=grid)
    {
        Gauge1->SetValue(i);
        calccGPS( file, sko, calc,
                (b*M_PI/180.0),(gData.L*M_PI/180.0),gData.H);
        fgps<<b<<"\t"<<
sko[0]<<"\t"<<sko[1]<<"\t"<<sko[2]<<"\t"<<sko[3]<<"\t"<<sko[4]<<"\n";
        calccGlonass( file2, sko,
        calcGln,(b*M_PI/180.0),(gData.L*M_PI/180.0),gData.H);

```

```

    fgl<<b<<"\t"<<
sko[0]<<"\t"<<sko[1]<<"\t"<<sko[2]<<"\t"<<sko[3]<<"\t"<<sko[4]<<"\n";
    calccGalileo(file3,
        sko,
        calc,
        calcGalileo,
        (b*M_PI/180.0),(gData.L*M_PI/180.0),gData.H);
    fgal<<b<<"\t"<<
sko[0]<<"\t"<<sko[1]<<"\t"<<sko[2]<<"\t"<<sko[3]<<"\t"<<sko[4]<<"\n";
    b +=grid;
}
fgps.close();
fgl.close();
fgal.close();
}
if (k == 2)
{
    range = endd-gData.L;
    Gauge1->SetRange(range);
    ofstream fgps;
    fgps.open("test\\Gps_L.txt");
    ofstream fgl;
    fgl.open("test\\Glonass_L.txt");
    ofstream fgal;
    fgal.open("test\\Galileo_L.txt");
    for (int i=0; i<=(endd-gData.L); i+=grid)
    {
        Gauge1->SetValue(i);
        calccGPS( file, sko, calc,

```

```

(gData.B*M_PI/180.0),(l*M_PI/180.0),gData.H);
//geoc2geod(sko[0], sko[1], sko[2],Bg,Lg,Hg);
fgps<<l<<"\t"<<
sko[0]<<"\t"<<sko[1]<<"\t"<<sko[2]<<"\t"<<sko[3]<<"\t"<<sko[4]<<Bg<<"\t"
<<Lg<<"\t"<<Hg<<"\n";
calccGlonass( file2, sko,
calcGln,(gData.B*M_PI/180.0),(l*M_PI/180.0),gData.H);
fgl<<l<<"\t"<<
sko[0]<<"\t"<<sko[1]<<"\t"<<sko[2]<<"\t"<<sko[3]<<"\t"<<sko[4]<<"\n";
calccGalileo(file3,
sko,
calc,
calcGalileo,
(gData.B*M_PI/180.0),(l*M_PI/180.0),gData.H);
fgal<<l<<"\t"<<
sko[0]<<"\t"<<sko[1]<<"\t"<<sko[2]<<"\t"<<sko[3]<<"\t"<<sko[4]<<"\n";
l += grid;
}
fgps.close();
fgl.close();
fgal.close();
}
f<<"choise ="<<k;
f<<"grid = "<<grid;
f<<"day_down ="<<day_down<<"\n";
f<<"month_down ="<<month_down<<"\n";
f<<"year_down ="<<year_down<<"\n";
f<<"gData.hour ="<<gData.hour<<"\n";
f<<"minutes ="<<gData.minutes<<"\n";

```

```

f<<"sec ="<<gData.sec<<"\n";
f<<"B ="<<gData.B<<"\n";
f<<"L ="<<gData.L<<"\n";
f<<"H ="<<gData.H<<"\n";
f<<"t ="<<gData.typeGnss<<"\n";
f.close();
}

```

```

void GridDialog::OnTextCtrl2Text(wxCommandEvent& event)
{
}

```

```

void GridDialog::OnButton2Click1(wxCommandEvent& event)
{

```

```

    int iter;
    double iteration;
    int year_down=2021;
    int month_down;
    int day_down;
    int hour_predsk;
    int min_predsk;
    int sec_predsk;
    TextCtrl2 ->GetValue().ToDouble(&iteration);
    iter = static_cast<int>(iteration);
    string text1="MCCJ_211024.agp";//gps
    const char* file; // для файла с алм
    file = (text1).c_str();
    timeCalc calcGalileo(gData.day, gData.month, gData.year, 0,0,0,0);

```



```

string text2 = "MCCJ_211108.agl";//gl
const char* file2; // для файла с алм
file2 = (text2).c_str();
const char* file3;
string text3 = "2021-10-22.xml";//gal
file3 = (text3).c_str();
double Bg,Lg,Hg;
Gauge1->SetRange(iter);
ofstream fgps;
fgps.open("test\\Gps_rand.log");
ofstream fgl;
fgl.open("test\\Glonass_rand.log");
ofstream fgal;
fgal.open("test\\Galileo_rand.log");
ofstream ftimes;
ftimes.open("test\\times_rand.log");
ofstream fcoord;
fcoord.open("test\\coord_rand.log");
double H,B,L;
double sko[5];
for (int i = 0; i<=iter; i++)
{
    Gauge1->SetValue(i);
    month_down = rand() % 12+1;
    day_down = rand()%30+1;
    hour_predsk = rand()%24;
    min_predsk = rand()%60;
    sec_predsk = rand()%60;
    B = rand()%180;

```

```

if (B>90)
    B =90-B;
L = rand()%180;
if (L>90)
    L=90-L;
H = rand()%1000;

fcoord<<H<<"\t"<<B<<"\t"<<L<<"\n";
timeCalc
calc(day_down,month_down,year_down,hour_predsk,min_predsk,sec_predsk,00
);
timeCalc
calcGln(day_down,month_down,year_down,hour_predsk,min_predsk,sec_preds
k,00);
timeCalc calcGalileo(day_down,month_down,year_down, 0,0,0,0);

ftimes<<day_down<<"\t"<<month_down<<"\t"<<year_down<<"\t"<<hour_pred
sk<<"\t"<<min_predsk<<"\t"<<sec_predsk<<"\n";

calccGPS( file, sko, calc,
    (B*M_PI/180.0),(L*M_PI/180.0),H);
geoc2geod(sko[0], sko[1], sko[2],&Bg,&Lg,&Hg);
fgps<<H<<"\t"<<L<<"\t"<<B<<"\t"<<
sko[0]<<"\t"<<sko[1]<<"\t"<<sko[2]<<"\t"<<sko[3]<<"\t"<<sko[4]<<"\t"<<Bg
<<"\t"<<Lg<<"\t"<<Hg<<"\n";
calccGlonass( file2, sko, calcGln,(B*M_PI/180.0),(L*M_PI/180.0),H);
geoc2geod(sko[0], sko[1], sko[2],&Bg,&Lg,&Hg);

```

```

        fgl<<H<<"\t"<<L<<"\t"<<B<<"\t"<<
sko[0]<<"\t"<<sko[1]<<"\t"<<sko[2]<<"\t"<<sko[3]<<"\t"<<sko[4]<<"\t"<<Bg
<<"\t"<<Lg<<"\t"<<Hg<<"\n";
        calccGalileo(file3,
                        sko,
                        calc,
                        calcGalileo,
                        (B*M_PI/180.0),(L*M_PI/180.0),H);
        geoc2geod(sko[0], sko[1], sko[2],&Bg,&Lg,&Hg);
        fgal<<H<<"\t"<<L<<"\t"<<B<<"\t"<<
sko[0]<<"\t"<<sko[1]<<"\t"<<sko[2]<<"\t"<<sko[3]<<"\t"<<sko[4]<<"\t"<<Bg
<<"\t"<<Lg<<"\t"<<Hg<<"\n";
    }
    fgps.close();
    fgl.close();
    fgal.close();
    ftimes.close();
    fcoord.close();
}
GridDialog.h"
#ifndef GRIDDIALOG_H
#define GRIDDIALOG_H

//(*Headers(GridDialog)
#include <wx/button.h>
#include <wx/dialog.h>
#include <wx/gauge.h>
#include <wx/radiobox.h>
#include <wx/sashwin.h>

```

```

#include <wx/stattext.h>
#include <wx/textctrl.h>
/*)
extern struct GlobalData{
    int year;
    int month;
    int day;
    int hour;
    int minutes;
    int sec;
    int typeGnss; //0 - Glonass;
    double B;/**PI/180
    double L;
    double H;
} gData;
//extern mydata aa;
class GridDialog: public wxDialog
{
    public:

        GridDialog(wxWindow* parent,wxWindowID
id=wxID_ANY,const wxPoint& pos=wxDefaultPosition,const wxSize&
size=wxDefaultSize);

        virtual ~GridDialog();

        /*Declarations(GridDialog)
        wxButton* Button1;
        wxButton* Button2;
        wxGauge* Gauge1;

```

```

wxRadioBox* RadioBox1;
wxStaticText* StaticText1;
wxStaticText* StaticText2;
wxStaticText* StaticText3;
wxStaticText* StaticText4;
wxTextCtrl* TextCtrl1;
wxTextCtrl* TextCtrl2;
wxTextCtrl* TextCtrl7;
//*)

```

protected:

```

//(*Identifiers(GridDialog)
static const long ID_TEXTCTRL1;
static const long ID_STATICTEXT3;
static const long ID_TEXTCTRL7;
static const long ID_STATICTEXT1;
static const long ID_RADIOBOX1;
static const long ID_STATICTEXT2;
static const long ID_BUTTON1;
static const long ID_GAUGE1;
static const long ID_TEXTCTRL2;
static const long ID_STATICTEXT4;
static const long ID_BUTTON2;
static const long ID_SASHWINDOW1;
//*)

```

private:

```

//(*Handlers(GridDialog)
void OnButton1Click(wxCommandEvent& event);
void OnRadioButton1Select(wxCommandEvent& event);
void OnRadioBox1Select(wxCommandEvent& event);
void OnCheckListBox1Toggled(wxCommandEvent& event);
void OnCheckListBox1Toggled1(wxCommandEvent& event);
void OnSashWindow1SashDragged(wxSashEvent& event);
void OnButton1Click2(wxCommandEvent& event);
void OnTextCtrl2Text(wxCommandEvent& event);
void OnButton2Click(wxCommandEvent& event);
void OnButton2Click1(wxCommandEvent& event);
//*)

DECLARE_EVENT_TABLE()

};

#endif

```

Приложение 2

Расчет матрицы \mathbf{D}_x :

$$\mathbf{D}_x = \left((\tilde{\mathbf{H}}^T \mathbf{D}_{n_R}^{-1} \tilde{\mathbf{H}})^{-1} \right)^T =$$

$$= \left(\left(\begin{array}{c|c} \begin{matrix} H_{1,1} & H_{2,1} & \dots & H_{n,1} \\ H_{1,2} & H_{2,2} & \dots & H_{n,2} \\ H_{1,3} & H_{2,3} & \dots & H_{n,3} \\ H_{1,4} & H_{2,4} & \dots & H_{n,4} \end{matrix} & \begin{matrix} \frac{1}{D_1} & 0 & 0 & 0 \\ 0 & \frac{1}{D_2} & 0 & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \frac{1}{D_n} \end{matrix} \end{array} \right) \cdot \left(\begin{array}{c|c} \begin{matrix} H_{1,1} & H_{1,2} & H_{1,3} & H_{1,4} \\ H_{2,1} & H_{2,2} & H_{2,3} & H_{2,4} \\ \dots & \dots & \dots & \dots \\ H_{n,1} & H_{n,2} & H_{n,3} & H_{n,4} \end{matrix} & \end{array} \right)^{-1} \right)^T =$$

$$\begin{aligned}
&= \left(\left(\begin{array}{c|c} \begin{matrix} H_{1,1} & H_{2,1} & \cdots & H_{n,1} \\ H_{1,2} & H_{2,2} & \cdots & H_{n,2} \\ H_{1,3} & H_{2,3} & \cdots & H_{n,3} \\ H_{1,4} & H_{2,4} & \cdots & H_{n,4} \end{matrix} & \begin{matrix} \frac{H_{1,1}}{D_1} & \frac{H_{1,2}}{D_1} & \frac{H_{1,3}}{D_1} & \frac{H_{1,4}}{D_1} \\ \frac{H_{2,1}}{D_2} & \frac{H_{2,2}}{D_2} & \frac{H_{2,3}}{D_2} & \frac{H_{2,4}}{D_2} \\ \dots & \dots & \dots & \dots \\ \frac{H_{n,1}}{D_n} & \frac{H_{n,2}}{D_n} & \frac{H_{n,3}}{D_n} & \frac{H_{n,4}}{D_n} \end{matrix} \end{array} \right)^{-1} \right)^T \\
&= \left(\left(\begin{array}{cc} \frac{H_{1,1}^2}{D_1} + \frac{H_{2,1}^2}{D_2} + \dots + \frac{H_{n,1}^2}{D_n} & \frac{H_{1,1} \cdot H_{1,2}}{D_1} + \frac{H_{2,1} \cdot H_{2,2}}{D_2} + \dots + \frac{H_{n,1} \cdot H_{n,2}}{D_n} \\ \frac{H_{1,2} \cdot H_{1,1}}{D_1} + \frac{H_{2,2} \cdot H_{2,1}}{D_2} + \dots + \frac{H_{n,2} \cdot H_{n,1}}{D_n} & \frac{H_{1,2}^2}{D_1} + \frac{H_{2,2}^2}{D_2} + \dots + \frac{H_{n,2}^2}{D_n} \\ \frac{H_{1,3} \cdot H_{1,1}}{D_1} + \frac{H_{2,3} \cdot H_{2,1}}{D_2} + \dots + \frac{H_{n,3} \cdot H_{n,1}}{D_n} & \frac{H_{1,3} \cdot H_{1,2}}{D_1} + \frac{H_{2,3} \cdot H_{2,2}}{D_2} + \dots + \frac{H_{n,3} \cdot H_{n,2}}{D_n} \\ \frac{H_{1,4} \cdot H_{1,1}}{D_1} + \frac{H_{2,4} \cdot H_{2,1}}{D_2} + \dots + \frac{H_{n,4} \cdot H_{n,1}}{D_n} & \frac{H_{1,4} \cdot H_{1,2}}{D_1} + \frac{H_{2,4} \cdot H_{2,2}}{D_2} + \dots + \frac{H_{n,4} \cdot H_{n,2}}{D_n} \end{array} \right)^{-1} \right)^T \\
&\quad \left(\begin{array}{cc} \frac{H_{1,1} \cdot H_{1,3}}{D_1} + \frac{H_{2,1} \cdot H_{2,3}}{D_2} + \dots + \frac{H_{n,1} \cdot H_{n,3}}{D_n} & \frac{H_{1,1} \cdot H_{1,4}}{D_1} + \frac{H_{2,1} \cdot H_{2,4}}{D_2} + \dots + \frac{H_{n,1} \cdot H_{n,4}}{D_n} \\ \frac{H_{1,2} \cdot H_{1,3}}{D_1} + \frac{H_{2,2} \cdot H_{2,3}}{D_2} + \dots + \frac{H_{n,2} \cdot H_{n,3}}{D_n} & \frac{H_{1,2} \cdot H_{1,4}}{D_1} + \frac{H_{2,2} \cdot H_{2,4}}{D_2} + \dots + \frac{H_{n,2} \cdot H_{n,4}}{D_n} \\ \frac{H_{1,3}^2}{D_1} + \frac{H_{2,3}^2}{D_2} + \dots + \frac{H_{n,3}^2}{D_n} & \frac{H_{1,3} \cdot H_{1,4}}{D_1} + \frac{H_{2,3} \cdot H_{2,4}}{D_2} + \dots + \frac{H_{n,3} \cdot H_{n,4}}{D_n} \\ \frac{H_{1,4} \cdot H_{1,3}}{D_1} + \frac{H_{2,4} \cdot H_{2,3}}{D_2} + \dots + \frac{H_{n,4} \cdot H_{n,3}}{D_n} & \frac{H_{1,4}^2}{D_1} + \frac{H_{2,4}^2}{D_2} + \dots + \frac{H_{n,4}^2}{D_n} \end{array} \right)^{-1} \right)^T
\end{aligned}$$