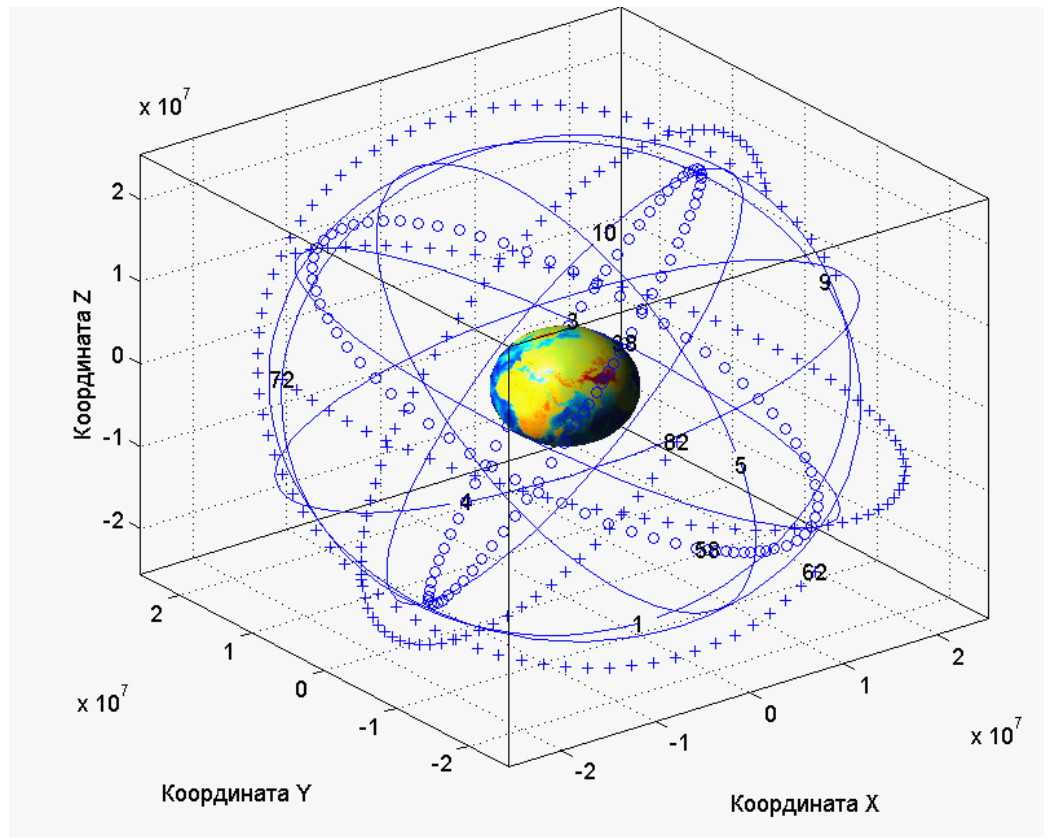


В. В. Конин, Л. А. Кони́на



Спутниковые системы навигации. Учебное пособие

(Лабораторный практикум на компьютере)

Київ -2008

Рецензенты:

Л. В. Сибрук, заведующий кафедрой электродинамики Института электроники и систем управления Национального авиационного университета, доктор техн. наук, профессор

Национальный технический университет «Киевский политехнический институт», кафедра Информационных и телекоммуникационных сетей ИТС НТУУ «КПИ», зав. кафедрой доктор техн. наук, профессор Л. С. Глоба

Харьковский национальный университет радиоэлектроники, кафедра Основ радиотехники, зав. кафедрой доктор техн. наук, профессор В. М. Шокколо

Конин В. В., Кони́на Л. А.

Спутниковые системы навигации. Лабораторный практикум на компьютере:

Приводятся лабораторный практикум по вопросам спутниковых радионавигационных систем в виде открытых программных комплексов в среде MatLab. Лабораторный практикум в виде 16 работ с заданиями, примерами и контрольными вопросами охватывает направления формирования сигналов спутниковых навигационных систем GPS и ГЛОНАСС, преобразования координат, моделирование орбит навигационных спутников GPS, ГЛОНАСС, GALILEO, декодирования и расшифровка данных навигационных спутников, решение навигационной задачи расчета позиции приемника пользователя.

Для студентов, аспирантов и преподавателей дисциплин, изучающих системы и компоненты спутниковых средств навигации.

ВВЕДЕНИЕ.....	6
РАЗДЕЛ 1 СИГНАЛЫ GNSS.....	7
1.1 Краткие сведения из теории.....	7
1.2 Относительная фазовая манипуляция.....	9
1.2.1 Лабораторная работа 1. 1 «Относительная фазовая манипуляция».....	9
1.3 Псевдослучайный код спутников GPS.....	12
1.3.1 Лабораторная работа 1. 2 «Код спутников GPS».....	12
1.4 Функции и файлы из папки cod_GPS.....	13
1.5 Псевдослучайный код спутников ГЛОНАСС.....	32
1.5.1 Лабораторная работа 1. 3 «Код спутников ГЛОНАСС».....	32
1.6 Функции и файлы из папки CodGLONASS.....	33
1.7 Вопросы и задания для самостоятельной подготовки.....	37
РАЗДЕЛ 2 Преобразование координат.....	38
2.1 Краткие сведения из теории.....	38
2.2 Лабораторная работа 2. 1 «Преобразование координат».....	38
2.3 Вопросы и задания для самоподготовки.....	40
2.4 Тексты программ.....	41
2.4.1 Функции и файлы из папки COORDINATES	41
2.4.2 Функции и файлы из папки ECI_ECEF_LLH	45
2.4.3 Функции и файлы из папки TEST	48
РАЗДЕЛ 3 Время.....	53
3.1 Краткие сведения из теории.....	53
3.2 Лабораторная работа 3. 1 «Время в спутниковых радионавигационных системах».....	53
3.3 Вопросы и задания для самоподготовки.....	55
3.4 Тексты программ.....	55
3.4.1 Функции и файлы из папки TIME	55
3.4.2 Функции и файлы из папки TIME_S0	60
РАЗДЕЛ 4 Орбитальное движение навигационных спутников.....	70
4.1 Орбитальное движение спутников GPS.....	70
4.1.1 Краткие сведения из теории.....	70
4.1.2 Лабораторная работа 4. 1 «Орбитальное движение спутников GPS».....	70
4.1.3 Вопросы и задания для самоподготовки.....	72

4.1.4	Функции и файлы из папки ORBITA_GPSv1	72
4.1.5	Примеры расчетов с помощью m-файла :Orbita_GPS.m	82
4.1.6	Лабораторная работа 4. 2 «Наблюдение спутников GPS»	85
4.1.7	Вопросы и задания для самоподготовки.....	85
4.1.8	Файл из папки ORBITA_GPSv2	86
4.1.9	Пример выполнения файла Orbita_GPS_1.m.....	93
4.2	Размножения эфемерид спутников ГЛОНАСС (иллюстрация решения системы дифференциальных уравнений).....	96
4.2.1	Краткие сведения из теории.....	96
4.2.2	Лабораторная работа 4. 3 «Решения системы дифференциальных уравнений»	96
4.2.3	Вопросы и задания для самоподготовки.....	97
4.2.4	Файл ORBITA_1.m.....	98
4.3	Орбитальное движение спутников ГЛОНАСС	99
4.3.1	Краткие сведения из теории.....	99
4.3.2	Лабораторная работа 4. 4 «Орбитальное движение спутников ГЛОНАСС»....	100
4.3.3	Вопросы и задания для самоподготовки.....	101
4.3.4	Функции и файлы из папки ORBITA_GL_NAVIOR	101
4.3.5	Примеры выполнения комплекса программ ORBITA_GL_NAVIOR	138
РАЗДЕЛ 5 Преобразование данных навигационных спутников.....		141
5.1	Преобразование данных альманаха приемника СН 4701 в формат YUMA.....	141
5.1.1	Краткие сведения из теории.....	141
5.1.2	Лабораторная работа 5. 1 «Конвертирование данных альманаха GPS и ГЛОНАСС в формат » YUMA	142
5.1.3	Задание для самоподготовки.....	142
5.1.4	Функции и файлы из папки ALM_CH4701_V3	142
5.2	Декодирование данных альманаха спутников GPS	165
5.2.1	Краткие сведения из теории.....	165
5.2.2	Лабораторная работа 5. 2 «Декодирование данных альманаха навигационных приемников на базе плат OEM- 4»	165
5.2.3	Задание для самоподготовки.....	166
5.2.4	Функции и файлы из папки RAW_ALM_PRG	166
5.3	Модель движения навигационных спутников GPS и ГЛОНАСС.....	173
5.3.1	Краткие сведения из теории.....	173
5.3.2	Лабораторная работа 5. 3 «Модель движения и визуализация спутников GPS и ГЛОНАСС».....	173

5.3.3 Контрольные вопросы и задания для самоподготовки	174
5.3.4 Функции и файлы из папки Vsion_GLONASS_GPS	174
5.4 Модель движения спутников GPS, ГЛОНАСС, GALILEO	185
5.4.1 Краткие сведения из теории.....	185
5.4.2 Лабораторная работа 5. 4 «Модель движения и визуализация спутников GPS, ГЛОНАСС, GALILEO »	186
5.4.3 Задание для самоподготовки.....	186
5.4.4 Файл из папки Vision_GLONASS_GPS_GALILEO	187
5.4.5 Лабораторная работа 5. 5 «Орбиты спутников GPS, ГЛОНАСС, GALILEO ».	195
5.4.6 Задания и вопросы для самоподготовки	196
5.4.7 Листинг файла ORBITA_GGG.m	196
5.4.8 Пример выполнения файла:ORBITA_GGG.m.....	202
5.5 Декодирование и расшифровка данных спутников ГЛОНАСС.....	202
5.5.1 Краткие сведения из теории.....	202
5.5.2 Лабораторная работа 5. 6 «Декодирование данных спутников ГЛОНАСС в навигационном приемнике».....	203
5.5.3 Задания и вопросы для самоподготовки	209
5.5.4 Листинг файла:Decod_GL3.m	210
5.5.5 Пример выполнения файла:Decod_GL3.m	222
РАЗДЕЛ 6 Решение навигационной задачи	224
6.1 Краткие сведения из теории.....	224
6.2 Лабораторная работа 4. 1 «Решение навигационной задачи»	224
6.3 Вопросы и задания для самоподготовки	224
6.4 Файл из папки «Координаты приемника»	225
ПРИЛОЖЕНИЯ.....	230
Приложение 1 Отчеты по лабораторной работе	230
Приложение 2 Альманах almanac_yuma_week0371_589824.txt	231
Приложение 3 Альманах 002.txt	241
Приложение 4 Альманах AlmGGG.yum	258
ПЕРЕЧЕНЬ ССЫЛОК	286

ВВЕДЕНИЕ

Рубеж XX- XXI веков выделяется двумя значимыми событиями. Глобальные спутниковые радионавигационные системы позиционирования становятся в ряд систем массового обслуживания, компьютер стал в неограниченных объемах доступен студентам. Действительно, спутниковая радионавигация применяется в авиации, управлении наземным и морским транспортом, геодезии, картографии, мониторинге газо и нефтепроводов, высотных сооружений, наблюдениям за смещением материков и многих других отраслях. При этом достигнуты точности определения координат от миллиметров до нескольких десятков метров. Мировое сообщество может пользоваться спутниковыми системами GPS и ГЛОНАСС безвозмездно. В стадии завершения находятся системы EGNOS и GALILEO, ориентированные только на гражданских потребителей. Следует ожидать, что в 2008-2009 годах спутниковая радионавигация будет присутствовать в каждом мобильном телефоне. Среди компьютерных технологий выделяется система MatLab, имеющая открытые программные коды и позволяющая решить практически любую вычислительную задачу. Руководство пользователя системой MatLab содержит более 4000 страниц, значительная часть которых переведена на русский язык. Достаточно обратиться к библиографии книг [7, 8].

В предлагаемом учебном пособии впервые предпринята попытка представить сложный вычислительный материал по вопросам спутниковой радионавигации в виде лабораторных работ, выполняемых по завершенным программным продуктам, написанным в среде MatLab. Всего таких работ 16, каждая работа рассчитана на 4 академических часа. Работы подготовлены на основании курсов, изучаемых в Национальном авиационном университете: «Спутниковые системы связи, навигации, наблюдений», «Моделирование спутниковых радионавигационных систем», «Авиационные геоинформационные системы и технологии», «Глобальные спутниковые системы позиционирования». Данные курсы читаются бакалаврам, специалистам и магистрантам. Справедливо, что приведенные в пособии работы в объеме, определяемом заданиями к каждой работе, дают базовый уровень знаний. Фактически же в каждую из работ заложен потенциал, позволяющий при творческом дополнении программ и выполнении исследований по программам трансформировать их в курсовые работы, а при интегрировании программных кодов выполнять эффективные дипломные работы. Все программные продукты, приведенные в пособии, проверены экспериментально.

Авторы с благодарностью примут замечания и предложения по совершенствованию пособия по адресам: cnstatm@nau.edu.ua, vkonin@mail.ru.

РАЗДЕЛ 1 СИГНАЛЫ GNSS

1.1 Краткие сведения из теории

Спутниковые радионавигационные системы формируют в околоземном пространстве радионавигационное поле. Средства, обеспечивающие навигацию: спутниковые навигационные приемники, станции с дифференциальным режимом (контрольно-корректирующие станции), аппаратура обслуживания принимают и обрабатывают информацию из радионавигационного поля и решают задачи в соответствии с функциональным назначением [1, 2, 3, 4, 8, 9].

На навигационных GPS, ГЛОНАСС и геостационарных спутниках формируются и излучаются следующие сигналы

$$s_i(t) = \sqrt{2 \cdot P_{i,I}} \cdot D_i(t) \cdot C_i(t) \cdot \cos(\omega_{L1} \cdot t + \theta) + \sqrt{2 \cdot P_{i,Q}} \cdot D_i(t) \cdot P_i(t) \cdot \sin(\omega_{L1} \cdot t + \theta); \quad (1.1)$$

$$s_i(t) = \sqrt{2 \cdot P_{i,Q}} \cdot D_i(t) \cdot P_i(t) \cdot \sin(\omega_{L2} \cdot t + \theta); \quad (1.2)$$

$$s_i(t) = \sqrt{2 \cdot P_{i,I}} \cdot D_i(t) \cdot C_i(t) \cdot \cos(\omega_{L5} \cdot t + \theta) + \sqrt{2 \cdot P_{i,Q}} \cdot C_i(t) \cdot \sin(\omega_{L5} \cdot t + \theta); \quad (1.3)$$

$$s_i(t) = \sqrt{2 \cdot P_i} \cdot D_{i,ГЛ}(t) \cdot C(t) \cdot \cos(\omega_{i,L1} \cdot t + \theta_i); \quad (1.4)$$

$$s_i(t) = \sqrt{2 \cdot P_i} \cdot D_{i,ГЛ}(t) \cdot C(t) \cdot \cos(\omega_{i,L2} \cdot t + \theta_i); \quad (1.5)$$

$$s_i(t) = \sqrt{2 \cdot P_{i,Geo}} \cdot D_{i,Geo}(t) \cdot C_{i,Geo}(t) \cdot \cos(\omega_{L1} \cdot t + \theta), \quad (1.6)$$

где $s_i(t)$ -сигнал i -го спутника; t -системное время соответствующего спутника; $P_{i,I}$ - мощность синфазной составляющей i -го спутника GPS; $D_i(t)$ -данные i -го спутника GPS; $C_i(t)$ - C/A- код i -го спутника GPS; ω_{L1} - круговая частота спутника GPS , соответствующая частоте $L1$ с учетом доплеровского сдвига; θ - начальный фазовый сдвиг; $P_{i,Q}$ - мощность квадратурной составляющей i -го спутника GPS; $P_i(t)$ - P - код i -го спутника GPS; ω_{L2} - круговая частота спутника GPS , соответствующая частоте $L2$ с учетом доплеровского сдвига; ω_{L5} - круговая частота спутника GPS , соответствующая частоте $L5$ с учетом доплеровского сдвига; P_i -мощность сигнала i -го спутника ГЛОНАСС; $D_{i,ГЛ}(t)$ -данные i -го спутника ГЛОНАСС; $C(t)$ - код спутника ГЛОНАСС; $\omega_{i,L1}$ - круговая частота i -го спутника ГЛОНАСС;

НАСС, соответствующая частоте $L1$ с учетом доплеровского сдвига; $\omega_{i,L2}$ - круговая частота i -го спутника ГЛОНАСС, соответствующая частоте $L2$ с учетом доплеровского сдвига; $P_{i,Geo}$ -мощность сигнала геостационарного спутника; $D_{i,Geo}(t)$ -данные геостационарного спутника; $C_{i,Geo}(t)$ - код геостационарного спутника.

В выражениях (1.1-1. 2) составляющие сигнала $D(t)$ (индексы опущены) в идеализированном виде представляют данные, передаваемые навигационными спутниками в виде символов с амплитудой ± 1 , длительностью 20 миллисекунд и частотой следования 50 Гц. Составляющие $C_i(t)$ есть псевдослучайные последовательности символов с амплитудой ± 1 , длительностью 0.97752 микросекунд, частотой следования 1.023 МГц, периодом повторения 1 миллисекунда. Составляющие $P_i(t)$ есть псевдослучайные последовательности символов с амплитудой ± 1 , длительностью 0.097752 микросекунд, частотой следования 10.23 МГц, периодом повторения 7 суток. В формулах (1. 3, 1. 4) составляющая сигнала $D(t)$ (индексы опущены) в идеализированном виде представляют данные, передаваемые спутниками в виде символов с амплитудой ± 1 , длительностью 20 миллисекунд и частотой следования 50 Гц. Составляющие $C(t)$ есть псевдослучайные последовательности символов с амплитудой ± 1 , длительностью 1.9569 микросекунд, частотой следования 0. 511 МГц, периодом повторения 1 миллисекунда. В формуле (1. 5) составляющая сигнала $D(t)$ (индексы опущены) в идеализированном виде представляют данные, передаваемые геостационарными спутниками, в виде символов с амплитудой ± 1 и частотой следования 250 бит/с. Составляющие $C(t)$ есть псевдослучайные последовательности символов с амплитудой ± 1 , длительностью 0.97752 микросекунд, частотой следования 1. 023 МГц, периодом повторения 1 миллисекунда.

Сигналы (1.1-1. 6) получили название - сигналов с расширенным спектром (spread-spectrum signal). Такие сигналы характеризуются следующими признаками: полоса частот, в которой передаются данные $D(t)$ значительно шире минимально необходимой; расширение спектра сигнала производится шумоподобными сигналами $C(t)$ или $P(t)$; восстановление данных в приемнике производится путем сопоставления принятого сигнала с его копией после синхронизации. При восстановлении сигнала происходит сужение спектра и выделение полезного сигнала из шумов.

Сигналы навигационных спутников, как на спутниках, так и в навигационных приемниках потребителя подвергаются специальной обработке для эффективной передачи, поиска, обнаружения, слежения, измерения в условиях помех данных без потери информации. Основные компоненты сигналов спутников и изучаются в приведенных в настоящем разделе лабораторных работах.

Рассматривается *метод относительной фазовой манипуляции*, применяемый в системе ГЛОНАСС. Метод осуществляется посредством перекодировки исходной последовательности информационных символов по следующему алгоритму:

$$a_{\text{вых } i} = a_{\text{вх } i} \oplus a_{\text{вых } i-1}, \quad (1.7)$$

где $a_{\text{вх } i}$, $a_{\text{вых } i}$ – входная и выходная последовательности символов при передаче соответственно.

При приеме перекодировка выполняется по правилу

$$b_i = a_{\text{вых } i-1} \oplus a_{\text{вых } i}, \quad (1.8)$$

где b_i – последовательность символов после перекодировки на выходе приемника.

Изучаются методы формирования *псевдослучайных кодов спутников GPS и ГЛОНАСС*, алгоритмы формирования которых основаны на получении псевдослучайных сигналов посредством последовательных регистров сдвига.

Исследуются приемы и методы *корреляции* псевдослучайных сигналов для нахождения сигнала спутника.

Детальные теоретические сведения о спутниковых навигационных сигналах можно найти в работах [1 -4, 9].

1.2 Относительная фазовая манипуляция

1.2.1 Лабораторная работа 1.1 «Относительная фазовая манипуляция»

Цель лабораторной работы: реализация относительной фазовой манипуляции в средах систем Simulink и MatLab.

Рекомендуется следующий порядок выполнения лабораторной работы.

1. Откройте MatLab и запустите Simulink [5, 6].
2. Откройте в Simulink окно для создания нового документа и выполните **задание 1, 2, 3**.
3. **Задание 1.** Из блоков библиотеки Simulink соберите схему, изображенную на рис. 1.1. В этой схеме применяются следующие блоки. Блок g_F содержит данные (символы), которые требуется перекодировать. Блоки $delay$ выполняют задержку на один символ, блоки mo_d сумматоры по модулю 2. Блок *Oscilloscope* трехканальный осциллограф, регистрирующий результаты работы схемы.
4. **Задание 2.** Установите параметры блоков g_F , $delay$, mo_d в соответствии с рис. 1.2.
5. **Задание 3.** Выполните моделирование в среде Simulink. Для приведенных входных данных результат моделирования изображен на рис. 1.3.

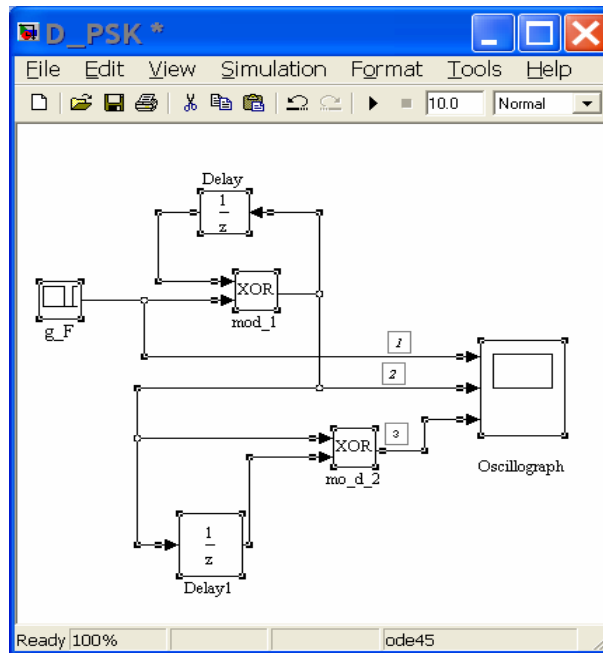
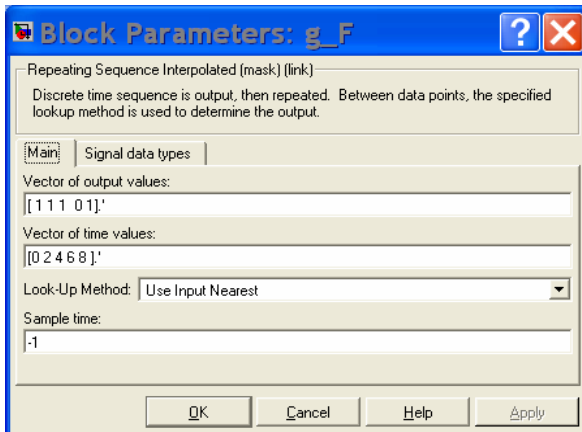
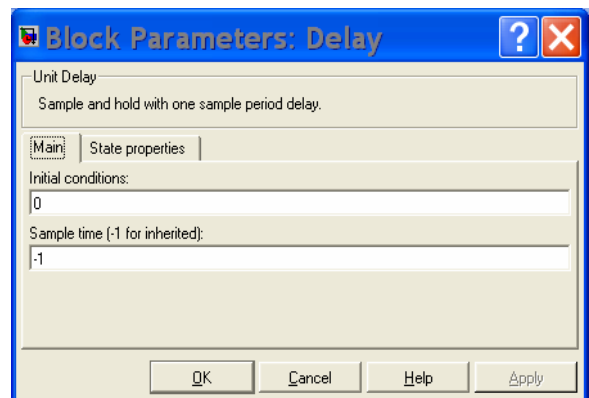


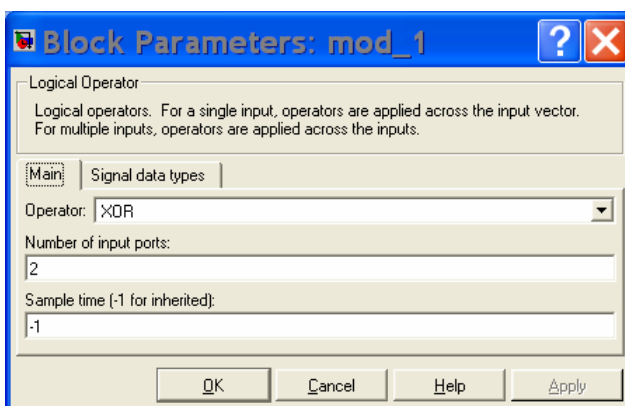
Рис. 1.1. Схема перекодировки символьной информации



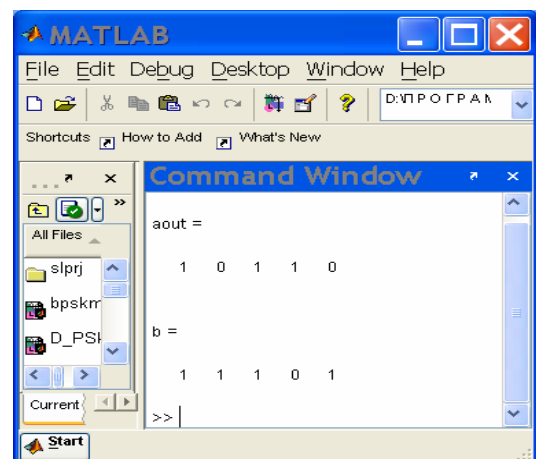
а



б



в



г

Рис. 1.2. Окна для установки параметров: а - блока *g_F*; б-блоков *delat*; в - блоков *mod*; г-командное окно *MatLab*.

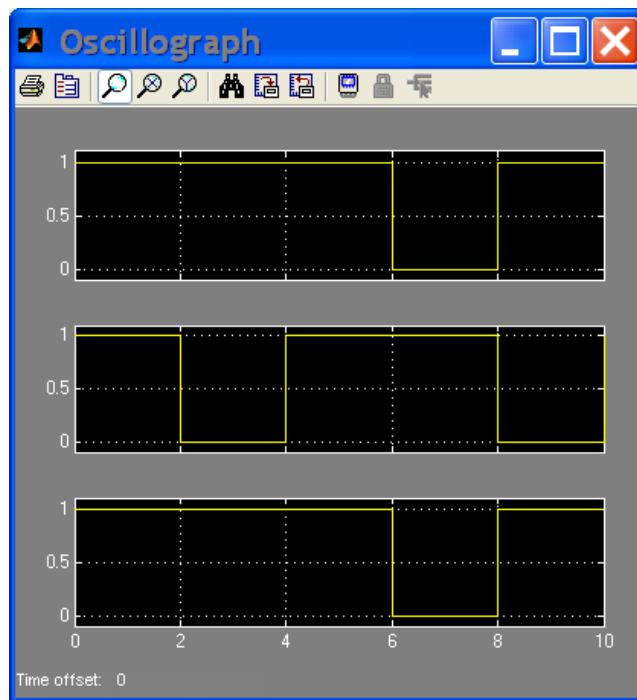


Рис. 1.3. Результат моделирования схемы рис. 1.1

Символы $[11101]$ в сигнальной форме представлены на верхнем графике рис.3 и соответствуют данным, передаваемым по каналу 1 (рис. 1.1). Средний график рис. 1.3 есть результат выполнения алгоритма (1.7) и соответствует данным, сформированным в канале 2 (рис. 1.1). Нижний график (рис. 1.3) есть результат применения алгоритма (1.8)- восстановление данных и соответствует данным, сформированным в канале 3 (рис. 1.1).

6. Выполните моделирование преобразований по алгоритмам (1.7, 1.8) в MatLab.
7. Откройте MatLab и выполните задания 4, 5, 6.
8. **Задание 4.** Создайте следующий m- файл.

Процедура перекодировки по алгоритмам (1.7) и (1.8) в виде m- файла

```
% m - файл процедура перекодировки
%Алгоритм (1.7)
% входная последовательность
a=[1 1 1 0 1];
% выходная последовательность
aout(1)=a(1);
i = 1;
for i = 2 : 5
    aout(i)=xor(a(i),aout(i-1));
end
aout %вывод выходной последовательности в командное окно
% Алгоритм (1.8)
b(1) = aout(1);
for i = 2 : 5
```

```
b(i) = xor(aout(i-1), aout(i));  
end  
b %вывод обратного преобразования в командное окно
```

9. **Задание 5.** Выполните m- файл и убедитесь, что результат выполнения соответствует данным, изображенным в командном окне (рис. 1.2, г).
10. **Задание 6.** Задайте данные для преобразования в виде последовательности из 10 символов. Для заданной последовательности выполните задания 1- 4. Результаты выполнения занесите в отчет

1.3 Псевдослучайный код спутников GPS

1.3.1 Лабораторная работа 1. 2 «Код спутников GPS»

Цель лабораторной работы - формирование и исследование псевдослучайного кода спутников GPS.

Рекомендуется следующий порядок выполнения лабораторной работы.

1. Создайте папку **cod_GPS_My** и скопируйте в ее все программы из папки **cod_GPS**.
2. Запустите MatLab, откройте функцию **cod_GPS**, изучите программные процедуры и комментарии, выполните **задания 1- 10**.
3. **Задание 1.** Откройте файл **PR1_cod_GPS.m**, сгенерируйте код спутника, определите, что получили в командном окне и занесите ответы в отчет.
4. **Задание 2.** Откройте файл **PR2_cod_GPS.m**, сгенерируйте коды спутников, определите, что получили в командном окне и занесите ответы в отчет.
5. **Задание 3.** Откройте файл **Pr3_cod_GPS.m**, сгенерируйте код спутника. Дайте описание полученного графического изображения. Описание внесите в отчет.
6. **Задание 4.** Откройте файл **Pr4_cod_GPS.m**, сгенерируйте код спутника. Дайте описание полученного графического изображения. Описание внесите в отчет.
7. **Задание 5.** Откройте файл **Pr5_cod_GPS.m**, сгенерируйте код спутника. Дайте описание полученного графического изображения. Описание внесите в отчет.
8. **Задание 6.** Откройте файл **Pr6_cod_GPS.m**, сгенерируйте код спутника. Дайте описание полученного графического изображения. Описание внесите в отчет.
9. **Задание 7.** Откройте файл **Pr7_cod_GPS.m**, сгенерируйте код спутника. Дайте описание полученного графического изображения. Описание внесите в отчет.
10. **Задание 8.** Откройте файл **Pr8_cod_GPS.m**, сгенерируйте код спутника. Дайте описание полученного графического изображения. Описание внесите в отчет.
11. **Задание 9.** Откройте файл **Pr09_cod_GPS.m**, сгенерируйте код спутника. Дайте описание полученного графического изображения. Описание внесите в отчет.

12. **Задание 10.** Откройте файл PrCodGPS.m, сгенерируйте код спутника. Дайте описание полученного графического изображения. Описание внесите в отчет.

1.4 Функции и файлы из папки cod_GPS

Функция cod_GPS (формирование псевдослучайного кода спутников GPS)

```
function [cod, cod_bin,column_7,n]=cod_GPS(Sv_id,n);
%{
Имя: cod_GPS
Функция предназначена для генерации псевдослучайного кода
любого спутника GPS генерация 37 C/A кодов
Sv_id - номер спутника
cod - вектор, содержащий выходную последовательность
g2s - вектор, содержащий соответствующее значения сдвига кода g2 для
получения кода C/A требуемого спутника (например,
для Sv_id=25 - используется сдвиг =513)
n- количество символов в последовательности
%}
%сдвиг кода в соответствии с ICD-GPS-200C для 37 значений

g2s=[5;6;7;8;17;18;139;140;141;251;252;254;255;256;257;258;...
    469;470;471;472;473;474;509;512;513;514;515;516;859;...
    860;861;862;863;950;947;948;950];
g2shift=g2s(Sv_id);
%{
генерация последовательности G1 согласно
n = 20;
%}
%начальное состояние регистра сдвига
shift_reg=-ones(1,10);
for i=1:n
    g1(i)=shift_reg(10); %последовательность G1
    % сложение (заменено умножением) по модулю 2 третьего и 10 разрядов
    % регистра
    modulo2=shift_reg(3)*shift_reg(10);
    % сдвиг разрядов 1:9
    shift_reg(2:10)=shift_reg(1:9);
    % присвоение modulo2 первому разряду регистра
    shift_reg(1)=modulo2;
end
g1;
```

```

% генерация кода G2 согласно
% начальное состояние регистра сдвига
shift_reg=-ones(1,10);
for i=1:n
    g2(i)=shift_reg(10); %последовательность G2
    % сложение (заменено умножением) по модулю 2 второго,3,6,8,9,10
    % разрядов регистра
    modulo2=shift_reg(2)*shift_reg(3)*shift_reg(6)*shift_reg(8)...
        *shift_reg(9)*shift_reg(10);
    % сдвиг разрядов 1:9
    shift_reg(2:10)=shift_reg(1:9);
    % присвоение modulo2 первому разряду регистра
    shift_reg(1)=modulo2;
end
g2;
[G1_G2] = [g1; g2]';
% сдвиг G2 на соответствующее номеру спутника число бит
g2tmp(1:g2shift)=g2(n-g2shift+1:n);
g2tmp(g2shift+1:n)=g2(1:n-g2shift);
%Сохранение сдвинутого кода
g2=g2tmp;
% формирование C/A кода поразрядным умножением G1 и G2
cod=g1.*g2;
%формирование кода спутника GPS в двоичных символах
cod_bin=-(cod-1)/2;
% формирование контрольного числа согласно в соответствии с ICD-GPS-200C
str=sprintf('%d',cod_bin(1:10));
column_7=dec2base(bin2dec(str),8,4);

```

Примеры применения функции cod_GPS.

Файл PR1_cod_GPS.m

```

%Пример PR1_cod_GPS.m
%Входные данные
Sv_id=2;% номер спутника
n=1023;% количество символов
[cod, cod_bin,column_7,n]=cod_GPS(Sv_id,n);
% Выходные данные
cod= cod(1:11) % 11 символов
cod_bin=cod_bin(1:11) % 11 символов в бинарном виде (0 и 1)
column_7% восьмеричное число первых 10 символов спутника № 2

```

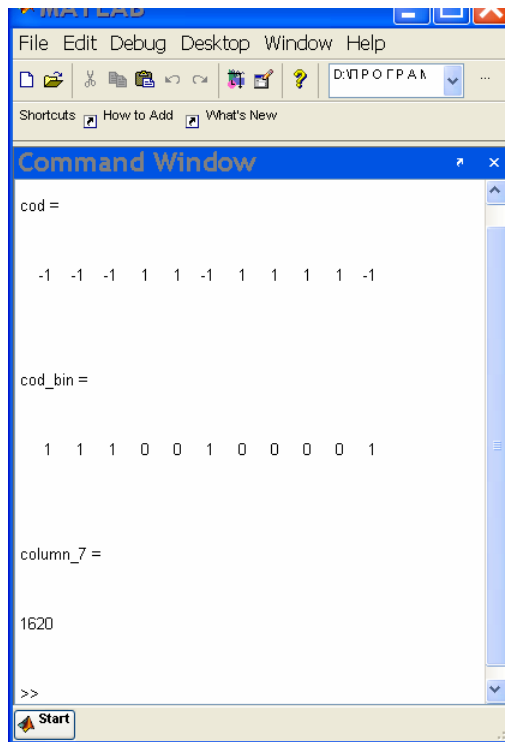


Рис. 1.4. Результат выполнения m- файла PR1_cod_GPS.m

Файл PR2_cod_GPS.m

```
clear
%Пример PR2_cod_GPS.m
%Расчет кодов любого или всех спутников
n=1023;
for Sv_id=1:37
[cod(Sv_id,:),cod_bin(Sv_id,:),col_7(Sv_id,:),n]=cod_GPS(Sv_id,n);
end
cod1=cod(1:23,10:30) %вывод в командное окно кодов 23 спутников в символах +1 и -1
cod2=cod_bin(1:23,10:30)' %вывод в командное окно кодов 23 спутников в символах 1 и 0
col_7(1:5,:) % вывод в командное окно восьмеричных значений кодов 1...5 спутников
%{
Результаты выполнения примера, скопированные из командного окна
cod1 =
```

```

1 -1 -1 -1 1 1 -1 1 -1 1 1 -1 1 1 -1 -1 -1 1 1 -1
1 -1 -1 -1 1 1 1 1 1 -1 -1 -1 -1 1 -1 1 1 -1 -1 1
1 -1 -1 -1 1 1 1 -1 1 1 1 1 -1 1 -1 1 1 -1 1 -1
1 -1 -1 -1 1 1 1 -1 -1 1 -1 -1 1 1 1 -1 -1 -1 -1 1
-1 1 1 1 -1 1 1 1 -1 -1 1 -1 -1 1 1 1 -1 1 -1 1 -1
-1 1 1 1 -1 -1 1 -1 1 -1 1 -1 -1 1 1 1 -1 1 1 1 1
-1 -1 -1 -1 -1 -1 -1 -1 -1 1 -1 1 1 -1 1 1 -1 -1 1 -1
1 1 -1 -1 1 -1 -1 1 -1 -1 -1 1 1 -1 -1 1 -1 1 -1 -1
1 -1 1 -1 1 1 -1 1 1 -1 1 1 1 -1 -1 -1 1 1 1 -1
1 -1 1 -1 1 1 1 -1 1 1 -1 -1 1 1 1 1 -1 -1 -1 1 1
1 1 -1 -1 1 -1 1 -1 -1 -1 1 1 1 -1 1 1 -1 1 1 1
```

```

1 -1 1 -1 1 1 -1 -1 -1 -1 1 -1 1 -1 -1 1 -1 1 1 -1 1
1 -1 -1 1 1 1 1 1 -1 -1 1 -1 -1 -1 -1 -1 1 1 -1 -1
-1 -1 -1 -1 -1 1 1 -1 1 -1 1 -1 -1 1 -1 -1 1 1 1 -1 -1
1 1 -1 -1 1 -1 1 -1 -1 1 1 -1 -1 1 1 -1 1 -1 1 -1 -1
1 1 1 1 1 1 1 1 -1 1 1 1 -1 1 1 -1 -1 -1 -1 -1 -1
-1 -1 1 1 -1 1 1 -1 1 -1 -1 -1 1 1 1 1 1 1 -1 1 -1
-1 1 -1 1 -1 -1 -1 -1 -1 -1 1 1 1 -1 -1 1 1 -1 1 1
-1 1 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 1 1 -1 1 -1 -1 -1 1
1 1 1 1 1 -1 -1 -1 -1 -1 1 1 -1 -1 1 -1 -1 1 1 1 -1
-1 -1 1 1 -1 1 -1 1 1 -1 1 -1 1 1 -1 1 1 1 1 -1 1
-1 -1 -1 1 -1 -1 -1 -1 1 1 1 1 1 -1 -1 1 1 -1 -1 -1

```

cod2 =

```

0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 0 1 1 1 0 1 1
1 1 1 1 0 0 1 0 1 1 1 0 1 1 1 0 0 1 0 0 0 1 1
1 1 1 1 0 0 1 1 0 0 1 1 0 1 1 1 0 0 1 0 0 0 1
1 1 1 1 0 0 1 1 1 1 0 1 1 0 1 1 0 0 0 1 0 0 0
0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 0 1 1 1 0 1 1
0 0 0 0 0 1 1 1 0 0 0 1 0 0 0 1 0 0 1 1 1 0 1
1 0 0 0 0 0 1 1 1 0 0 0 1 0 0 0 0 0 0 1 1 1 1
0 0 1 1 0 1 1 0 0 1 1 1 1 0 1 1 0 1 1 1 0 0 1
1 0 0 1 1 0 1 1 0 0 1 1 1 1 0 1 1 0 1 1 1 0 0
0 1 0 0 1 1 0 1 1 0 0 1 1 1 1 0 0 1 0 1 1 1 0
0 1 0 1 0 0 1 1 0 1 1 0 0 0 0 0 0 1 0 1 0 0 0
1 1 0 1 1 1 0 0 0 1 0 0 1 1 1 1 0 1 0 1 0 1 0
0 1 1 0 1 1 0 0 0 1 1 0 0 1 1 1 1 0 1 0 1 0 0
0 1 0 0 0 0 1 1 1 0 0 1 1 1 0 0 0 0 1 0 1 0 1
1 0 1 0 0 0 1 1 0 0 1 1 0 0 0 1 1 1 0 0 0 1 1
1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 0 0 1 0 1 1 0 0
1 0 0 1 1 1 1 1 1 1 0 1 1 1 1 0 0 1 0 1 1 0 0
1 0 1 1 0 0 1 0 0 1 1 0 0 0 0 1 1 0 1 0 0 0 1
0 1 0 1 1 0 0 1 0 1 1 0 0 0 0 1 1 0 1 0 0 1
0 1 0 1 0 0 1 1 0 0 0 0 1 1 1 1 1 0 0 1 0 1 1
1 0 1 0 1 0 0 1 1 1 0 0 0 1 1 1 1 1 0 0 1 0 1

```

ans =

1440

1620

1710

1744

1133

%{

Файл Pr3_cod_GPS.m

clear

%Пример Pr3_cod_GPS

%Визуализация кода GPS

n=1023;

for Sv_id=1:37

[cod(Sv_id,:),cod_bin(Sv_id,:),column_7,n]=cod_GPS(Sv_id,n);


```

col_7(Sv_id,:)= column_7;
end
M_P=cod_bin(2,1:30);
M_P1=cod(2,1:30);
stem(M_P), axis([ 0 30 -0.1 1.1])
xlabel('Код GPS в двоичных символах','FontName','Times New Roman ','FontSize',12)
grid
figure
stem(M_P1),axis([ 0 30 -1.1 1.1])
xlabel('Код GPS в символах "1","-1"', 'FontName','Times New Roman ','FontSize',12)
grid

```

Графическое изображение 30 первых символов кода спутника GPS приведено на рис.

1.5.

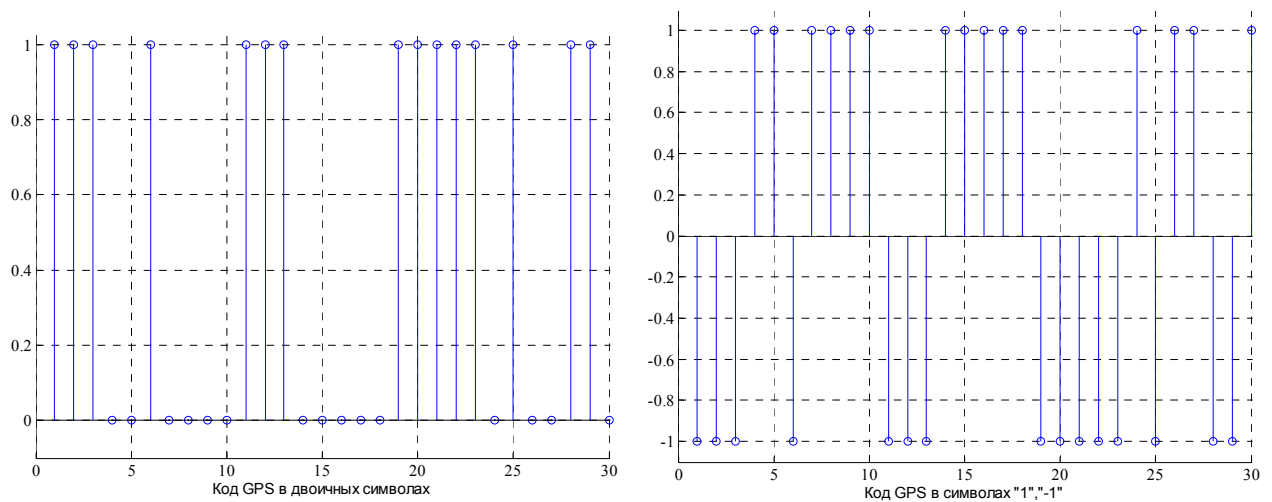


Рис. 1.5. Код спутника GPS

Файл Pr4_cod_GPS.m

```

clear
%Пример Pr4_cod_GPS.m
%Дискретизация кода спутника GPS
n=1023;
for Sv_id=1:37
[cod(Sv_id,:),cod_bin(Sv_id,:),column_7,n]=cod_GPS(Sv_id,n);
col_7(Sv_id,:)= column_7;
end
Cod3 = cod(3,:);
%Дискретизация
Fd=1.023*10^6;%символьная скорость
FsFd=2;% отношение частоты дискретизации к символьной скорости
Fs=Fd*FsFd;% частота дискретизации
t=(0:length(Cod3)*FsFd-1)/Fs;% дискретное время
F_mod1 = Cod3(floor(Fd*t)+1);%код

```

```

A_2=F_mod1(1:n);%код GPS
subplot(1,2,1), stem( cod(3,1:30)),axis([ 0 30 -1.1 1.1])
xlabel('a', 'FontSize',12, 'FontName','TimesNewRoman')
grid
subplot(1,2,2), stem( A_2(1:60)),axis([ 0 60 -1.1 1.1])
xlabel('b', 'FontSize',12, 'FontName','TimesNewRoman')
grid

```

Результат выполнения файла изображен на рис. 1.6.

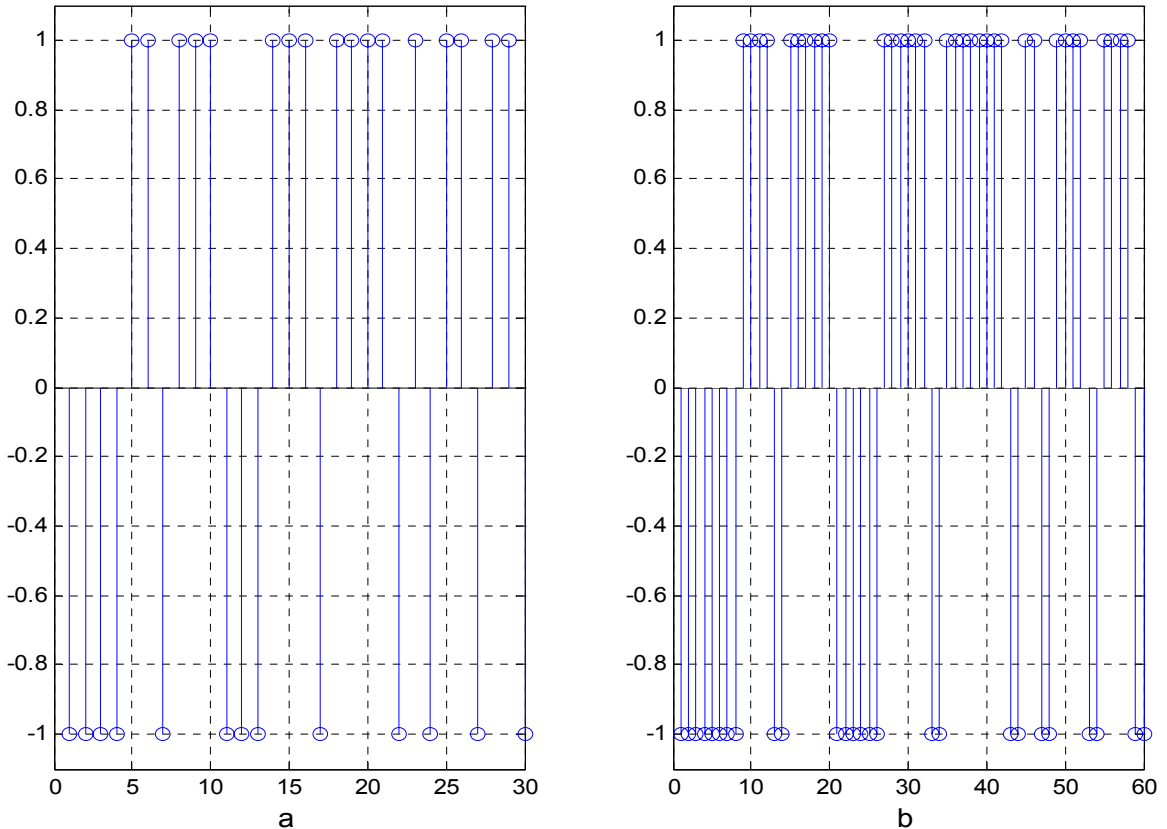


Рис. 1.6. Тридцать символов кода спутника GPS № 3: а - без дискретизации, б - частота дискретизации 2

Файл Pr5_cod_GPS.m

```

clear
%Имя файла:Pr5_cod_GPS.m
%Автокорреляционная функция кода GPS
n=1023;
for Sv_id=1:37
[cod(Sv_id,:),cod_bin(Sv_id,:),column_7,n]=cod_GPS(Sv_id,n);
col_7(Sv_id,:)= column_7;
end
Cod4 = cod(4,:);
%Дискретизация
Fd=1.023*10^6;%символьная скорость

```

```

FsFd=2;% отношение частоты дискретизации к символьной скорости
Fs=Fd*FsFd;% частота дискретизации
t=(0:length(Cod4)*FsFd-1)/Fs;% дискретное время
F_mod1 = (1/sqrt(FsFd))*Cod4(floor(Fd*t)+1);%код
m=n*FsFd;
nn=1023*FsFd;
A_2=F_mod1(1:m);%код GPS
[CorA2,lags]=xcorr(A_2,nn);%автокорреляция кода GPS
%Графика сигналов
subplot(1,2,1) ,plot(CorA2), axis([ 0 2*nn -100 1100])
xlabel('а', 'FontSize',12, 'FontName','TimesNewRoman')
grid on
subplot(1,2,2) , plot(CorA2), axis([ 0 nn -150 150])
xlabel('б', 'FontSize',12, 'FontName','TimesNewRoman')
grid on

```

Результат выполнения файла изображен на рис. 1.7.

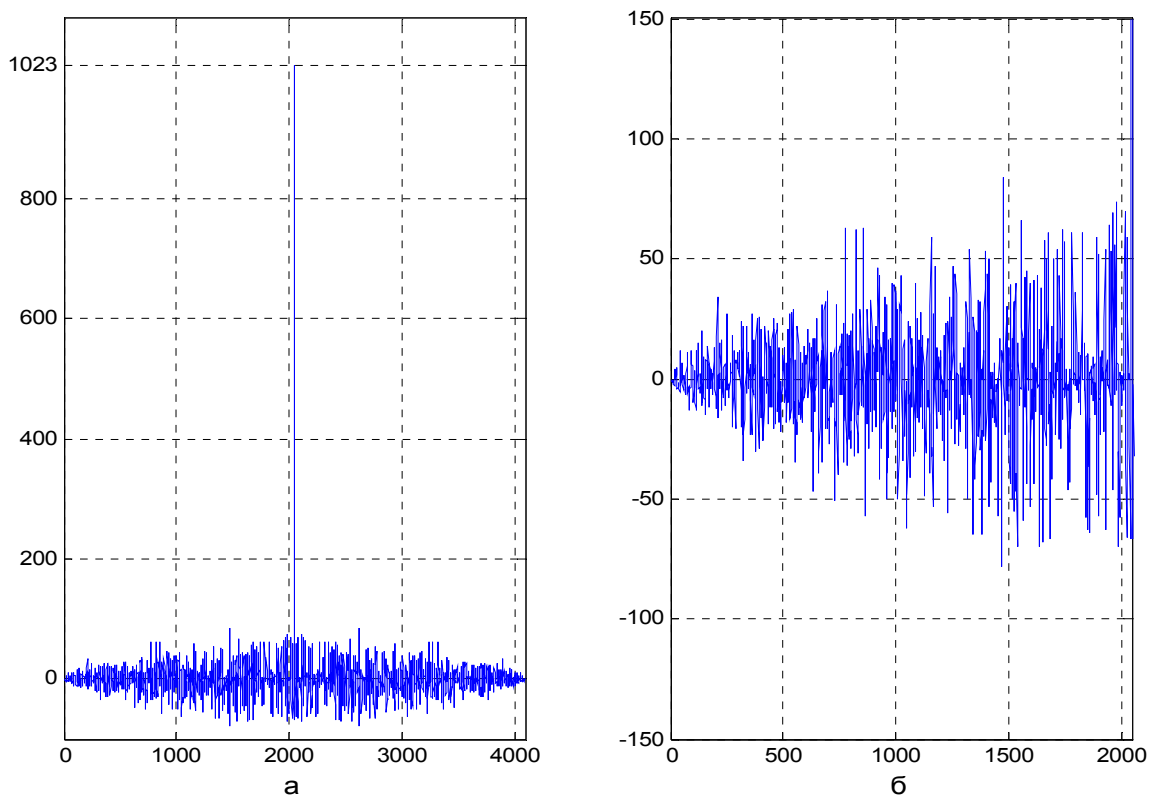


Рис. 1.7. Автокорреляционная функция кода GPS: а - автокорреляция кода сигнала спутника № 4; б - боковые лепестки функции автокорреляции

Файл Pr6_cod_GPS.m

```
clear
%Имя файла:Pr6_cod_GPS.m
%Взаимная корреляция кодов спутников GPS
n=1023;
for Sv_id=1:37
[cod(Sv_id,:),cod_bin(Sv_id,:),column_7,n]=cod_GPS(Sv_id,n);
col_7(Sv_id,:)= column_7;
end
Cod24 = cod(24,:);
Cod13 =cod(13,:);
%Dискретизация
Fd=1.023*10^6;%символьная скорость
FsFd=2;% отношение частоты дискретизации к символьной скорости
Fs=Fd*FsFd;% частота дискретизации
t=(0:length(Cod24)*FsFd-1)/Fs;% дискретное время
F_mod1 = (1/sqrt(FsFd))*Cod24(floor(Fd*t)+1);%код
F_mod3 = (1/sqrt(FsFd))*Cod13(floor(Fd*t)+1);%код
m=n;
m=n*FsFd;
nn=1023*FsFd;
A_2=F_mod1(1:m);%код GPS
A_3=F_mod3(1:m);%код GPS
Cor_13_24 = xcorr(A_2,A_3,nn);%взаимная корреляция кода 13 и 24
Cor_13_24_1 = xcorr(A_2(1:1023),A_3(1:1023),nn/2);
%Графика сигналов
subplot(1, 2, 1), plot(Cor_13_24),axis([ 0 2*nn -100 100])
xlabel('a ', 'FontSize',12,'FontName','TimesNewRoman')
grid on
subplot(1, 2, 2), plot(Cor_13_24_1),axis([0 nn -100 100])
xlabel('б ', 'FontSize',12,'FontName','TimesNewRoman')
grid on
```

Результат выполнения файла изображен на рис. 1.8.

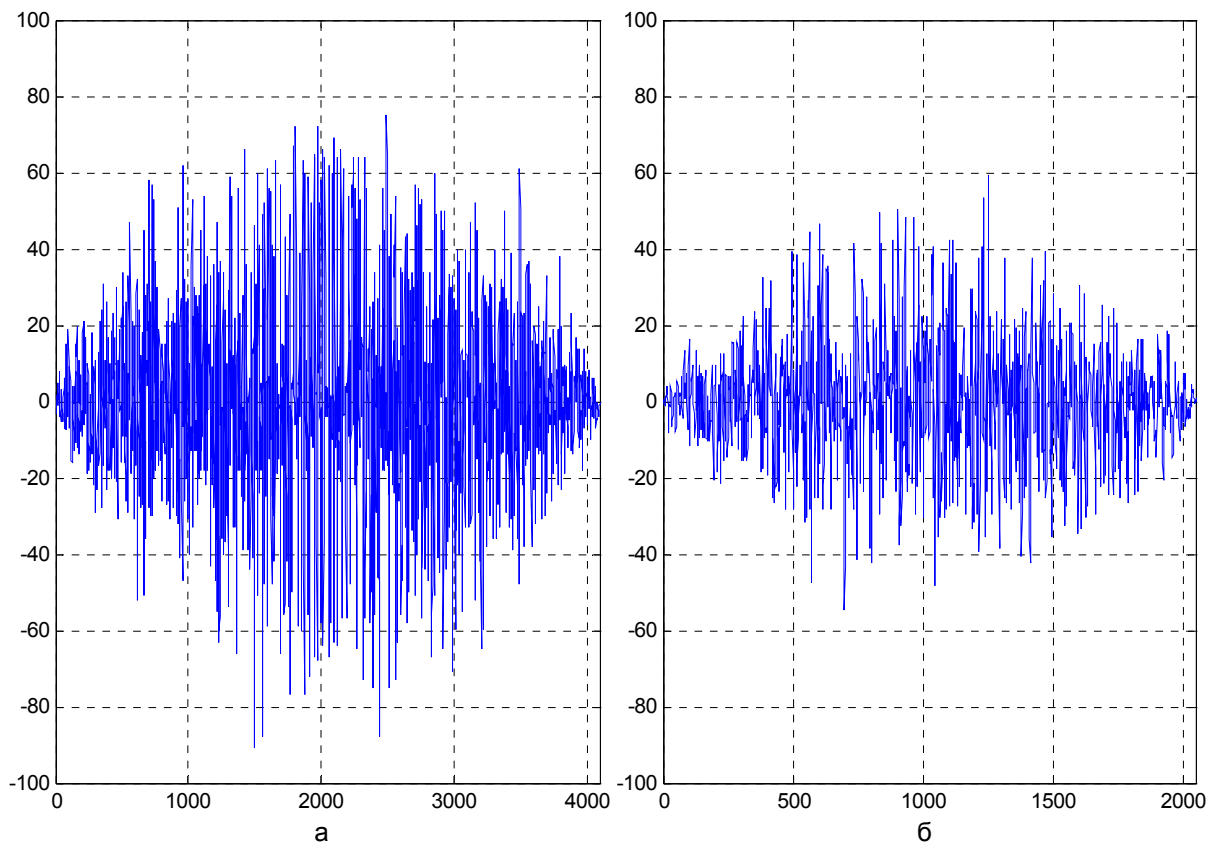


Рис. 1.8.. Взаимная корреляция сигналов спутников № 13 и 24: а - на всем интервале одного периода кода; б-на половине интервала одного периода кода

Файл Pr7_cod_GPS.m

```

clear
%Имя файла:Pr7_cod_GPS.m
%Взаимная корреляция кода и копии кода спутников GPS
n=1024*2;
for Sv_id=1:37
[cod(Sv_id,:),cod_bin(Sv_id,:),column_7,n]=cod_GPS(Sv_id,n);
col_7(Sv_id,:)= column_7;
end
Cod24 = cod(24,:);
%Дискретизация
Fd=1.023*10^6;%символьная скорость
FsFd=2;% отношение частоты дискретизации к символьной скорости
Fs=Fd*FsFd;% частота дискретизации
t=(0:length(Cod24)*FsFd-1)/Fs;% дискретное время
F_mod1 = (1/sqrt(FsFd))*Cod24(floor(Fd*t)+1);%код
m=n*FsFd;
nn=1023*FsFd;
A_2=F_mod1(1:m);%код GPS
Cor_24 = xcorr(A_2(500:2546),nn);%автокорреляция кода спутника

```

```

Cor_24_24_2 = xcorr(A_2(500:2546),A_2(500:2546),nn);%автокорреляция кода спутника с точной ко-
пией
Cor_24_24 = xcorr(A_2(498:2540),A_2(500:2546),nn);%взаимная корреляция кода с задержанной на 2
отсчета копией кода
Cor_24_24_1 = xcorr(A_2(502:2548),A_2(500:2536),nn);%взаимная корреляция кода с опережающей на
2 отсчета копией кода
%Графика сигналов
subplot(4, 1, 1), stem(Cor_24_24),axis([ 2040 2055 -100 1100])
xlabel('а ',FontSize,12,FontName,'TimesNewRoman')
grid on
subplot(4, 1, 2), stem(Cor_24),axis([ 2040 2055 -100 1100])
xlabel('б ',FontSize,12,FontName,'TimesNewRoman')
grid on
subplot(4,1,3), stem(Cor_24_24_2),axis([ 2040 2055 -100 1100])
xlabel('в ',FontSize,12,FontName,'TimesNewRoman')
grid on
subplot(4, 1, 4), stem(Cor_24_24_1),axis([2040 2055 -100 1100])
xlabel('г ',FontSize,12,FontName,'TimesNewRoman')
grid on

```

Результат выполнения файла изображен на рис. 1.9.

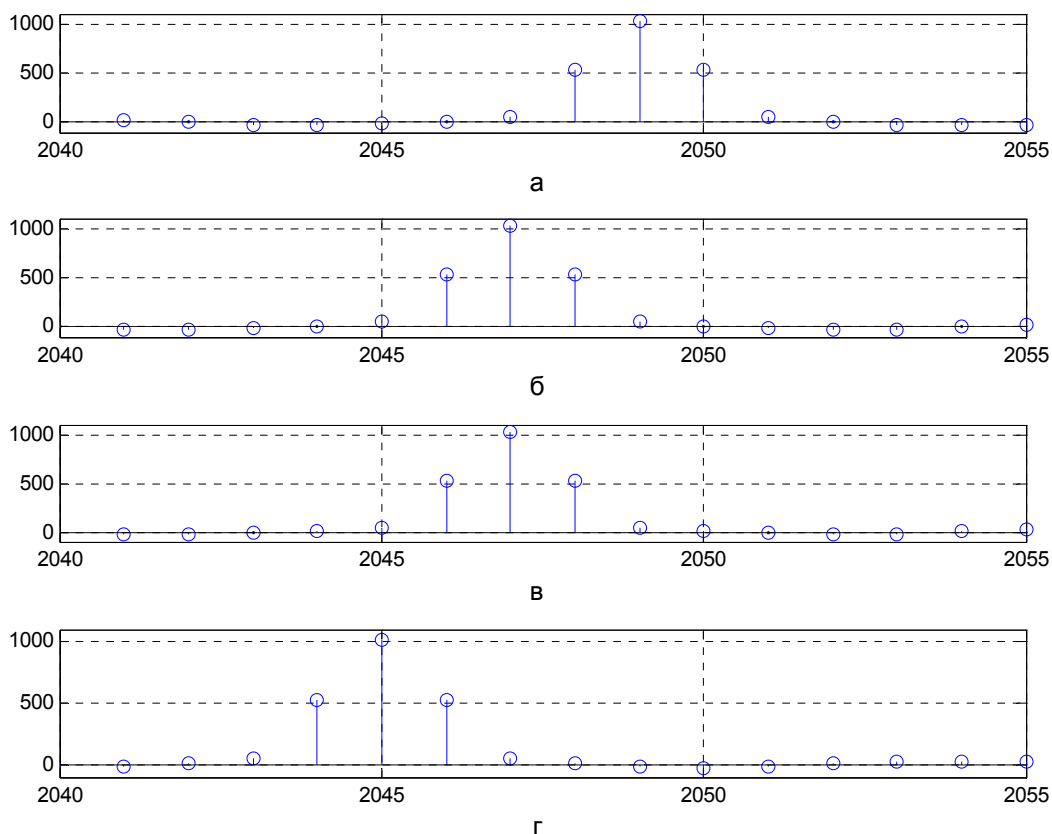


Рис. 1.9. Взаимная корреляция кода и копии кода сигнала спутников № 24:
а - копия опережает код на 2 отсчета; б - автокорреляция кода сигнала; в - взаимная корреляция кода и ко-
пии;
г-копия отстает от кода сигнала на 2 отсчета

Пример m- файла для исследования корреляционных свойств суммы сигналов.

Файл Pr8_CodGPS.m моделирует ситуацию, когда в зоне видимости находятся спутники, изображенные на рис. 6. 9. Спутники, данные которых обрабатывает приемник, имеют номера 2, 4, 5, 9,14, 17, 24, 30. Спутник 26 «не видим» приемником. Результаты выполнения m- файла изображены на рис. 1.10.

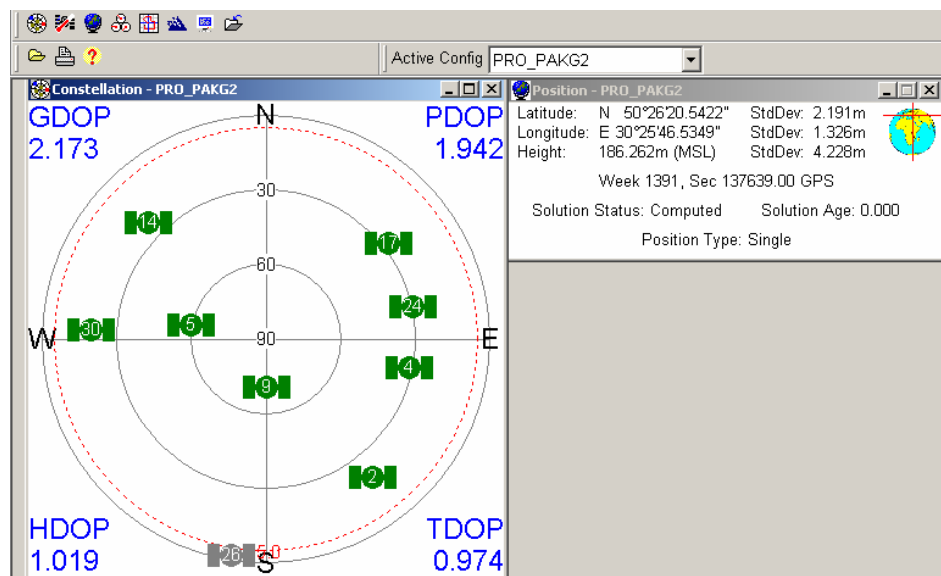


Рис. 1.10. Видимые спутники GPS №№ 2, 4, 5, 9,14, 17, 24, 30

Файл Pr8_cod_GPS.m

```
clear
%Имя файла:Pr8_CodGPS.m
% программа позволяет исследовать корреляционные свойства суммы сигналов
% спутников GPS
n=1023;
for Sv_id=1:37
[cod(Sv_id,:),cod_bin(Sv_id,:),column_7,n]=cod_GPS(Sv_id,n);
col_7(Sv_id,:)= column_7;
end
Cod9 = cod(9,:);
col_7(37,:);
Cod26 =cod(26,:);
M_P=cod_bin(37,:);
% Сумма кодов видимых спутников 2, 4, 5, 9,14, 17, 24, 30
Cod_Sym= cod(2,:)+cod(4,:)+cod(5,:)+cod(9,:)+cod(14,:)+...
cod(17,:)+cod(24,:)+cod(30,:);
%Dискретизация
Fd=1.023*10^6;%символьная скорость
FsFd=2;% отношение частоты дискретизации к символьной скорости
Fs=Fd*FsFd;% частота дискретизации
```

```

t=(0:length(Cod9)*FsFd-1)/Fs;% дискретное время
F_mod26 = (1/sqrt(FsFd))*Cod26(floor(Fd*t)+1);%код спутника 26, находящегося вне зоны видимости
F_mod9 = (1/sqrt(FsFd))*Cod9(floor(Fd*t)+1);%код
% сумма кодов 8 спутников 2, 4, 5, 9,14, 17, 24, 30
Sym_Cod = (1/sqrt(FsFd))*Cod_Sym(floor(Fd*t)+1);
%k=0;%сдвиг кода
m=n*FsFd;
nn=1023*FsFd;;
A_26=F_mod26(1:m);%код GPS
A_9=F_mod9(1:m);%код GPS
Cor_Sym = xcorr(Sym_Cod,nn);%автокорреляция сумма кодов 8 спутников
Cor_26_Sym = xcorr(Sym_Cod,A_26,nn);%взаимная корреляция
Cor_9_Sym = xcorr(Sym_Cod,A_9,nn);%взаимная корреляция
%Графика сигналов
subplot(4,1,1) , plot(Sym_Cod),axis([ 0 2100 -10 10])
xlabel('a','FontSize',12,'FontName','TimesNewRoman')
grid on
subplot(4,1,2) , plot(Cor_Sym),axis([ 0 2*nn -1000 10000])
xlabel('б','FontSize',12, 'FontName','TimesNewRoman')
grid on
subplot(4,1,3) , plot(Cor_26_Sym),axis([ 0 2*nn -400 400])
xlabel('в','FontSize',12, 'FontName','TimesNewRoman')
grid on
subplot(4,1,4) , plot(Cor_9_Sym),axis([ 0 2*nn -100 1100])
xlabel('г','FontSize',12, 'FontName','TimesNewRoman')
grid on

```

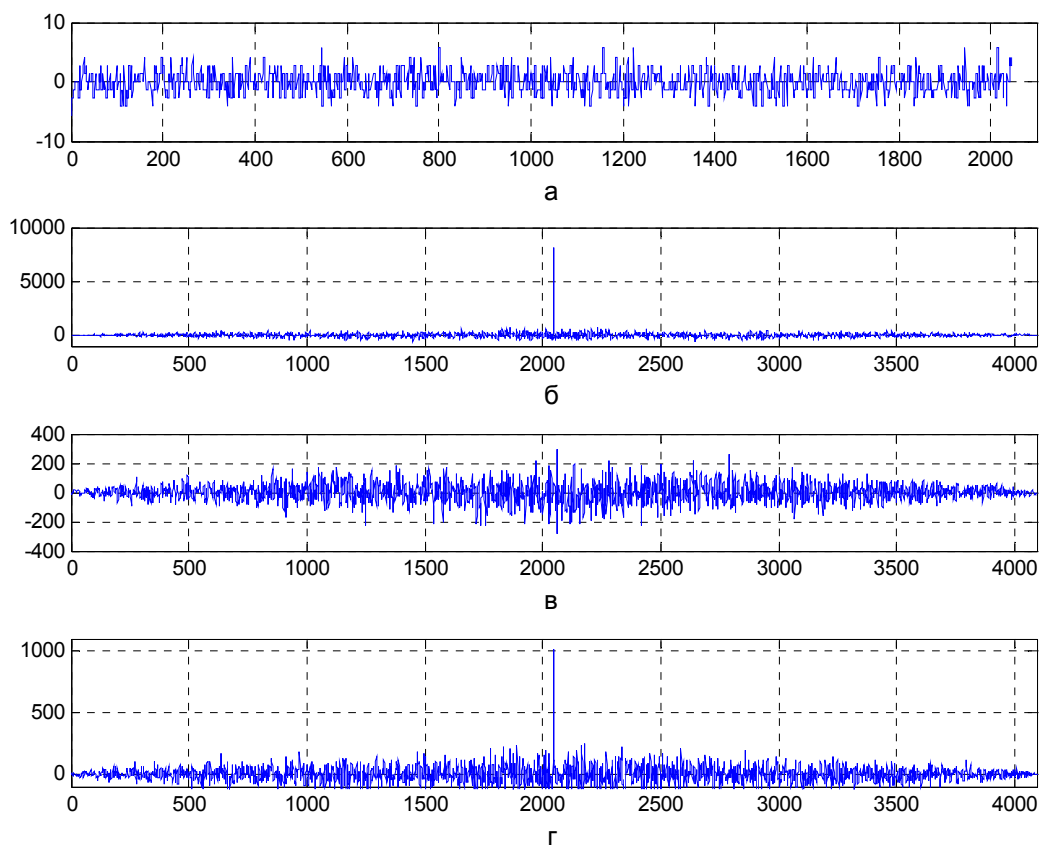



Рис. 1.11. Корреляционные характеристики сумм сигналов: а-сумма кодов 8 спутников; б-автокорреляция суммы кодов 8 спутников; в-корреляция суммы кодов 8 спутников и спутника 26; г-корреляция суммы кодов 8 спутников и спутника 9.

Пример *m*- файла для исследования корреляционных свойств сигнала с шумом и копии сигнала. В файле в качестве источника шума использована функция MatLab «*randn*». Программа иллюстрирует качественную картину влияния шума на взаимную корреляцию сигнала с шумом и копии сигнала. Для получения количественных значений в *m*- файл следует подставить конкретные значения соотношения сигнал/шум.

Файл *Pr09_cod_GPS.m*

```
clear
%Имя файла:Pr09_CodGPS.m
% программа взаимной корреляции сигнала с шумом и копии сигнала
% спутников GPS
n=1023;
for Sv_id=1:37
[cod(Sv_id,:),cod_bin(Sv_id,:),column_7,n]=cod_GPS(Sv_id,n);
col_7(Sv_id,:)= column_7;
end
Cod37 = cod(37,:);
col_7(37,:);
Noise=0;%коэффициент усиления генератора шума
Noise1= Noise*(randn(n,1))';%генерация шума
```

```

Noise2= 4*(randn(n,1));%генерация шума
Noise5= 8*(randn(n,1));%генерация шума
Noise10=12*(randn(n,1));%генерация шума
%Дискретизация
Fd=1.023*10^6;%символьная скорость
FsFd=2;% отношение частоты дискретизации к символьной скорости
Fs=Fd*FsFd;% частота дискретизации
t=(0:length(Cod37)*FsFd-1)/Fs;% дискретное время
Cod37 = (1/sqrt(FsFd))*Cod37(floor(Fd*t)+1);%код
Noise_mod=(1/sqrt(FsFd))*Noise1(floor(Fd*t)+1);%шум
Noise_mod2=(1/sqrt(FsFd))*Noise2(floor(Fd*t)+1);%шум
Noise_mod5=(1/sqrt(FsFd))*Noise5(floor(Fd*t)+1);%шум
Noise_mod10=(1/sqrt(FsFd))*Noise10(floor(Fd*t)+1);%шум
%Noise_mod=(1/sqrt(FsFd))*Noise1(floor(Fd*t)+1);%шум
%Noise_mod=(1/sqrt(FsFd))*Noise1(floor(Fd*t)+1);%шум
Cod37Noise = Cod37 +Noise_mod;% код плюс шум
Cod37Noise2 = Cod37 +Noise_mod2;% код плюс шум
Cod37Noise5 = Cod37 +Noise_mod5;% код плюс шум
Cod37Noise10 = Cod37 +Noise_mod10;% код плюс шум
m=n*FsFd;
nn=1023*FsFd;;
Cod37Noise = Cod37Noise(1:m);%код GPS+ шум
Cod37Noise2= Cod37Noise2(1:m);%код GPS+ шум
Cod37Noise5= Cod37Noise5(1:m);%код GPS+ шум
Cod37Noise10= Cod37Noise10(1:m);%код GPS+ шум
Cod37=Cod37(1:m);%код GPS
Cor_Cod37_Noise = xcorr(Cod37,Cod37Noise ,nn);%взаимная корреляция копии кода GPS и сигнала с шумом
Cor_Cod37_Noise2 = xcorr(Cod37,Cod37Noise2 ,nn);%взаимная корреляция копии кода GPS и сигнала с шумом
Cor_Cod37_Noise5 = xcorr(Cod37,Cod37Noise5 ,nn);%взаимная корреляция копии кода GPS и сигнала с шумом
Cor_Cod37_Noise10 = xcorr(Cod37,Cod37Noise10 ,nn);%взаимная корреляция кода GPS и шума

%Графика
subplot(4,1,1) ,stem(Cor_Cod37_Noise), axis([1500 2500 -100 1200])
xlabel('a','FontSize',12,'FontName','TimesNewRoman')
grid on
subplot(4,1,2) ,stem(Cor_Cod37_Noise2 ),axis([ 1500 2500 -100 1300])
xlabel('б','FontSize',12,'FontName','TimesNewRoman')
grid on
subplot(4,1,3) ,stem( Cor_Cod37_Noise5),axis([ 1500 2500 -100 1300])

```

```

xlabel('b','FontSize',12,'FontName','TimesNewRoman')
grid on
subplot(4,1,4) ,stem(Cor_Cod37_Noise10),axis([ 1500 2500 -100 1300])
xlabel('r','FontSize',12,'FontName','TimesNewRoman')
grid on

```

Результаты выполнения *m*- файла изображены на рис. 1.12.

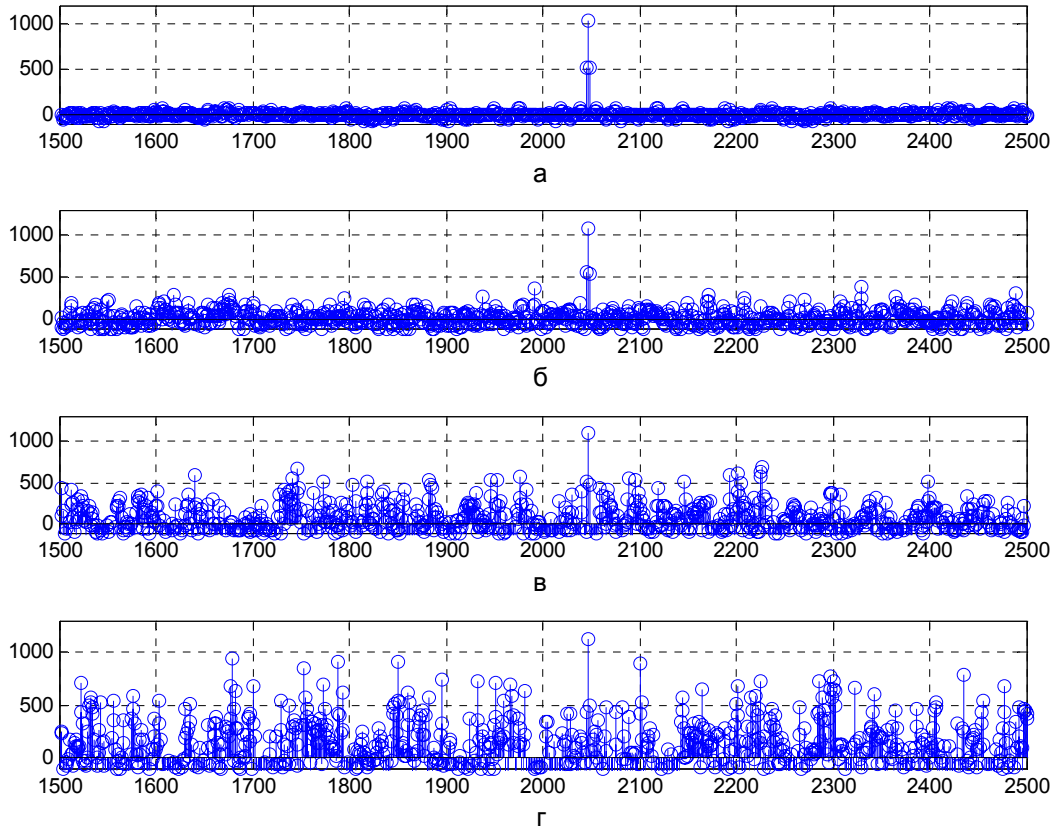


Рис. 1.12. Корреляция сигнала с шумом и копии сигнала: а-шум отсутствует; б-шум в 4 раза больше сигнала; в - шум в 8 раз больше сигнала; г- шум в 12 раз больше сигнала.

Комплексная программа обработки псевдослучайных кодов спутников GPS. Приведенный ниже *m*- файл позволяет одновременно выполнить все процедуры обработки псевдослучайных кодов спутников GPS, рассмотренные в предыдущих примерах.

Файл PrCodGPS S.m

```

clear
%Имя файла:PrCodGPS.m
% Комплексная программа обработки псевдослучайных кодов спутников GPS.
n=1023;
for Sv_id=1:37
[cod(Sv_id,:),cod_bin(Sv_id,:),column_7,n]=cod_GPS(Sv_id,n);
col_7(Sv_id,)= column_7;
end

```

```

Cod37 = cod(37,:);
col_7(37,:);
Cod37;
Cod13 =cod(13,:);
M_P=cod_bin(37,:);
Cod_Sym= cod(1,:)+cod(4,:)+cod(7,:)+cod(11,:)+cod(13,:)+cod(14,:)+...
    cod(20,:)+cod(21,:)+cod(23,:)+cod(25,:)+cod(31,:);
Noise=1;%коэффициент усиления генератора шума
Noise1= Noise*(randn(2*n,1))';%генерация шума
%для просмотра результатов выполнения программы убрать символ ";"
%Dискретизация
Fd=1.023*10^6;%символьная скорость
FsFd=2;% отношение частоты дискретизации к символьной скорости
Fs=Fd*FsFd;% частота дискретизации
t=(0:length(Cod37)*FsFd-1)/Fs;% дискретное время
Noise_mod=(1/sqrt(FsFd))*Noise1(floor(Fd*t)+1);
F_mod = (1/sqrt(FsFd))*Cod37(floor(Fd*t)+1)+Noise_mod;% код плюс шум
F_mod1 = (1/sqrt(FsFd))*Cod37(floor(Fd*t)+1);%код
F_mod2 = (1/sqrt(FsFd))*Cod37(floor(Fd*t)+1);%код
F_mod3 = (1/sqrt(FsFd))*Cod13(floor(Fd*t)+1);%код
% сумма кодов 11 спутников 1,4,7,11,13,14,20,21,23,25,31
Sym_Cod = (1/sqrt(FsFd))*Cod_Sym(floor(Fd*t)+1);
k=0;%сдвиг кода
m=n*FsFd;
nn=1023*FsFd;
A_0 = F_mod(1:m);%код GPS+ шум
A_1 = F_mod1(1+k:m+k);% сдвинутый код
A_2=F_mod1(1:m);%код GPS
A_3=F_mod3(1:m);%код GPS
Cor = xcorr(A_2,Noise_mod,nn);%взаимная корреляция кода GPS и шума
CorA0 = xcorr(A_0,nn);%автокорреляция кода GPS и шума
[CorA1,lags] = xcorr(A_1,nn);%автокорреляция сдвинутого код
Cor_R = xcorr(Noise_mod,nn);%автокорреляция шума
[CorA2,lags]=xcorr(A_2,nn);%автокорреляция кода GPS
Cor_GRG = xcorr(A_0,A_1,nn);%взаимная корреляция кода GPS+ шум и
%сдвинутого кода
Cor_13_37 = xcorr(A_2,A_3,nn);%взаимная корреляция кода 13 и 37
Cor_Sym = xcorr(Sym_Cod,nn);%автокорреляция сумма кодов 11 спутников
Cor_13_Sym = xcorr(Sym_Cod,A_3,nn);%взаимная корреляция суммы кодов 11 спутников и кода
спутника 13
%Графика сигналов
%%%%%%%%%%%%%%

```

```

fig1=figure;
subplot(4,1,1) ,plot(M_P), axis([ 0 1100 -0.1 1.1])
xlabel ('Код GPS в двоичных символах','FontSize',12,...
    'FontName','TimesNewRoman')
grid on
subplot(4,1,2) , plot(Cod37),axis([ 0 1100 -1.1 1.1])
xlabel('Код GPS в символах "1","-1"', 'FontSize',12,...
    'FontName','TimesNewRoman')
grid on

subplot(4,1,3) ,plot(Noise_mod),axis([ 0 2*n -3 3])
xlabel('Псевдослучайный шум','FontSize',12,'FontName','TimesNewRoman')
grid on
subplot(4,1,4) ,plot(A_0 ),axis([ 0 2*n -5 5])
xlabel('Код GPS плюс шум','FontSize',12,'FontName','TimesNewRoman')
grid on
%Графика обработки сигналов
figure(fig1);
fig2=figure;
subplot(4,1,1) ,plot(CorA2), axis([ 0 2*nn -100 1100])
xlabel('Автокорреляционная функция кода GPS','FontSize',12,...
    'FontName','TimesNewRoman')
grid on
subplot(4,1,2) , plot(lags,CorA2), axis([ 0 nn -150 150])
xlabel ('Боковые лепестки автокорреляционной функции кода GPS',...
    'FontSize',12,'FontName','TimesNewRoman')
grid on

subplot(4,1,3) ,plot(Cor_R ), axis([ 0 2*nn -100 1100])
xlabel('Автокорреляционная функция шума','FontSize',12,'FontName',...
    'TimesNewRoman')
grid on
subplot(4,1,4) ,plot(Cor_R ),axis([ 0 nn -150 150])
xlabel ('Боковые лепестки автокорреляционной функции шума','FontSize',...
    12,'FontName','TimesNewRoman')
grid on
fig3=figure;
subplot(4,1,1) ,plot(Cor),axis([ 0 2*nn -150 150])
xlabel('Взаимная корреляция кода GPS и шума','FontSize',...
    12,'FontName','TimesNewRoman')
grid on
subplot(4,1,2) ,plot(CorA0),axis([ 0 2*nn -100 2500])

```

```

xlabel('Автокорреляционная функция кода GPS плюс шум','FontSize',...
      12,'FontName','TimesNewRoman')
grid on
subplot(4,1,3) ,plot(Cor_GRG),axis([ 0 2*nn -100 1100])
xlabel ('Взаимная корреляция кода GPS плюс шум и копии кода GPS ',...
      'FontSize',12,'FontName','TimesNewRoman')
grid on
subplot(4,1,4) ,plot(Cor_GRG),axis([ 0 2*n -200 200])
xlabel ('Взаимная корреляция кода GPS плюс шум и копии кода GPS ',...
      'FontSize',12,'FontName','TimesNewRoman')
grid on
fig4=figure;
subplot(4,1,1) , plot(Sym_Cod),axis([ 0 2*n -10 10])
xlabel('Сумма кодов 11 спутников','FontSize',12,'FontName','TimesNewRoman')
grid on
subplot(4,1,2) , plot(Cor_Sym),axis([ 0 2*nn -1000 2500])
xlabel ('Автокорреляция суммы кодов 11 спутников','FontSize',12,...
      'FontName','TimesNewRoman')
grid on
subplot(4,1,3) , plot(Cor_13_Sym),axis([ 0 2*nn -100 1500])
xlabel ('Взаимная корреляция суммы кодов 11 спутников и кода 13',...
      'FontSize',12,'FontName','TimesNewRoman')
grid on
subplot(4,1,4) , plot(Cor_13_Sym),axis([ 0 2*n -500 600])
xlabel ('Боковые лепестки взаимной корреляции суммы кодов 11 спутников и кода 13',...
      'FontSize',12,'FontName','TimesNewRoman')
grid on

fig5=figure;
plot(Cor_13_37),axis([ 0 2*nn -100 100])
xlabel('Взаимная корреляция кода GPS (коды 13и 37) ',...
      'FontSize',12,'FontName','TimesNewRoman')
grid on

```

Графические результаты выполнения m- файла *PrCodGPS.m* изображены на рис. 1.13 ...рис. 1.17.

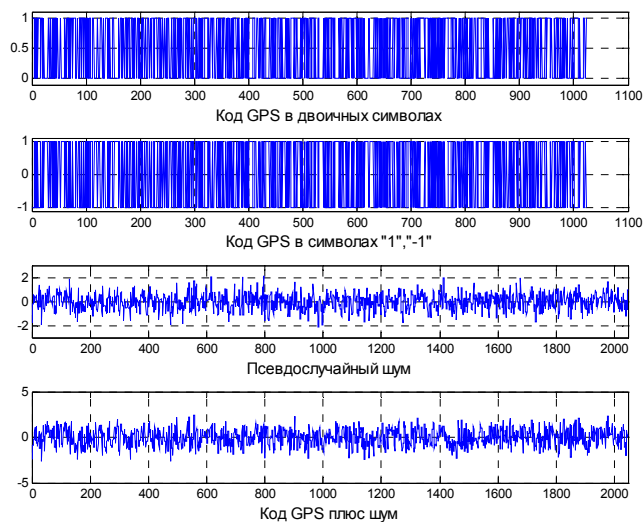


Рис. 1.13. Графика файла PrCodGPS.m- Figure 1

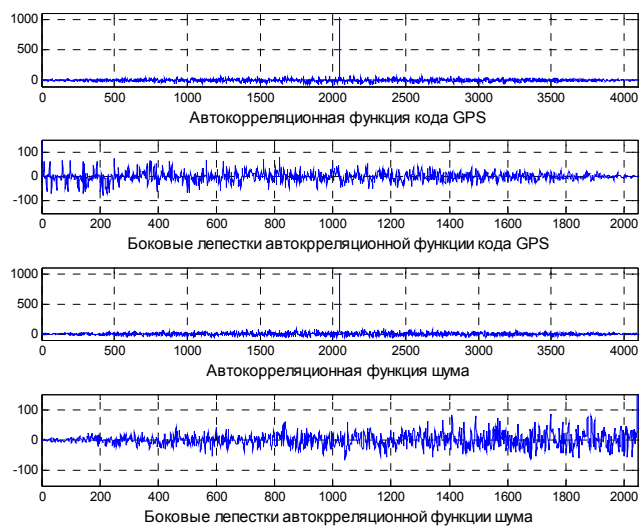


Рис. 1.14.. Графика файла PrCodGPS.m- Figure 2

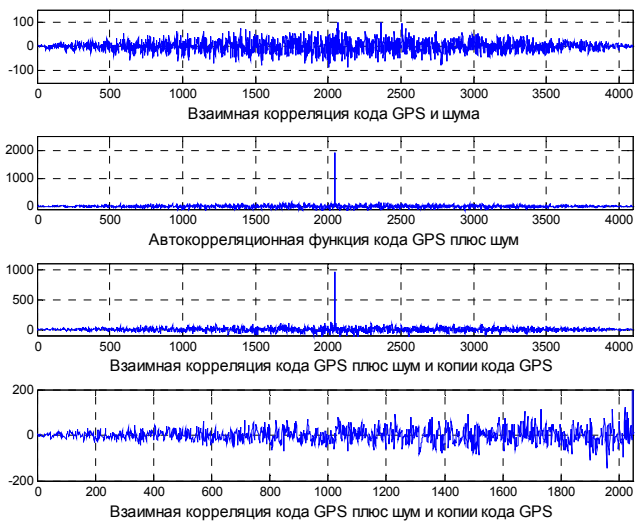


Рис. 1.15.. Графика файла PrCodGPS.m- Figure 3

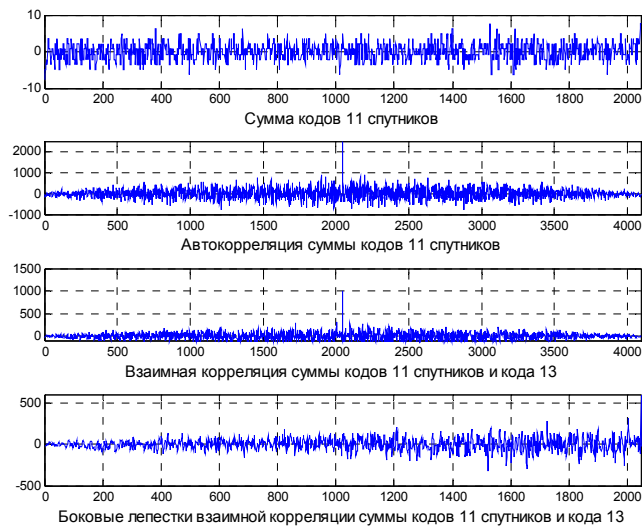


Рис. 1.16. Графика файла PrCodGPS.m- Figure 4

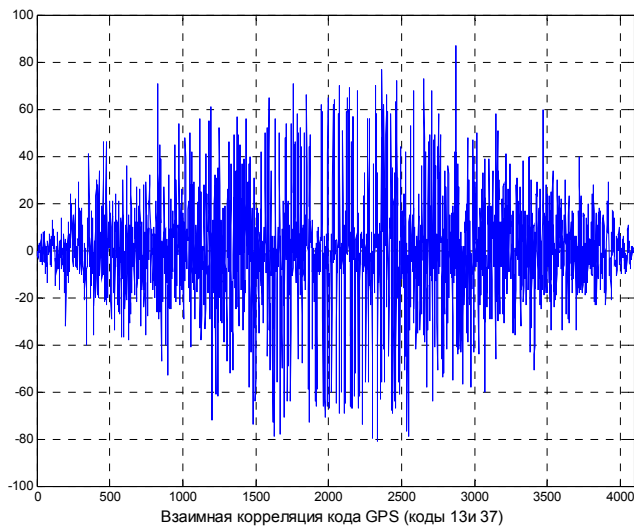


Рис. 1.17. Графика файла PrCodGPS.m- Figure 5

1.5 Псевдслучайный код спутников ГЛОНАСС

1.5.1 Лабораторная работа 1. 3 «Код спутников ГЛОНАСС»

Цель лабораторной работы- формирование и исследование псевдслучайного кода спутников ГЛОНАСС.

Рекомендуется следующий порядок выполнения лабораторной работы.

1. Создайте папку **CodGLONASS_My** и скопируйте в нее все программы из папки **CodGLONASS**.
2. Запустите MatLab, откройте функцию `cod_GLONASS`, изучите программные процедуры и комментарии, выполните **задание**.
3. **Задание.** Откройте файл `BPS_K.m`. Выполните файл. Опишите полученные графики и занесите описание графиков в отчет.

1.6 Функции и файлы из папки CodGLONASS

Функция *cod_GLONASS* (формирование M- последовательности спутников ГЛОНАСС)

```
function [codGL, codGL_bin,Out,n]=cod_GLONASS(n);
%Имя функции: cod_GLONASS
%Функция cod_GLONASS вычисляет псевдослучайный код ГЛОНАСС
%(M-последовательность). Входные данные: n- количество символов в M-последовательности.
%Выходные данные: codGL- M- последовательность в символах "1", "-1",
%codGL_bin- M-последовательности в символах "0", "1",
%Out- столбцы кодов для сравнения, n- длина кода
shift_reg=ones(1,9); %Начальное состояние регистра сдвига
for i=1:n
    codGL_bin(i)=shift_reg(7); % Выход M-последовательности в символах "0", "1"
    modulo2 = xor(shift_reg(9),shift_reg(5)); %Сложение по модулю 2 символов с 9 и 5 выходов регистра
    сдвига
    shift_reg(2:9)=shift_reg(1:8); %Смещение данных в регистре сдвига
    shift_reg(1)=modulo2; % Сложение по модулю 2 на входе 1 регистра сдвига
end
codGL=2*codGL_bin-1;%M- последовательность в символах "1", "-1"
Out=[codGL_bin' codGL']; % Столбцы кодов для сравнения
```

Файла: *BPS_K.m*

```
%Имя m-файла: BPS_K.m
%программа расчета характеристик сигнала спутника ГЛОНАСС
clear;
n=511;%количество символов в M-последовательности
[codGL, codGL_bin,Out,n]=cod_GLONASS(n);%функция, формирующая M-последовательность

Dat= codGL_bin;

fDat=0.511*10^6;% скорость передачи данных
fc=4*fDat ; %carrier frequency-несущая частота
fd=32*fDat; %частота дискретизации
fd_fDat=fd/fDat; %отношение частоты дискретизации к символьной скорости
t=(0:length(Dat)*fd_fDat-1)/fd; % дискретное время
t1=(0:length(codGL)*fd_fDat-1)/fd;% дискретное время
BPSK=cos(2*pi*fc*t + pi*Dat(floor(fDat*t)+1));%модулированный сигнал
%y= pmmmod(Dat(floor(fDat*t)+1),fc,fd,pi) ;%модулированный сигнал (вариант)
y=BPSK;
%преобразование частоты
x1=y.*cos(2*pi*fc*t);
```

```

x=y.*cos(2*pi*fc*t);
[b1,a1]=butter(3,fc*2/fd);%фильтр нижних частот с максимально-плоской характеристикой
[b,a]=ellip(10,2,60,fc*2/(32*fDat));%фильтр нижних частот с эллиптической характеристикой
%фильтрация высокочастотных составляющих
x1=filtfilt(b1,a1,x1);
x=filtfilt(b,a,x);
m=3.20;
%Графика
fig1=figure;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Сигналы спутников ГЛОНАСС
%Несущая
subplot(3,1,1),plot(t,cos(2*pi*fc*t))
xlabel('a','FontSize',12,'FontName','TimesNewRoman')
grid on
xlim([3.07*10^(-4) m*10^(-4)])
ylim([-1.5 1.5])
%Несущая, модулированная М-последовательностью
subplot(3,1,2),plot(t,BPSK)
xlabel('б','FontSize',12,'FontName','TimesNewRoman')
grid on
xlim([3.07*10^(-4) m*10^(-4)])
ylim([-1.5 1.5])
%M-последовательность
subplot(3,1,3),stem(t,Dat(floor(fDat*t)+1))
xlabel('в','FontSize',12,'FontName','TimesNewRoman')
grid on
xlim([3.07*10^(-4) m*10^(-4)])
ylim([-1.5 1.5])

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Фильтрация демодулированного сигнала ГЛОНАСС фильтром нижних частот с
%максимально плоской и эллиптической характеристиками
m=5;

%figure(fig1);
fig2=figure;
subplot(3,1,1), plot(t,Dat(floor(fDat*t)+1))
xlabel('a','FontSize',12,'FontName','TimesNewRoman')
grid on
xlim([3.07*10^(-4) m*10^(-4)])
ylim([-1.5 1.5])

```

```

subplot(3,1,2), plot(t,x1)
xlabel('б','FontSize',12,'FontName', 'TimesNewRoman')
grid on
xlim([3.07*10^(-4) m*10^(-4)])
ylim([-1.5 1.5])
subplot(3,1,3), plot(t,x)
xlabel('в','FontSize',12,'FontName', 'TimesNewRoman')
grid on
xlim([3.07*10^(-4) m*10^(-4)])
ylim([-1.5 1.5])
%Фильтры
fig3=figure;
freqz(b1,a1,512,32*fDat),
xlabel('а','FontSize',12,'FontName', 'TimesNewRoman')
fig4=figure;
freqz(b,a,512,32*fDat),
xlabel('а','FontSize',12,'FontName', 'TimesNewRoman')

```

Результаты выполнения файла -:BPS_K.m приведены на рис. 1.18...рис. 1.21.

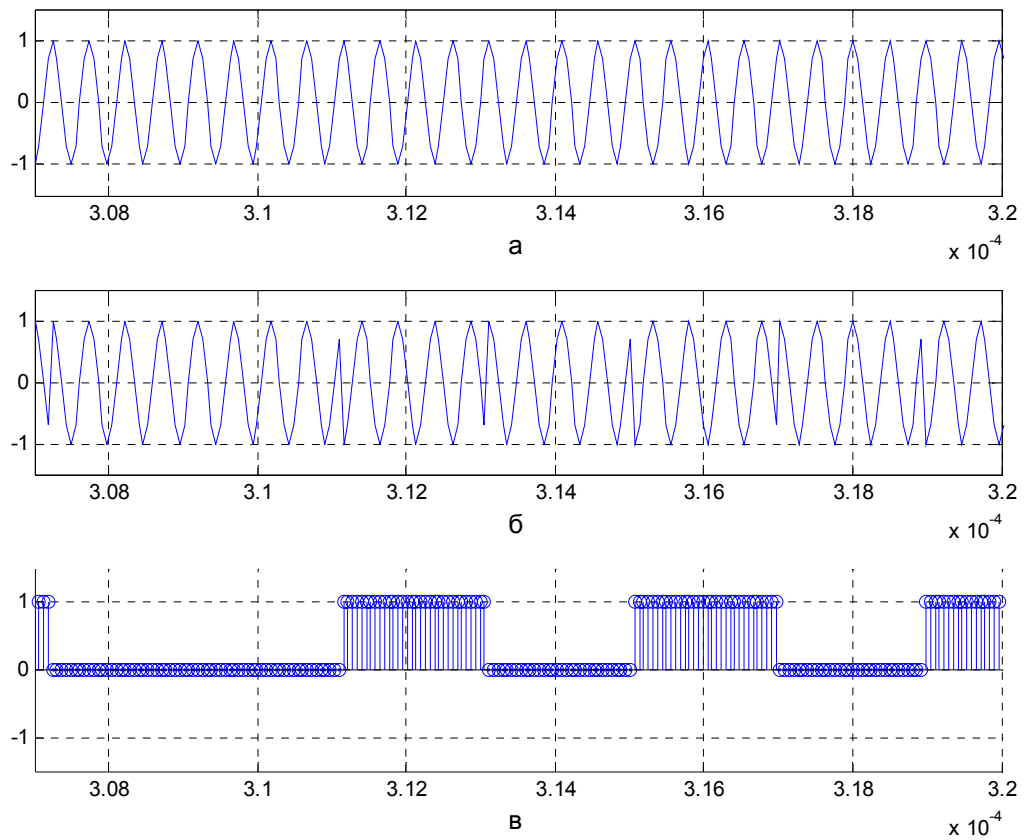


Рис. 1.18. Сигнал ГЛОНАСС- Figure 1: а- несущая; б- модулированная несущая;
в- выборка дискретизированной М-последовательности

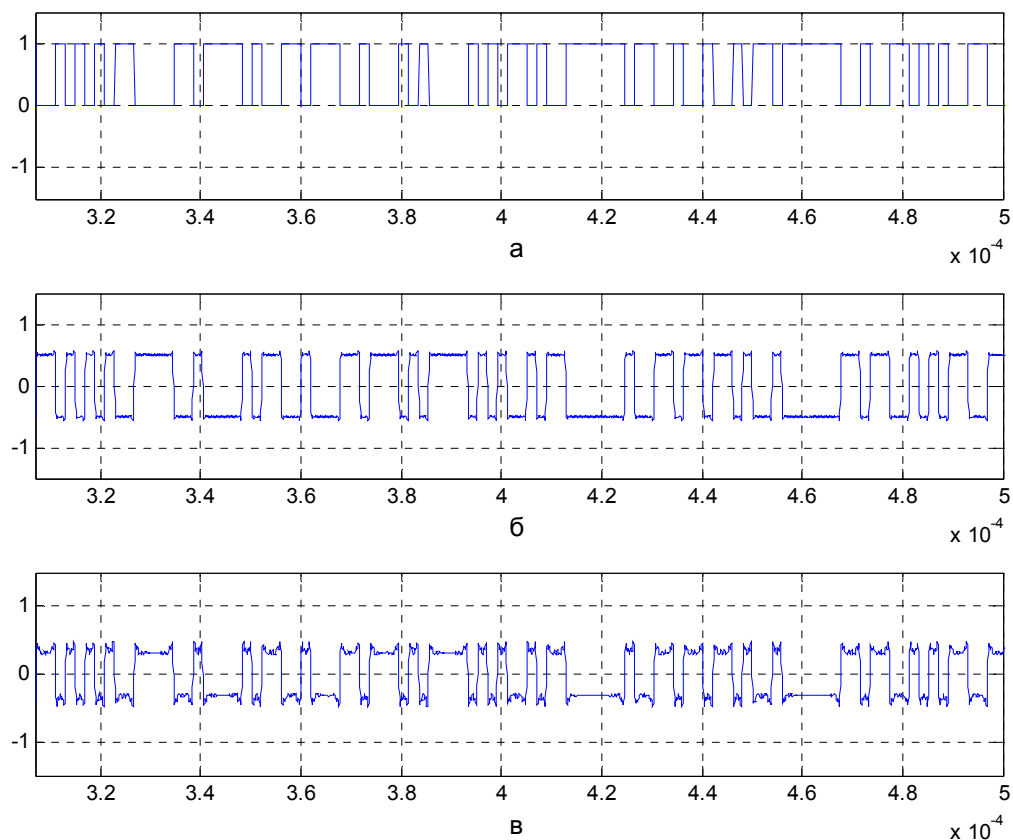


Рис. 1.19. Сигнал ГЛОНАСС- Figure 2: а- M-последовательность до модуляции; б- M-последовательность после модуляции и фильтрации фильтром с максимально плоской характеристикой; в- M-последовательность после модуляции и фильтрации фильтром с эллиптической характеристикой

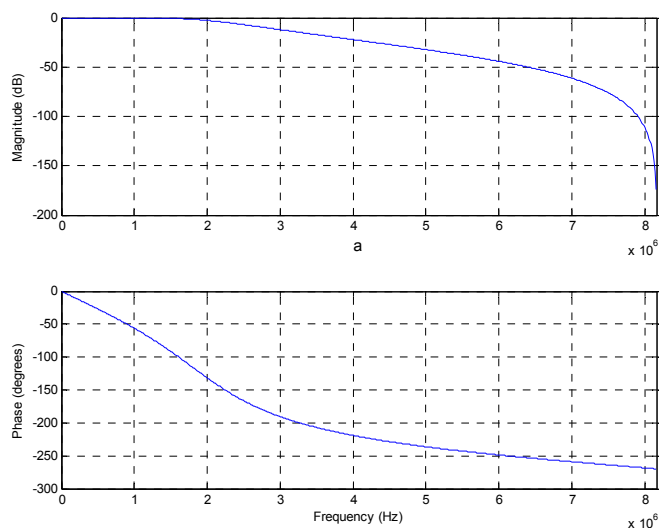


Рис. 1.20. Фильтр с максимально плоской характеристикой- Figure 3: верхний график- АЧХ; нижний график-ФЧХ

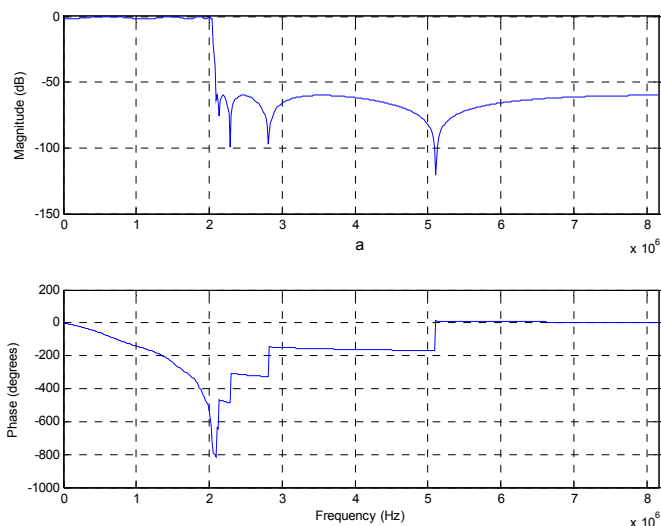


Рис. 1.21. Фильтр с эллиптической характеристикой- Figure 4: верхний график- АЧХ; нижний график-ФЧХ

1.7 Вопросы и задания для самостоятельной подготовки

1. Запишите математическую процедуру «сложение по модулю 2».
2. При каких условиях «сложение по модулю 2» можно заменить умножением?
3. Что такое псевдослучайный сигнал?
4. Есть ли разница между псевдослучайным сигналом и M- последовательностью?
5. Для чего кодируются сигналы навигационных спутников?
6. Запишите образующий полином для сигналов спутников ГЛОНАСС.
7. Запишите образующие полиномы для сигналов спутников GPS.
8. Какая связь между регистрами сдвига, формирующими коды сигналов спутников и образующими полиномами?
9. Опишите механизм формирования псевдослучайного кода спутников GPS.
10. Опишите механизм формирования M- последовательности спутников ГЛОНАСС.
11. Что обозначает понятие кодовое разделение сигналов?
12. Как идентифицируются спутники GPS?
13. Как идентифицируются спутники ГЛОНАСС?
14. Как понимать термин «отношение сигнал/шум»?
15. Опишите параметры функции корреляции.

РАЗДЕЛ 2 Преобразование координат

2.1 Краткие сведения из теории

В спутниковых системах радионавигации применяются различные системы координат для расчета орбитального движения спутников и позиции потребителя. Системы координат, методы и алгоритмы их расчета и преобразования, на основании которых разработаны программы данного раздела изложены в книге [1] (раздел 1. 3, стр. 33 -40), в руководстве [7]. Пакет программ в среде MatLab дается в папке **COORDINATES** и в прилагаемых листингах программ.

Цель лабораторной работы: Изучение и практическое освоение систем координат, применяемых в спутниковых радионавигационных системах

2.2 Лабораторная работа 2. 1 «Преобразование координат»

Рекомендуется следующий порядок выполнения лабораторной работы.

1. Создайте папку **COORDINATES_My** и скопируйте в нее все программы из папки **COORDINATES**.
2. Запустите MatLab.
3. Обратитесь к папке **COORDINATES_My** и откройте ее.
4. Откройте функцию **ECEFLLH_N**, внимательно изучите ее по комментариям и программным процедурам, описывающим алгоритм расчета (раздел 1. 3 книги [1]).
5. Откройте пример расчета **Pr_Coord1.m** и выполните m- файл.
6. Основываясь на m- файле **Pr_Coord1.m** выполните **задание 1**.
7. Откройте функцию **LLHECEF_N**, внимательно изучите ее по комментариям и программным процедурам, описывающим алгоритм расчета (раздел 1. 3 книги [1]).
8. Откройте пример расчета **Pr_Coord2.m** и выполните m- файл.
9. Основываясь на m- файле **Pr_Coord2.m** выполните **задание 2**.
10. Откройте функцию **top_coord**, внимательно изучите ее по комментариям и программным процедурам, описывающим алгоритм расчета (раздел 1. 3 книги [1]).
11. Откройте пример расчета **prim_top_coord.m** и выполните m- файл.
12. Основываясь на m- файле **prim_top_coord.m** выполните **задание 3**.
13. **Задание 1.** По географической карте определите широту (градусы/минуты/секунды) и долготу (градусы/минуты/секунды) любого города Европы. Преобразуйте выбранные значения в радианы. Задайте высоту в метрах (произвольно, например, 195). В соответствии с п. п. 5, 6 и выбранными входными данными сформируйте m-файл и выполните его. Результат выполнения из командного окна MatLab перенесите в отчет.

14. **Задание 2.** Используя результаты, полученные в п. 13, в качестве входных данных в соответствии с п. п. 8, 9 сформируйте m-файл и выполните. Результат выполнения из командного окна MatLab перенесите в отчет.
15. **Задание 3.** Задайте координаты двух объектов, находящихся в прямой видимости широта (градусы, минуты, секунды), долгота (градусы, минуты, секунды), высота (метры). Преобразуйте заданные координаты широты и долготы в градусы. Используя эти исходные данные и п. п. 11, 12 сформируйте m-файл и выполните его. Результат выполнения из командного окна MatLab перенесите в отчет.
16. Обратитесь к папке **.ECI_ECEF_LLH** и откройте ее.
17. Откройте функцию `eci_to_ecef`, внимательно изучите ее по комментариям и программным процедурам, описывающим алгоритм расчета (раздел 1. 3 книги [1]).
18. Откройте пример расчета `Pr_eci_ecef.m` и выполните m- файл.
19. Основываясь на m- файле `Pr_eci_ecef.m` выполните **задание 4**.
20. Откройте функцию `llh_to_eci`, внимательно изучите ее по комментариям и программным процедурам, описывающим алгоритм расчета (раздел 1. 3 книги [1]).
21. Откройте пример расчета `Pr_ecef_eci.m` и выполните m- файл.
22. Основываясь на m- файле `Pr_ecef_eci.m` выполните **задание 5**.
23. **Задание 4.** Задайте координаты и скорости объекта `satpos_eci`, истинное звездное время $s0$, текущее время ti . Используя эти исходные данные и п. п. 17, 18 сформируйте m-файл и выполните его. Результат выполнения из командного окна MatLab перенесите в отчет.
24. **Задание 5.** Выберите эллипсоид и задайте его полуоси a , b ; задайте текущее время ti , истинное звездное время $time_s0$, координаты потребителя `llh_loc`. Используя эти исходные данные и п. п. 20, 21 сформируйте m-файл и выполните его. Результат выполнения из командного окна MatLab перенесите в отчет.
25. Обратитесь к папке **.TEST** и откройте ее.
26. Откройте функции `ECEF_to_LLH_Dg_Zu`, `ECEF_to_LLH_Itera`, `ECEF_to_LLH_Kelly`, внимательно изучите их по комментариям и программным процедурам, описывающим алгоритм расчета (раздел 1. 3 книги [], руководство []).
27. Откройте пример тестирования расчетов `Test_Coord.m` , изучите его, выполните m-файл.
28. Основываясь на m- файле `Test_Coord.m` выполните **задание 6**.
29. **Задание 6.** Используя входные данные из заданий 1- 5 и функции `ECEF_to_LLH_Dg_Zu`, `ECEF_to_LLH_Itera`, `ECEF_to_LLH_Kelly` выполните сопоставление расчетов по точной , итерационной и приближенной формулам при изменении

высоты в пределах 0 – 500 м; 1 – 10 км; 1 – 20 000 км. Расчеты проведите в области экватора, северного полюса и Киева. Результаты выполнения из графиков включите в отчет.

2.3 Вопросы и задания для самоподготовки

1. Какие системы координат применяются в спутниковых радионавигационных системах?
2. Какая разница между геоцентрическими и геодезическими координатами?
3. Что обозначают понятия правая и левая системы координат?
4. Что обозначают понятия подвижная и неподвижная системы координат?
5. Запишите в аналитическом виде формулы перехода из пространственной эллипсоидной географической системы в геоцентрическую фиксированную систему (ECEF).
6. Дайте определение пространственной эллипсоидной географической системе координат (центр, широта, долгота, высота).
7. Как определяются эллипсоид, геоид?
8. Сформулируйте определение системы координат, имеющей международное обозначение ECEF.
9. Сформулируйте определение системе координат, имеющей международное обозначение ECI.
10. Дайте определение топоцентрической системе координат (центр, направления осей).
11. Дайте определение системе координат WGS 84 (центр, направления осей, параметры эллипсоида, в каких спутниковых радионавигационных системах является опорной).
12. Дайте определение системе координат ПЗ 90 (центр, направления осей, параметры эллипсоида, в каких спутниковых радионавигационных системах является опорной).
13. Объясните понятие «прямая видимость».
14. Запишите формулу перевода градусов, минут, секунд в градусы, радианы; составьте программу в виде m- файла и убедитесь в правильности работы программы.
15. Запишите формулу перевода радиан в градусы, минуты, секунды; в градусы. Составьте программу в виде m- файла и убедитесь в правильности работы программы.

2.4 Тексты программ

2.4.1 Функции и файлы из папки COORDINATES

Функция ECEFLLH_N

function [XYZ] = ECEFLLH_N(llh,ab)

%Имя функции: ECEFLLH_N

%Назначение функции: преобразование координат из географической системы в прямоугольную

%Входные данные:

%llh.lon-долгота;

%llh.lat-широта;

%llh.h-высота;

%ab.a-большая полуось эллипсоида;

%ab.b- малая полуось эллипсоида в WGS-84;

%Выходные данные:

%XYZ.x,XYZ.y,XYZ.z- координаты X, Y, Z соответственно в ECEF

% Справочные данные:

%ECEF- прямоугольная геоцентрическая система координат

%a=6378137.0 (м)- большая полуось эллипсоида для WGS-84;

%b=6356752.314 (м)- малая полуось эллипсоида для WGS-84;

%A_PZ90_M = 6 378 136 (м)- большая полуось эллипсоида для ПЗ 90;

%B_PZ90_M = 6356751.36174 (м)- малая полуось эллипсоида для ПЗ 90;

a2=ab.a*ab.a;

b2=ab.b*ab.b;

r=a2/sqrt(a2*cos(llh.lat)*cos(llh.lat)+b2*sin(llh.lat)*sin(llh.lat));

XYZ.x=(r+llh.h)*cos(llh.lat)*cos(llh.lon);

XYZ.y=(r+llh.h)*cos(llh.lat)*sin(llh.lon);

XYZ.z=(b2/a2*r+llh.h)*sin(llh.lat);

Файл Pr_Coord1.m

%Имя m- файла:Pr_Coord1.m

%Пример расчета

llh.lat=0.881278698506528;llh.lon=0.53169758803674;

llh.h=122.899802776054;

ab.a=6378137.0;

ab.b=6356752.314;

[XYZ] = ECEFLLH_N(llh,ab)

Функция LLHECEF_N

function [llh] = LLHECEF_N(XYZ,ab)

%Имя функции: LLHECEF_N

%Назначение функции: преобразование координат из прямоугольной системы в географическую

%Входные данные:

```

%XYZ.x,XYZ.y,XYZ.z- координаты X, Y, Z соответственно в ECEF
%ab.a-большая полуось эллипсоида;
%ab.b- малая полуось эллипсоида в WGS-84;
%Выходные данные:
%llh.lon-долгота;
%llh.lat-широта;
%llh.h-высота;
% Справочные данные:
%ECEF- прямоугольная геоцентрическая система координат
%a=6378137.0 (м)- большая полуось эллипсоида для WGS-84;
%b=6356752.314 (м)- малая полуось эллипсоида для WGS-84;
%A_PZ90_M=6 378 136 (м)- большая полуось эллипсоида для ПЗ 90;
%B_PZ90_M= 6356751.36174 (м)- малая полуось эллипсоида для ПЗ 90;
a=6378137.0;
b=6356752.314;
a2=ab.a*ab.a;
b2=ab.b*ab.b;
xy = sqrt(XYZ.x*XYZ.x + XYZ.y*XYZ.y);
thet = atan(XYZ.z*ab.a/(xy*ab.b));
esq = 1.0-b2/a2;
epsq = a2/b2-1.0;
llh.lat = atan((XYZ.z+epsq*ab.b*(sin(thet)^3))/(xy-esq*ab.a*(cos(thet)^3)));
llh.lon = atan2(XYZ.y,XYZ.x);%!
if llh.lon < 0
llh.lon = 2*pi + llh.lon;
end ;
r = a2/sqrt(a2*cos(llh.lat)*cos(llh.lat) + b2*sin(llh.lat)*sin(llh.lat));
llh.h = xy/cos(llh.lat)-r;
end

```

Файл :Pr_Coord2.m

```

%Имя m- файла:Pr_Coord2.m
%Пример расчета
ab.a=6378137.0;
ab.b=6356752.314;
XYZ.x=3.504451023000798e+006;XYZ.y=2.061316876000462e+006;
XYZ.z=4.897990974997338e+006;
[llh] = LLHECEF_N(XYZ,ab)

```

Функция top_coord

```
function [top] = top_coord(rec_llh, rec_xyz, nlo_xyz)
```

```

% Имя функции: top_coord
%Назначение функции: расчет топоцентрических координат объекта по заданным
%географическим (долгота, широта, высота) и геоцентрическим (x, y, z)
%координатам приемника и геоцентрическим координатам объекта (x, y, z)
% Входные данные:
% rec_llh.lat - широта (рад) приемника;
%rec_llh.lon -- долгота (рад) приемника;
%rec_llh.h- высота (м) приемника;
%прямоугольные геоцентрические координаты приемника (м):
% rec_xyz.x
% rec_xyz.y
%,, rec_xyz.z
%прямоугольные геоцентрические координаты объекта (м):
% ns.x - координата x;
% ns.y -координата y;
% ns.z- координата z ;
% Выходные данные:
% top.s - проекция вектора дальности на ось (м) , направленную на Юг (South)
% top.e - проекция вектора дальности на ось (м) , направленную на Восток (East)
% top.z - проекция вектора дальности на ось (м) , направленную в Зенит
% top.daln - дальность до объекта (м)
% top.az - угол азимута объекта (градус)
% top.el - угол видимости объекта (градус)

rx = nlo_xyz.x - rec_xyz.x;
ry = nlo_xyz.y - rec_xyz.y;
rz = nlo_xyz.z - rec_xyz.z;
r_sat = sqrt(rx*rx + ry*ry + rz*rz);
r_rec = sqrt((rec_xyz.x)^2 + (rec_xyz.y)^2+ (rec_xyz.z)^2);
top.r = r_sat;
rx1 = rx; ry1 = ry; rz1 = rz;
sin_lat = sin(rec_llh.lat);
cos_lat = cos(rec_llh.lat);
sin_lon = sin(rec_llh.lon);
cos_lon = cos(rec_llh.lon);
% Projections of vector of range in topocentric coordinate system:
top.e = -sin_lon * rx1 + cos_lon * ry1;
top.s = cos_lon * sin_lat * rx1 + sin_lon * sin_lat * ry1 - cos_lat * rz1;
top.z = cos_lat * cos_lon * rx1 + cos_lat * sin_lon * ry1 + sin_lat * rz1;
% azimuth: отсчет по часовой стрелке от оси направленной на Север (N or -S) (-top.s)
eps = 10e-10;
if ( (abs(top.e) < eps) || (abs(top.s) < eps))

```

```

    top.az = 0.0;
else
    top.az = atan2(top.e,-top.s);
end;
if (top.az < 0.0)
    top.az = top.az + pi * 2;
end;
% elevation:
cos_el_top = (rec_xyz.x * rx + rec_xyz.y * ry + rec_xyz.z * rz) / (r_sat * r_rec);
if ( cos_el_top >= 1.00 )
    el = 0.0;
else
    if ( cos_el_top <= -1.00 )
        el = pi;
    else
        el = acos(cos_el_top);
    end;
end;
top.el = pi / 2.0 - el;

```

Файл prim_top_coord.m

```

%Имя m-файла: prim_top_coord.m
%Пример расчета
a=6378137.0; b=6356752.314; % для WGS-84;
% Коэффициенты перевода градусов в радианы и обратно
A2R = pi/180;
R2A = 180/pi;
%Входные данные координаты, например, приемника
rec_deg.lon = 100;
rec_deg.lat = 40;
rec_deg.h = 0;
%Входные данные координат объекта
nlo_deg.lon = 280;
nlo_deg.lat = -40;
nlo_deg.h = 0;
%Преобразование градусов в радианы
rec_llh.lon = rec_deg.lon * A2R;
rec_llh.lat = rec_deg.lat * A2R;
rec_llh.h = rec_deg.h;
nlo_llh.lon = nlo_deg.lon * A2R;
nlo_llh.lat = nlo_deg.lat * A2R;
nlo_llh.h = nlo_deg.h;

```

```

%Преобразование координат приемника и объекта систему ECEF
[rec_xyz] = ECEFLLH(a, b, rec_llh);
[nlo_xyz] = ECEFLLH(a, b, nlo_llh);
%Преобразование координат приемника и объекта в топоцентрическую
%систему координат
[top] = top_coord(rec_llh, rec_xyz, nlo_xyz);
%Вывод данных приемника в топоцентрической системе координат
fprintf('e=%22.16e s=%22.16f z=%22.16f az=%f el=%f r=%f\n', top.e, top.s, top.z, top.az*R2A,
top.el*R2A, top.r);
%Вывод данных объекта в топоцентрической системе координат
fprintf('e=%22.16e s=%22.16f z=%22.16f az=%f el=%f r=%f\n', top.e, top.s, top.z, top.az*R2A,
top.el*R2A, top.r);

```

Функция ECEFLLH

```

function [R] = ECEFLLH(a, b, llh)
%Имя функции: ECEFLLH
%Назначение- вариант функции ECEFLLH_N
a2 = a * a;
b2 = b * b;
n = a2 / sqrt(a2 * cos(llh.lat)*cos(llh.lat) + b2 * sin(llh.lat) * sin(llh.lat));
R.x = (n + llh.h) * cos(llh.lat) * cos(llh.lon);
R.y = (n + llh.h) * cos(llh.lat) * sin(llh.lon);
R.z = (b2 / a2 * n + llh.h) * sin(llh.lat);

```

2.4.2 Функции и файлы из папки **ECI_ECEF_LLH**

Функция eci_to_ecef

```

function [satpos_ecef] =eci_to_ecef(s0, ti, satpos_eci)
%Имя функции:eci_to_ecef
%Функция преобразования координат
%Входные данные: s0 - истинное звездное время в текущий момент обсервации ,
%ti - текущее время; satpos_eci
%Структура satpos_eci
%satpos_eci.x - координата x в абсолютной неподвижной системе координат (ECI);
%satpos_eci.y - координата y в абсолютной неподвижной системе координат (ECI);
%satpos_eci.z - координата z в абсолютной неподвижной системе координат (ECI);
%satpos_eci.vx - скорость vx в абсолютной неподвижной системе координат (ECI);
%satpos_eci.vy - скорость vy в абсолютной неподвижной системе координат (ECI);
% satpos_eci.vz - скорость vz в абсолютной неподвижной системе координат (ECI);
%Выходные данные:
% Структура satpos_ecef
%satpos_ecef.x - координата x в подвижной системе координат (ECEF);

```

```

%satpos_ecef.y - координата y в подвижной системе координат (ECEF);
%satpos_ecef.z - координата z в подвижной системе координат (ECEF);
%satpos_ecef.vx - скорость по оси x в подвижной системе координат (ECEF);
%satpos_ecef.vy - скорость по оси y в подвижной системе координат (ECEF);
%satpos_ecef.vz- скорость по оси z в подвижной системе координат (ECEF);
%Коэффициенты
% SEC_IN_RAD - коэффициент преобразования секунд в радианы
% s0(radian) = s0 (sek) * SEC_IN_RAD, where
% SEC_IN_RAD = 2 * pi / (24 * 3600) = pi / 43200
SEC_IN_RAD = pi / 43200;
OMEGA_Z = 0.7292115e-4; %( скорость вращения Земли (angular speed of rotation of the Earth,
рад/сек)
s_zv = s0 * SEC_IN_RAD + OMEGA_Z * ti;
cos_s = cos(s_zv);
sin_s = sin(s_zv);

satpos_ecef.x = satpos_ecef.y * cos_s + satpos_ecef.z * sin_s;
satpos_ecef.y = -satpos_ecef.x * sin_s + satpos_ecef.z * cos_s;
satpos_ecef.z = satpos_ecef.z;

satpos_ecef.vx = satpos_ecef.vx * cos_s + satpos_ecef.vy * sin_s + OMEGA_Z * satpos_ecef.y;
satpos_ecef.vy = -satpos_ecef.vx * sin_s + satpos_ecef.vy * cos_s - OMEGA_Z * satpos_ecef.x;
satpos_ecef.vz = satpos_ecef.vz;

```

Файл Pr_ecef.m

```

%Имя файла:Pr_ecef.m
%Назначение- пример преобразования координат
%Входные данные:
s0 = 400;
ti =500;
satpos_ecef.x= 20000000;% координата x в абсолютной неподвижной системе координат (ECI);
satpos_ecef.y= 15000000 ;%координата y в абсолютной неподвижной системе координат (ECI);
satpos_ecef.z = 10000000;% координата z в абсолютной неподвижной системе координат (ECI);
satpos_ecef.vx = 5000;% скорость vx в абсолютной неподвижной системе координат (ECI);
satpos_ecef.vy= 6000;% скорость vy в абсолютной неподвижной системе координат (ECI);
satpos_ecef.vz= 7000;%скорость vz в абсолютной неподвижной системе координат (ECI);
%Выходные данные
[satpos_ecef] =eci_to_ecef(s0, ti, satpos_ecef)

```

Функция llh_to_eci

```

function [eci_llh, eci_xyz] = llh_to_eci(a, b, ti, time_s0, llh_loc) ;
%Имя функции:llh_to_eci

```

```

%Функция вычисляет позицию приемника в абсолютной геоцентрической системе координат (ECI)
%Входные данные:
%a, b-большая и малая полуоси земного эллипсоида (метр);
%ti- текущее время (секунды) ,
%time_s0- истинное звездное время,
%Структура llh_loc - координаты приемника; {
%llh_loc.lon-долгота (радиан);
%llh_loc.lat-широта (радиан);
%llh_loc.h-высота (метр);
%Выходные данные:
%Структура eci_llh - географические координаты приемника в абсолютной геоцентрической системе
координат (ECI)
%eci_llh.lon - долгота (радиан);
%eci_llh.lat - широта (радиан);
%eci_llh.h - высота (метр);
%Структура eci_xyz- координаты приемника в абсолютной прямоугольной геоцентрической системе
координат (ECI)
% eci_xyz.x - координата x;
% eci_xyz.y - координата y;
% eci_xyz.z - координата z;
OMEGA_Z = 0.7292115e-4; % угловая скорость вращения Земли, (рад/сек)
SEC_IN_RAD = 7.2722052166430e-005; % (PI / 43200.0 ) // Number radian
eci_llh.lon = llh_loc.lon + ti * OMEGA_Z + SEC_IN_RAD * time_s0;
eci_llh.lat = llh_loc.lat;
eci_llh.h = llh_loc.h;
eci_xyz = llh_to_ecef( a, b, eci_llh);

```

Файл :Pr_ecef_eci.m

```

%Имя файла:Pr_ecef_eci.m
%Назначение- пример преобразования координат
%Входные данные:

a=6378137.0;% (м)- большая полуось эллипсоида для WGS-84;
b=6356752.314;% (м)- малая полуось эллипсоида для WGS-84;
ti= 700;% текущее время (секунды) ,
time_s0= 100;% истинное звездное время,
%координаты приемника:
llh_loc.lon = 30*pi/180; %долгота (радиан);
llh_loc.lat= 55*pi/180;%широта (радиан);
llh_loc.h= 184; %высота (метр);
%Выходные данные

```

```
[eci_llh, eci_xyz] = llh_to_eci(a, b, ti, time_s0, llh_loc)
```

Функция llh_to_ecef

```
function [XYZ] = llh_to_ecef( a, b, llh)
```

```
%Имя функции:LLH_to_ECEF.m
```

```
%Функция преобразования географических координат в прямоугольную геоцентрическую систему координат (ECEF)
```

```
%Входные данные:
```

```
%Структура llh
```

```
%llh.lon-долгота (радиан),
```

```
%llh.lat-широта (радиан),
```

```
%llh.h-высота (метр);
```

```
%a, b-большая и малая полуоси земного эллипсоида в WGS-84 (метр);
```

```
%Выходные данные:
```

```
%Структура XYZ
```

```
%XYZ.x - координата x в ECEF;
```

```
%XYZ.y - координата y в ECEF;
```

```
%XYZ.z - координата z в ECEF;
```

```
a2=a * a;
```

```
b2=b * b;
```

```
r = a2 / sqrt(a2 * cos(llh.lat) * cos(llh.lat) + b2 * sin(llh.lat) * sin(llh.lat));
```

```
XYZ.x = (r + llh.h) * cos(llh.lat) * cos(llh.lon);
```

```
XYZ.y = (r + llh.h) * cos(llh.lat) * sin(llh.lon);
```

```
XYZ.z = (b2 / a2 * r + llh.h) * sin(llh.lat);
```

2.4.3 Функции и файлы из папки **TEST**

Функция ECEF_to_LLH_Dg_Zu

```
function [llh_D] = ECEF_to_LLH_Dg_Zu(a, b, XYZ)
```

```
%Имя функции: ECEF_to_LLH_Dg_Zu
```

```
%Назначение функции преобразование координат из прямоугольной системы в географическую  
% по точным формулам
```

```
%Входные данные:
```

```
%XYZ.x- координата x, (м) в системе ECEF;
```

```
%XYZ.y- координата y, (м) в системе ECEF;
```

```
%XYZ.z- координата z, (м) в системе ECEF;
```

```
%a=6378137.0 (м)- большая полуось эллипсоида для WGS-84;
```

```
%b=6356752.314 (м)- малая полуось эллипсоида для WGS-84;
```

```
%A_PZ90_M=6 378 136 (м)- большая полуось эллипсоида для ПЗ 90;
```

```
%B_PZ90_M= 6356751.36174 (м)- малая полуось эллипсоида для ПЗ 90;
```

```
%Выходные данные:
```



```

%llh_D.lambda-долгота;
%llh_D.Fi-широта;
%llh_D.h-высота;
e2=0.00669437999; %квadrat эксцентриситета эллипсоида
a=6378137;% экваториальный радиус эллипсоида
b=a*sqrt(1-e2);% полярный радиус эллипсоида
%{
XYZ.x= 1.449310528799138e+007;
XYZ.y= 8.513116350113131e+006;
XYZ.z= 2.031312580583197e+007;
%}
w=sqrt(XYZ.x*XYZ.x+XYZ.y*XYZ.y);
l=e2/2;
m=(w/a)*(w/a);
n= ((1-e2)*XYZ.z/b)^2;
i=-(2*1+m+n)/2;
k= 1*(1-m-n);
q=(m+n-4*1)^3/216 +m*n*1;
D=sqrt((2*q-m*1)*m*n*1);
bet=i/3-(q+D)^(1/3) - (q-D)^(1/3);
t=sqrt(sqrt(bet*bet-k)-(bet+i)/2)-sign(m-n)*sqrt((bet-i)/2);
w1=w/(t+1);
z1=(1-e2)*XYZ.z/(t-1);
llh_D.Fi= atan(z1/((1-e2)*w1));
llh_D.lambda= 2*atan((w-XYZ.x)/XYZ.y);
llh_D.h= sign(t-1)*sqrt((w-w1)^2 +(XYZ.z-z1)^2);

```

Функция ECEF_to_LLH_Itera

```

function [llh] = ECEF_to_LLH_Itera(a, b, XYZ)
%Имя функции: ECEF_to_LLH_Itera
%Назначение функции преобразование координат из прямоугольной системы в географическую
% по итерационным формулам
%Входные данные:
%XYZ.x- координата x, (м) в системе ECEF;
%XYZ.y- координата y, (м) в системе ECEF;
%XYZ.z- координата z, (м) в системе ECEF;
%a=6378137.0 (м)- большая полуось эллипсоида для WGS-84;
%b=6356752.314 (м)- малая полуось эллипсоида для WGS-84;
%A_PZ90_M =6 378 136 (м)- большая полуось эллипсоида для ПЗ 90;
%B_PZ90_M = 6356751.36174 (м)- малая полуось эллипсоида для ПЗ 90;
%Выходные данные:
%llh.lon - долгота;

```

```

% llh.lat - широта;
%llh.h- высота;
a2 = a * a;
b2 = b * b;
sqrt_xy = sqrt(XYZ.x * XYZ.x + XYZ.y*XYZ.y);
e2 = 1.0 - b2 / a2; % ' e =Эксцентриситет, e2 = e^2
eps = 0.001;
delta_h = 100;
n = 0;
v = a;% радиус кривизны в главном вертикале
h = 0;
while (abs(delta_h) > eps)
    n = n + 1;
    fi = atan(XYZ.z / (sqrt_xy * (1 - e2 * v / (v + h)))); % lat
    llh.lat = fi;
    sin_fi = sin(fi);
    cos_fi = cos(fi);
    v = a / sqrt(1 - e2 * sin_fi * sin_fi);
    llh.lon = atan2(XYZ.y,XYZ.x);
    llh.h = sqrt_xy / cos_fi - v;
    delta_h = llh.h - h;
    % fprintf('n=%i h=%f lat=%f lon=%f h=%f delta_h= %f\n',n, h, llh.lat, llh.lon, llh.h, delta_h);
    h = llh.h;
end;
end

```

Функция ECEF_to_LLH_Kelly

```

function [llh] = ECEF_to_LLH_Kelly(a, b, XYZ)
%Имя функции: ECEF_to_LLH_Kelly
%Назначение функции преобразование координат из прямоугольной системы в географическую
% по итерационным формулам
%Входные данные:
%XYZ.x- координата x, (м) в системе ECEF;
%XYZ.y- координата y, (м) в системе ECEF;
%XYZ.z- координата z, (м) в системе ECEF;
%a=6378137.0 (м)- большая полуось эллипсоида для WGS-84;
%b=6356752.314 (м)- малая полуось эллипсоида для WGS-84;
%A_PZ90_M=6 378 136 (м)- большая полуось эллипсоида для ПЗ 90;
%B_PZ90_M= 6356751.36174 (м)- малая полуось эллипсоида для ПЗ 90;
%Выходные данные:
%llh.lon - долгота;
% llh.lat - широта;

```

```

%llh.h- высота;
a2 = a * a;
b2 = b * b;
xy = sqrt(XYZ.x * XYZ.x + XYZ.y*XYZ.y);
thet = atan(XYZ.z * a / (xy * b));
esq = 1.0 - b2/a2; % 'e = Эксцентриситет, esq = e^2
epsq = a2/b2 - 1.0;
llh.lat = atan((XYZ.z + epsq * b * (sin(thet)^3)) / (xy - esq * a * (cos(thet)^3)));
llh.lon = atan2(XYZ.y,XYZ.x);
if llh.lon < 0
    llh.lon = 2*pi + llh.lon;
end ;
r = a2 / sqrt(a2 * cos(llh.lat) * cos(llh.lat) + b2 * sin(llh.lat) * sin(llh.lat));
llh.h = xy/cos(llh.lat) - r;
end

```

Функция ECEFLH

```

function [R] = ECEFLH(a, b, llh)
%Имя функции: ECEFLH
%Назначение- вариант функции ECEFLH_N
a2 = a * a;
b2 = b * b;
n = a2 / sqrt(a2 * cos(llh.lat)*cos(llh.lat) + b2 * sin(llh.lat) * sin(llh.lat));
R.x = (n + llh.h) * cos(llh.lat) * cos(llh.lon);
R.y = (n + llh.h) * cos(llh.lat) * sin(llh.lon);
R.z = (b2 / a2 * n + llh.h) * sin(llh.lat);

```

Файл Test_Coord.m

```

%Имя m-файла:Test_Coord.m
%Назначение: пример тестирования программ преобразования координат
%llh.lat = 0.88032730015257; %50 град; 26 мин.; 20.54 с
%ввод входных данных
llh.lat =55*pi/180;
llh.lon = 0.53109641675259;%30 град; 25 мин.; 46.4995 с
a=6378137.0;
b=6356752.314;
% ввод высоты и шага изменения высоты
llh0.h=0;%184;%высота в метрах
step_h= 1000;
kol=40000;
%nn=0;

```

```

for nn=1:kol
%   nn=nn+1;
    llh.h= llh0.h+step_h*(nn-1);
%применение функции вычисления прямоугольных координат при переменной высоте
[R] = ECEFLLH(a, b, llh);

%преобразование прямоугольных координат в географические по точной формуле
[llh_D] = ECEF_to_LLH_Dg_Zu(a, b, R);
%преобразование прямоугольных координат в географические поприближенной формуле
%указание снять блокировку строки и переписать вывод в графике
% [llh] = ECEF_to_LLH_Kelly(a, b,R);
%[llh_K] =Kelly(a, b, R);
%преобразование прямоугольных координат в географические по итерационной формуле
[llh1] = ECEF_to_LLH_Itera(a, b, R);
%{
delta.lat(nn)=llh_D.Fi- llh.lat;
delta.lon(nn)=llh_D.lambda- llh.lon;
delta.h(nn)= llh_D.h- llh.h;
%}
h1(nn)= llh.h; %llh0.h+step_h*(nn-1);
delta.lat(nn) = llh1.lat - llh.lat;
%delta.lon(nn)=llh_D.lambda- llh_K.lon;
%delta.h(nn) = llh.h - llh_K.h;
delta.h(nn) = llh.h - llh1.h;
end
%Графика
subplot (1,2,1), plot(h1(2:kol)/1000,delta.h(2:kol)*1000), grid on
xlabel('a','FontSize',14,'FontName','TimesNewRoman')
set(get(gcf,'CurrentAxes'),'FontSize',14,'FontName','TimesNewRoman');
subplot (1,2,2),plot(h1(2:kol)/1000,delta.lat(2:kol)*180*3600/pi), grid on
xlabel('б','FontSize',14,'FontName','TimesNewRoman')
set(get(gcf,'CurrentAxes'),'FontSize',14,'FontName','TimesNewRoman');
%plot(h1,delta.lat*180/pi)
%plot(h1*10^(-3),delta.h)
%plot(1:kol+1, h1)

```

РАЗДЕЛ 3 Время

3.1 Краткие сведения из теории

В спутниковой радионавигации время имеет большое значение, поскольку основные навигационные определения производятся по формулам, в которых параметр времени присутствует многократно. Прежде всего, это время распространения электромагнитного сигнала от навигационного спутника до потребителя, время «включения» часов спутника, время синхронизации данных передаваемых со спутника, время прохождения электромагнитного сигнала через атмосферу, влияние на время релятивистских эффектов, совмещение шкал времени спутника и потребителя, системное время СРНС, опорные моменты отсчета времени (эпохи), единицы счета времени (год, неделя, день, час, минута, секунда, миллисекунда) и многое другое.

Алгоритмы расчета времени, запрограммированные в прилагаемом пакете программ, изложены в книге [1] (раздел 1. 3, стр.40 -50), а также в источниках, приведенных в библиографии к книге [1]. Пакет программ в среде MatLab дается в папке **TIME** и в прилагаемых листингах программ.

Цель лабораторной работы: Изучение основных временных составляющих, применяемых в алгоритмах и программах спутниковой аппаратуры потребителя для решений навигационных задач.

3.2 Лабораторная работа 3. 1 «Время в спутниковых радионавигационных системах»

Рекомендуется следующий порядок действий для выполнения лабораторной работы.

1. Создайте папку **TIME_Му** и скопируйте в ее все программы из папки **TIME**.
2. Запустите MatLab [7, 8].
3. Обратитесь к папке **TIME_Му** и откройте ее.
4. Откройте функцию **JD_epoch**, внимательно изучите ее по комментариям и программным процедурам, описывающим алгоритм расчета (раздел 1. 3 книги [1]).
5. Откройте пример расчета **PR_JD_epoch.m** и выполните m- файл.
6. Основываясь на m- файле **PR_JD_epoch.m** выполните **задание 1**.
7. **Задание 1.** Используя в качестве основы m- файл **PR_JD_epoch.m** , сформируйте m- файл и рассчитайте юлианского день для опорного года, в котором Вы родились. Рассчитайте юлианский день эпохи 2000 (2000 год) и объясните причину разницы в 1 день. Результат выполнения из командного окна MatLab перенесите в отчет.
8. Откройте функцию **JD_data**, внимательно изучите ее по комментариям и программным процедурам, описывающим алгоритм расчета (раздел 1. 3 книги [1]).

9. Откройте пример расчета PR_JD_data.m и выполните m- файл.
10. Основываясь на m- файле PR_JD_data.m выполните **задание 2**.
11. **Задание 2.** Используя в качестве основы m- файл PR_JD_data.m , сформируйте m- файл и рассчитайте юлианского день для опорного года и номер дня года, в котором Вы родились. Рассчитайте юлианский день эпохи 2000 (2000 год) и объясните причину разницы в 1 день. Результат выполнения из командного окна MatLab перенесите в отчет.
12. Откройте функцию time_gps, внимательно изучите ее по комментариям и программным процедурам, описывающим алгоритм расчета (раздел 1. 3 книги [1]).
13. Откройте пример расчета PR_time_gps.m и выполните m- файл.
14. Основываясь на m- файле PR_time_gps.m выполните **задания 3 и 4**.
15. **Задание 3.** Используя в качестве основы m- файл PR_JD_data.m , сформируйте m- файл и рассчитайте номер GPS-недели, время UTC с начала текущей недели, время GPS с начала текущей недели, номер дня года, номера юлианского дня на момент выполнения лабораторной работы. Результат выполнения из командного окна MatLab перенесите в отчет.
16. **Задание 4.** В сформированном файле задайте номера дней, соответствующих субботе и воскресению и убедитесь, что происходит изменение номера недели. Результат выполнения из командного окна MatLab перенесите в отчет.
17. Откройте функцию week_GLONAS_gps, внимательно изучите ее по комментариям и программным процедурам.
18. Откройте пример расчета PR_time_gps_GLON.m и выполните m- файл.
19. Основываясь на m- файле PR_time_gps_GLON.m выполните **задание 5**.
20. **Задание 5.** Используя в качестве основы m- файл PR_time_gps_GLON.m , сформируйте m- файл, задайте Na- календарный номер суток внутри четырехлетнего периода от ближайшего високосного года, выполните m- файл . Результат выполнения из командного окна MatLab перенесите в отчет.
21. Создайте папку **TIME_S0_My** и скопируйте в нее все программы из папки **TIME_S0**.
22. Обратитесь к папке **TIME_S0_My** и откройте ее.
23. Последовательно открывая функции: s0_Nut, utc_nut, koef, utc_nut_fi_eps внимательно изучите их по комментариям и программным процедурам, описывающим алгоритм расчета (раздел 1. 3 книги []).
24. Выполните **указание 1** и дополните папку **TIME_S0_My** функциями *JD_epochi*, *JD_data*.
25. Откройте пример расчета PR_s0_Nut.m и выполните m- файл.

26. Основываясь на m- файле PR_s0_Nut.m выполните **задание 6**.
27. **Задание 6.** Используя в качестве прототипа m- файл PR_s0_Nut.m сформируйте m- файл, введите в него исходные данные, соответствующие времени выполнения работы выполните m- файл (рассчитайте истинное и среднее звездное время) и занесите результаты выполнения из командного окна MatLab в отчет.

3.3 Вопросы и задания для самоподготовки

1. Что понимается под терминами звездное время, истинное звездное время, среднее звездное время, время на гринвичском меридиане ?
2. В каких спутниковых радионавигационных системах и для чего применяется юлианский день?
3. Назовите основные фундаментальные эпохи, используемые в спутниковой радионавигации.
4. Какие единицы измерения времени применяются в GPS?
5. Какие единицы измерения времени применяются в ГЛОНАСС?
6. Какие единицы измерения времени применяются в EGNOS?
7. Какие единицы измерения времени применяются в GALILEO?
8. Что в GPS обозначает дата с 5. 01. 80 на 6. 01. 80?
9. На сколько секунд системное время GPS опережает время UTC?
10. Какой смысл в ГЛОНАСС вкладывается в определение «московское декретное время»?
11. Что такое универсальное всемирное время?
12. Что такое атомное время?
13. Напишите формулу, связывающую время ГЛОНАСС и время UTC.
14. Объясните физический смысл нутации.
15. Как изменяется время GPS в течение недели.
16. Как изменяется время ГЛОНАСС в течение суток.

3.4 Тексты программ

3.4.1 Функции и файлы из папки TIME

Функция JD_epohi

```
function jden = JD_epohi(epoha)
%Имя: JD_epohi
%Функция JD_epohi(epoha) рассчитывает юлианский день
%для опорного года (epoha)
```

```

%Входные данные: epoха, размерность-год
%Выходные данные:jden- юлианский день, размерность -дни
rk = mod(epoха,4);
if ( rk == 0 ) rk = 1.0;
else
    rk = 2.0 - rk * 0.25;
end;
n100 = floor(epoха / 100);
n400 = floor(epoха / 400);
jden = (4712 + epoха) * 365.25 + n400 - n100 + rk;
% fprintf('epoха=%d rk=%f jden=%6.2f \n', epoха, rk, jden);

```

Файл PR_JD_epoхи.m

```

%Пример расчета JD_epoхи
epoха=2000;
jden = JD_epoхи(epoха)

```

Решение по данным примера. Юлианский день 2000 года (эпоха 2000) равен 2451544 (**нулевой день !!!**)

Функция JD_data (расчет номера юлианского дня).

```

function [JD, day_year] = JD_data(year, mon, day)
%Имя:JD_data
% Функция JD_data(year, mon, day) вычисляет :
%JD - номер юлианского дня, day_year - номер дня года.
%Входные данные:
%year - год,
% mon - месяц,
% day - день.
%Выходные данные:
%JD - юлианский день, day_year- день от начала года.
%Оригинальные функции: function jd0 = JD_epoхи(year),
%(смотри комментарий).
% число дней от начала периода до 12ч. всемирного времени (полдень)
% указанной даты ( по Гринвичу)
%Входные данные для контрольного примера:
%year = 2000; mon = 1; day = 1;
%количество дней в месяцах
DnMon = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31];
%Вычисление номера юлианского дня опорной эпохи
jd0 = JD_epoхи(year);
%Учет високосного года
nfebr = 0;
if mod(year,4) == 0

```



```

    nfebr = 1;
end;
%Расчет номера дня года
    k = 0;
    for i = 2 : mon
        k = k + DnMon(i - 1);
        if (i == 2)
            k = k + nfebr;
        end;
    end;
    day_year = k + day;
%Расчет номера юлианского дня
    JD = jd0 + day_year;
%2451545 - номер юлианского дня; 1- первый день года.

```

Файл PR_JD_data.m

```

%Пример расчета JD_data
year = 2000; mon = 1; day = 1;
[JD, day_year] = JD_data(year, mon, day);
[JD, day_year]'
%Результат расчета JD= 2451545, day_year= 1

```

Функция *time_gps* (расчет номера GPS-недели, времени UTC с начала текущей недели, времени GPS с начала текущей недели, номер дня года, номера юлианского дня)

```

function [wGPS, TimesUTC, TimesGPS, day_year, JD, weekday] = time_gps(year, mon, day, h, min, sec)
% Имя: time_gps
% Функция time_gps(year, mon, day, h, min, sec) вычисляет :
%wGPS -номер GPS- недели, TimesUTC -время UTC с начала текущей недели (секунда),
% TimesGPS - время GPS с начала текущей недели (секунда),
%day_year- номер дня года,
% JD - номер юлианского дня,
% weekday - день недели (0 - воскресенье, 1 - понедельник, 2 - вторник, 3 - среда,
%4 - четверг, 5 - пятница, 6 - суббота.
% Выходные данные: wGPS, TimesUTC, TimesGPS, day_year, JD, weekday.
% Входные данные: year - год, mon - месяц, day - день, h - час,
%min - минута, sec -секунда.
%Оригинальные функции: function [JD, day_year] = JD_data(year, mon, day),
%(смотри комментарий).
% число дней от начала периода до 0ч. всемирного времени UTC
% указанной даты ( по Гринвичу)
%Входные данные для контрольного примера:
%year = 2007; mon = 11; day = 28;
%h = 20; min = 7; sec = 51;

```

```

%Используемые константы:
% сдвиг времени между UTC и системным временем GPS на начало 2004 года
%diff.UTC_GPS = 13;(до конца 2005 года)
diff.UTC_GPS = 14;%(с 2006 года)
%количество дней в месяцах
    DnMon = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31];
% юлианский день начала отсчета недели GPS на ноль часов UTC с 5 на 6
% января 1980 года.
    wGPS_0 = 2444244.5;
% Расчет номеров юлианского дня и дня года
    [JD, day_year] = JD_data(year, mon, day);
%Сдвиг на ноль часов UTC на гринвичском меридиане
    JD = JD - 0.5;
%Расчет номера GPS неделе стандартной функцией MatLab "floor"
    wGPS = floor((JD - wGPS_0) / 7);
%Расчет дня недели стандартной функцией MatLab "mod"
    weekday = mod((JD - wGPS_0) , 7);
%Расчет секунд с начала недели в шкале времени UTC
    TimesUTC = 86400 * weekday + 3600 * h + 60 * min + sec;
%Расчет секунд с начала недели в шкале времени GPS
    TimesGPS = TimesUTC + diff.UTC_GPS;
%Результат решения задачи по входным данным контрольного примера
    %1.0e+006 *- множитель для всех данных;
% 0.00145500000000 - номер GPS недели;
% 0.33167100000000 - количество секунд с начала недели по UTC;
%0.33168500000000- количество секунд с начала недели по GPS;
%0.00033200000000 - день года;
%2.45443250000000 -номер юлианского дня;
%0.00000300000000 - день недели (среда).

```

Файл PR_time_gps.m

```

%Пример 1 расчета функции time_gps(year, mon, day, h, min, sec)
year=2007;
mon= 11;
day = 28;
h = 20;
min= 7;
sec = 51;
[wGPS, TimesUTC, TimesGPS, day_year, JD, weekday] = time_gps(year, mon, day, h, min, sec) ;
[wGPS, TimesUTC, TimesGPS, day_year, JD, weekday]'

```

```

%Пример2 расчет функции time_gps(year, mon, day, h, min, sec)
year=1999;
mon= 8;
day = 28;
h = 23;
min= 59;
sec = 59;
[wGPS, TimesUTC, TimesGPS, day_year, JD, weekday] = time_gps(year, mon, day, h, min, sec) ;
[wGPS, TimesUTC, TimesGPS, day_year, JD, weekday]'

```

Результат расчета:

```

wGPS= 1024;
TimesUTC =604799;
TimesGPS = 604813;
day_year =240;
JD = 24514185;
weekday = 6 (суббота);

```

Пример 2 характерен тем, что в ночь с субботы на воскресенье 28 августа 1999 года неделя GPS равна 1024 и через одну секунду счет недель в системе GPS начался с начала.

Функция week_GLONAS_gps

```

function [wGPS,weekday] = week_GLONAS_gps(Na)
% Имя функции: week_GLONAS_gps
% Назначение: функция устанавливает связь между параметром Na, передаваемом с навигационного
%спутника ГЛОНАСС и номером GPS- недели (wGPS), днем недели (weekday)
% weekday - день недели (0 - воскресенье, 1 - понедельник, 2 - вторник, 3 - среда,
%4 - четверг, 5 - пятница, 6 - суббота.
%Входные данные:
%Na- календарный номер суток внутри четырехлетнего периода от ближайшего високосного года, к
%которым относится информация, передаваемая со спутника ГЛОНАСС (согласно интерфейсного
конт-
%рольного документа ГЛОНАСС
% Выходные данные:
%wGPS- номер недели GPS;
% weekday- день недели.
%Используемые константы:
% сдвиг времени между UTC и системным временем GPS на начало 2004 года
%diff.UTC_GPS = 13;(до конца 2003 года)
diff.UTC_GPS = 14;%(с 2004 года)
%количество дней в месяцах
DnMon = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31];

```

```

% юлианский день начала отсчета недели GPS на ноль часов UTC с 5 на 6
% января 1980 года.
wGPS_0 = 2444244.5;
% Расчет номеров юлианского дня и дня года, от которого отсчитываются дни ГЛОНАСС
year = 2003;
mon=12;
day=31;
[JD, day_year] = JD_data(year, mon, day);
%Сдвиг на ноль часов UTC на гринвичском меридиане
JD = JD - 0.5;
%Расчет номера GPS неделе стандартной функцией MatLab "floor"
wGPS = floor((JD+Na - wGPS_0) / 7);
%Расчет дня недели стандартной функцией MatLab "mod"
weekday = mod((JD+Na - wGPS_0) , 7);

```

Файл PR_time_gps_GLON.m

```

%Имя m- файла:PR_time_gps_GLON.m
%Пример2 расчет функции week_GLONAS_gps(Na)
Na=901;
[wGPS,weekday] = week_GLONAS_gps(Na)
% ответ: wGPS = 1380; weekday = 1

```

3.4.2 Функции и файлы из папки TIME_S0

Папка TIME_S0 расчет истинного и среднего звездного времени. Расчет истинного звездного времени на 0 часов UTC предусматривается алгоритмом расчета эфемерид спутника ГЛОНАСС на текущий момент времени. Истинное звездное время рассчитывается по следующим программам.

Функция s0_Nut

```

function [s0_m_mod, s0_nut_mod,s0_m_hour, s0_m_min, s0_m_sec, s0_nut_hour, s0_nut_min,
s0_nut_sec] = s0_Nut(year, mon, day, nut)
%Имя: s0_Nut
%функция рассчитывает истинное и среднее звездное время по формулам ( )
% среднее звездное время s0 на 0ч UTC
%year = 1993; mon = 1; day = 0;
%fprintf('function s0_m - start \n');
jd2000 = 2451545; % 12h UTC 1 января
% Применить функцию JD_data
[jd, day_year] = JD_data(year, mon, day);

```

```

if (jd == NaN)
    s0_mod = NaN; h = NaN; min = NaN; sec = NaN;
    fprintf('function s0_m - end0 \n');
    return;
end;
jd = jd - 0.5;
d = jd - jd2000;
t = d / 36525.0; % 36525 - юлианский период 100 лет
    t2 = t * t;
h1 = 24110.54841;
%h1=6.0*3600.0+41.0*60.0+50.54841;
% h2 = 236.555367908 * d;
    h2 = 8640184.812866 * t ;
    h3 = 0.093104 * t2;
    h4 = t2 * t * 6.2E-6;
if (nut == 0)
    na = 0;
else
    na = utc_nut(t);
end;
s0_m = h1 + h2 + h3 - h4;
s0_nut = s0_m + na;
s0_m_mod = mod(s0_m, 86400);
s0_day = floor(s0_m / 86400);
s0_m_hour = s0_m_mod / 3600.0;
s0_m_hour = floor(s0_m_mod / 3600);
sec_min = s0_m_mod - s0_m_hour * 3600;
s0_m_min = floor(sec_min / 60);
s0_m_sec = sec_min - s0_m_min * 60;
s0_nut_mod = mod(s0_nut, 86400);
s0_day = floor(s0_nut / 86400);
s0_nut_hour = s0_nut_mod / 3600.0;
s0_nut_hour = floor(s0_nut_mod / 3600);
sec_min = s0_nut_mod - s0_nut_hour * 3600;
s0_nut_min = floor(sec_min / 60);
s0_nut_sec = sec_min - s0_nut_min * 60;

```

Функция utc_nut

```

function nut = utc_nut(t)
%Имя: utc_nut
%функция рассчитывает формулы ( )
%t=6.023472005475702e+002;

```

```

R = 1296000; % ( 1r=360grad=1 296 000 cek)
RAD_SEK_ANGL = pi/(3600*180);
t2 = t * t;
t3 = t2 * t;
l = 485866.733 + (1325.0 * R + 715922.633) * t + 31.310 * t2 + 0.064 * t3;%1.034807679476340e+012
l1 = 1287099.804 + (99 * R + 1292581.224) * t - 0.577 * t2 - 0.012 * t3;
f = 335778.877 + (1342 * R + 295263.137) * t - 13.257 * t2 + 0.011 * t3;
dd = 1072261.307 + (1236 * R + 1105601.328) * t - 6.891 * t2 + 0.019 * t3;
omega = 450160.280 - (5 * R + 482890.539) * t + 7.455 * t2 + 0.008 * t3;
eps0 = 84381.448 - 46.8150 * t - 0.00059 * t2 + 0.001813 * t3;
% eps0 = 84381.447996 - 46.8150 * t - 0.00059 * t2 + 0.001813 * t3;
eps_d = utc_nut_fi_eps(t, l, l1, f, dd, omega, 'd','e');
eps_k = utc_nut_fi_eps(t, l, l1, f, dd, omega, 'k','e');
eps = eps0 + eps_d + eps_k;
cos_eps = cos(RAD_SEK_ANGL * eps) / 15.0;
d_fi = utc_nut_fi_eps(t, l, l1, f, dd, omega, 'd', 'f');
k_fi = utc_nut_fi_eps(t, l, l1, f, dd, omega, 'k', 'f');
nut1 = d_fi * cos_eps;
nut2 = k_fi * cos_eps;
% nut3 = 0.00264 * sin(omega) + 0.000063 * sin(2.0 * omega) формула ( )
nut3 = 0;
nut1_dop = nut1;
nut2_dop = nut2;
nut3_dop = nut3;
nut = nut1 + nut2 + nut3;

```

Функция koef

```

function[koef_id, koef_abd, koef_ik, koef_abk] = koef
% Имя:koef
% функция содержит коэффициенты табл. и табл.
koef_id = [ 0, 0, 0, 0, 1 ; % 1
0, 0, 0, 0, 2 ; % 2
-2, 0, 2, 0, 1 ; % 3
2, 0,-2, 0, 0 ; % 4
-2, 0, 2, 0, 2 ; % 5
1,-1, 0,-1, 0 ; % 6
0,-2, 2,-2, 1 ; % 7
2, 0,-2, 0, 1 ; % 8
0, 0, 2,-2, 2 ; % 9
0, -1, 0, 0, 0 ; % 10 % исправлено
0, 1, 2,-2, 2 ; % 11
0,-1, 2,-2, 2 ; % 12
0, 0, 2,-2, 1 ; % 13

```

-2, 0, 0,2, 0 ; % 14 % исправлено
0, 0, 2,-2, 0 ; % 15
0, 2, 0, 0, 0 ; % 16
0, 1, 0, 0, 1 ; % 17
0, 2, 2,-2, 2 ; % 18
0,-1, 0, 0, 1 ; % 19
-2, 0, 0, 2, 1 ; % 20
0,-1, 2,-2, 1 ; % 21
2, 0, 0,-2, 1 ; % 22
0, 1, 2,-2, 1 ; % 23
1, 0, 0,-1, 0 ; % 24
2, 1, 0,-2, 0 ; % 25
0, 0,-2, 2, 1 ; % 26
0, 1,-2, 2, 0 ; % 27
0, 1, 0, 0, 2 ; % 28
-1, 0, 0, 1, 1 ; % 29
0, 1, 2,-2, 0 ; % 30

koef_abd = [-171996.0,-174.2, 92025.0, 8.9; % 1
2062.0, 0.2, -895.0, 0.5; % 2
46.0, 0.0, -24.0, 0.0; % 3
11.0, 0.0, 0.0, 0.0; % 4
-3.0, 0.0, 1.0, 0.0; % 5
-3.0, 0.0, 0.0, 0.0; % 6
-2.0, 0.0, 1.0, 0.0; % 7
1.0, 0.0, 0.0, 0.0; % 8
-13187.0, -1.6, 5736.0,-3.1; % 9
-1426.0, 3.4, 54.0,-0.1; % 10 % исправлено
-517.0, 1.2, 224.0,-0.6; % 11
217.0, -0.5, -95.0, 0.3; % 12
129.0, 0.1, -70.0, 0.0; % 13
-48.0, 0.0, 1.0, 0.0; % 14 % исправлено
-22.0, 0.0, 0.0, 0.0; % 15
17.0, -0.1, 0.0, 0.0; % 16
-15.0, 0.0, 9.0, 0.0; % 17
-16.0, 0.1, 7.0, 0.0; % 18
-12.0, 0.0, 6.0, 0.0; % 19
-6.0, 0.0, 3.0, 0.0; % 20
-5.0, 0.0, 3.0, 0.0; % 21
4.0, 0.0, -2.0, 0.0; % 22
4.0, 0.0, -2.0, 0.0; % 23
-4.0, 0.0, 0.0, 0.0; % 24

```

1.0, 0.0, 0.0, 0.0; % 25
1.0, 0.0, 0.0, 0.0; % 26
-1.0, 0.0, 0.0, 0.0; % 27
1.0, 0.0, 0.0, 0.0; % 28
1.0, 0.0, 0.0, 0.0; % 29
-1.0, 0.0, 0.0, 0.0]; % 30
koef_ik = [ 0, 0, 2, 0, 2; % 31
1, 0, 0, 0, 0; % 32
0, 0, 2, 0, 1; % 33
1, 0, 2, 0, 2; % 34
1, 0, 0, -2, 0; % 35
-1, 0, 2, 0, 2; % 36
0, 0, 0, 2, 0; % 37
1, 0, 0, 0, 1; % 38
-1, 0, 0, 0, 1; % 39
-1, 0, 2, 2, 2; % 40
1, 0, 2, 0, 1; % 41
0, 0, 2, 2, 2; % 42
2, 0, 0, 0, 0; % 43
1, 0, 2, -2, 2; % 44
2, 0, 2, 0, 2; % 45
0, 0, 2, 0, 0; % 46
-1, 0, 2, 0, 1; % 47
-1, 0, 0, 2, 1; % 48
1, 0, 0, -2, 1; % 49
-1, 0, 2, 2, 1; % 50
1, 1, 0, -2, 0; % 51
0, 1, 2, 0, 2; % 52
0, -1, 2, 0, 2; % 53
1, 0, 2, 2, 2; % 54
1, 0, 0, 2, 0; % 55
2, 0, 2, -2, 2; % 56
0, 0, 0, 2, 1; % 57
0, 0, 2, 2, 1; % 58
1, 0, 2, -2, 1; % 59
0, 0, 0, -2, 1; % 60
1, -1, 0, 0, 0; % 61
2, 0, 2, 0, 1; % 62
0, 1, 0, -2, 0; % 63
1, 0, -2, 0, 0; % 64
0, 0, 0, 1, 0; % 65
1, 1, 0, 0, 0; % 66

```


1, 0, 2, 0, 0; % 67
1,-1, 2, 0, 2; % 68
-1,-1, 2, 2, 2; % 69
-2, 0, 0, 0, 1; % 70
3, 0, 2, 0, 2; % 71
0,-1, 2, 2, 2; % 72
1, 1, 2, 0, 2; % 73
-1, 0, 2,-2, 1; % 74
2, 0, 0, 0, 1; % 75
1, 0, 0, 0, 2; % 76
3, 0, 0, 0, 0; % 77
0, 0, 2, 1, 2; % 78
-1, 0, 0, 0, 2; % 79
1, 0, 0,-4, 0; % 80
-2, 0, 2, 2, 2; % 81
-1, 0, 2, 4, 2; % 82
2, 0, 0,-4, 0; % 83
1, 1, 2,-2, 2; % 84
1, 0, 2, 2, 1; % 85
-2, 0, 2, 4, 2; % 86
-1, 0, 4, 0, 2; % 87
1,-1, 0,-2, 0; % 88
2, 0, 2,-2, 1; % 89
2, 0, 2, 2, 2; % 90
1, 0, 0, 2, 1; % 91
0, 0, 4,-2, 2; % 92
3, 0, 2,-2, 2; % 93
1, 0, 2,-2, 0; % 94
0, 1, 2, 0, 1; % 95
-1,-1, 0, 2, 1; % 96
0, 0,-2, 0, 1; % 97
0, 0, 2,-1, 2; % 98
0, 1, 0, 2, 0; % 99
1, 0,-2,-2, 0; % 100
0,-1, 2, 0, 1; % 101
1, 1, 0,-2, 1; % 102
1, 0,-2, 2, 0; % 103
2, 0, 0, 2, 0; % 104
0, 0, 2, 4, 2; % 105
0, 1, 0, 1, 0]; % 106
koef_abk = [-2274.0, -0.2, 977.0, -0.5; % 31
712.0, 0.1, -7.0, 0.0; % 32

-386.0, -0.4, 200.0, 0.0; % 33
-301.0, 0.0, 129.0, -0.1; % 34
-158.0, 0.0, -1.0, 0.0; % 35
123.0, 0.0, -53.0, 0.0; % 36
63.0, 0.0, -2.0, 0.0; % 37
63.0, 0.1, -33.0, 0.0; % 38
-58.0, -0.1, 32.0, 0.0; % 39
-59.0, 0.0, 26.0, 0.0; % 40
-51.0, 0.0, 27.0, 0.0; % 41
-38.0, 0.0, 16.0, 0.0; % 42
29.0, 0.0, -1.0, 0.0; % 43
29.0, 0.0, -12.0, 0.0; % 44
-31.0, 0.0, 13.0, 0.0; % 45
26.0, 0.0, -1.0, 0.0; % 46
21.0, 0.0, -10.0, 0.0; % 47
16.0, 0.0, -8.0, 0.0; % 48
-13.0, 0.0, 7.0, 0.0; % 49
-10.0, 0.0, 5.0, 0.0; % 50
-7.0, 0.0, 0.0, 0.0; % 51
7.0, 0.0, -3.0, 0.0; % 52
-7.0, 0.0, 3.0, 0.0; % 53
-8.0, 0.0, 3.0, 0.0; % 54
6.0, 0.0, 0.0, 0.0; % 55
6.0, 0.0, -3.0, 0.0; % 56
-6.0, 0.0, 3.0, 0.0; % 57
-7.0, 0.0, 3.0, 0.0; % 58
6.0, 0.0, -3.0, 0.0; % 59
-5.0, 0.0, 3.0, 0.0; % 60
5.0, 0.0, 0.0, 0.0; % 61
-5.0, 0.0, 3.0, 0.0; % 62
-4.0, 0.0, 0.0, 0.0; % 63
4.0, 0.0, 0.0, 0.0; % 64
-4.0, 0.0, 0.0, 0.0; % 65
-3.0, 0.0, 0.0, 0.0; % 66
3.0, 0.0, 0.0, 0.0; % 67
-3.0, 0.0, 1.0, 0.0; % 68
-3.0, 0.0, 1.0, 0.0; % 69
-2.0, 0.0, 1.0, 0.0; % 70
-3.0, 0.0, 1.0, 0.0; % 71
-3.0, 0.0, 1.0, 0.0; % 72
2.0, 0.0, -1.0, 0.0; % 73
-2.0, 0.0, 1.0, 0.0; % 74

```

2.0, 0.0, -1.0, 0.0; % 75
-2.0, 0.0, 1.0, 0.0; % 76
2.0, 0.0, 0.0, 0.0; % 77
2.0, 0.0, -1.0, 0.0; % 78
1.0, 0.0, -1.0, 0.0; % 79
-1.0, 0.0, 0.0, 0.0; % 80
1.0, 0.0, -1.0, 0.0; % 81
-2.0, 0.0, 1.0, 0.0; % 82
-1.0, 0.0, 0.0, 0.0; % 83
1.0, 0.0, -1.0, 0.0; % 84
-1.0, 0.0, 1.0, 0.0; % 85
-1.0, 0.0, 1.0, 0.0; % 86
1.0, 0.0, 0.0, 0.0; % 87
1.0, 0.0, 0.0, 0.0; % 88
1.0, 0.0, -1.0, 0.0; % 89
-1.0, 0.0, 0.0, 0.0; % 90
-1.0, 0.0, 0.0, 0.0; % 91
1.0, 0.0, 0.0, 0.0; % 92
1.0, 0.0, 0.0, 0.0; % 93
-1.0, 0.0, 0.0, 0.0; % 94
1.0, 0.0, 0.0, 0.0; % 95
1.0, 0.0, 0.0, 0.0; % 96
-1.0, 0.0, 0.0, 0.0; % 97
-1.0, 0.0, 0.0, 0.0; % 98
-1.0, 0.0, 0.0, 0.0; % 99
-1.0, 0.0, 0.0, 0.0; % 100
-1.0, 0.0, 0.0, 0.0; % 101
-1.0, 0.0, 0.0, 0.0; % 102
-1.0, 0.0, 0.0, 0.0; % 103
1.0, 0.0, 0.0, 0.0; % 104
-1.0, 0.0, 0.0, 0.0; % 105
1.0, 0.0, 0.0, 0.0]; % 106

```

Функция utc_nut_fi_eps

```

function nut_fi_eps = utc_nut_fi_eps(t, l, ll, f, dd, omega, typ_nut, fi_eps)
% Имя: utc_nut_fi_eps
% функция рассчитывает нутацию
% применить функцию koef
[koef_id, koef_abd, koef_ik, koef_abk] = koef;
RAD_SEK_ANGL = pi/(3600*180);
if (typ_nut == 'd')

```

```

    n = 30;
else
    n = 76;
end;
sum_a = 0;
sum_b = 0;
for i = 1 : n
if (typ_nut == 'd')
s1 = koef_id(i,1) * l + koef_id(i,2) * l1 + koef_id(i,3) * f + koef_id(i,4) * dd + koef_id(i,5) * omega;
    if (fi_eps == 'f')
        a = koef_abd(i,1) * 1e-4;
        bt = koef_abd(i,2) * 1e-4;
    else
        a = koef_abd(i,3) * 1e-4;
        bt = koef_abd(i,4) * 1e-4;
    end;
else
s1 = koef_ik(i,1) * l + koef_ik(i,2) * l1 + koef_ik(i,3) * f + koef_ik(i,4) * dd + koef_ik(i,5) * omega;
    if (fi_eps == 'f')
        a = koef_abk(i,1) * 1e-4;
        bt = koef_abk(i,2) * 1e-4;
    else
        a = koef_abk(i,3) * 1e-4;
        bt = koef_abk(i,4) * 1e-4;
    end;
end;
if (fi_eps == 'f')
    sin_s1 = sin(RAD_SEK_ANGL * s1);
    sa = a * sin_s1;
    sb = bt * sin_s1;
else
    cos_s1 = cos(RAD_SEK_ANGL * s1);
    sa = a * cos_s1;
    sb = bt * cos_s1;
end;
arg = RAD_SEK_ANGL * s1;
sum_a = sum_a + sa;
sum_b = sum_b + sb;
end;
nut_fi_eps = sum_a + sum_b * t;

```

Указание 1

К функциям *s0_Nut*, *utc_nut*, *:koef*, *utc_nut_fi_eps*, необходимо добавить функции *JD_epochi*, *JD_data* и выполнить расчет на требуемую дату, ориентируясь на пример

Файл *PR_s0_Nut.m*

```
%Пример расчета истинного и среднего звездного времени
%Для истинного времени nut=1, для среднего nut=0
year=1994; mon= 7; day=16; nut=1;
[s0_m_mod, s0_nut_mod,s0_m_hour, s0_m_min, s0_m_sec, s0_nut_hour, s0_nut_min, s0_nut_sec] =
s0_Nut(year, mon, day, nut);
[s0_m_mod, s0_nut_mod,s0_m_hour, s0_m_min, s0_m_sec, s0_nut_hour, s0_nut_min, s0_nut_sec]'
% Решение для приведенного примера
%s0_m_mod= 70464.3120260519
%s0_nut_mod=70465.1816369200
%s0_m_hour=0.001900000000000
%s0_m_min=0.003400000000000
%s0_m_sec=0.00243120260519
%s0_nut_hour=19 часов
%s0_nut_min= 34 минуты
%s0_nut_sec=25.1816369200 секунд
```

РАЗДЕЛ 4 Орбитальное движение навигационных спутников

4.1 Орбитальное движение спутников GPS

4.1.1 Краткие сведения из теории

Орбитального движения спутников имеет исключительно важное значение при решении основной навигационной задачи – определении координат, скорости и времени на основании измерений псевдодальностей до спутников и использовании информации, передаваемой в соответствующих сообщениях спутников. Представленные ниже программные продукты написаны в полном соответствии с интерфейсным контрольным документом для GPS [9]. Подробные сведения о примененных алгоритмах даются в книге [1] (подраздел 2. 2, стр. 87- 96, подраздел 4. 2, стр. 201- 204) и ICD GPS [9]. Отметим, что получаемые по приведенным программам данные соответствуют экспериментальным результатам, полученным с навигационных приемников различного типа.

Для изучения и исследования орбитального движения спутников GPS применяются следующие m-файлы и функции: ECEFLH.m, LLHECEF.m-функции для преобразования координат; Tim.m- функция для расчета времени; Yuma_GPS_Alm1.m- функция считывания данных альманаха и m-файл Orbita_GPS.m, рассчитывающий орбиты спутников GPS по данным альманаха, представленного в формате YUMA. Более полные сведения о перечисленных программах даются в комментариях к ним.

Цель лабораторных работ: изучение основ вторичной обработки информации в навигационных приемниках, работающих со спутникам GPS.

4.1.2 Лабораторная работа 4. 1 «Орбитальное движение спутников GPS»

Пакет программ для выполнения лабораторной работы расположен в папке **ORBITA_GPSv1**. Для выполнения работы в качестве входных данных потребуется альманах спутников GPS в формате YUMA, который можно получить с нескольких сайтов или с навигационного приемника, если приемник имеется в лаборатории.

Рекомендуется следующий порядок выполнения лабораторной работы.

1. **Задание 1.** Запишите альманах в формате YUMA на жесткий диск компьютера. По умолчанию в комплексе программ установлен альманах с именем `almanac_yuma_week0371_589824.txt` (альманах приведен в приложении). Имя альманаха внесите в отчет.
2. Создайте папку **ORBITA_GPSv1_My** и скопируйте в ее все программы из папки **ORBITA_GPSv1** и альманах, с которым Вы будете работать.

3. Запустите MatLab [7, 8].
4. Обратитесь к папке **ORBITA_GPSv1_My** и откройте ее.
5. Последовательно откройте функции и m- файлы из папки **ORBITA_GPSv1**: Orbita_GPS.m, Yuma_GPS_Alm1.m, Tim.m, LLHECEF.m, ECEFLH.m, PR_Tim.m, PR_Yuma_GPS_Alm1.m, almanac_yuma_week0371_589824.txt. Изучите тексты программ и комментарии в каждой программе. Выполните **задания 2 и 3**.
6. **Задание 2.** Выполните m- файл PR_Tim.m и изучите в командном окне полученный результат. Сформируйте на основе m- файла PR_Tim.m файл с именем PR_Tim_My.m. Введите в сформированный файл данные, соответствующие дате выполнения работы, сохраните файл и выполните файл. Результат из командного окна запишите в отчет.
7. **Задание 3.** . Выполните m- файл PR_Yuma_GPS_Alm1.m и изучите в командном окне полученный результат. Сопоставьте данные из командного окна с данными альманаха almanac_yuma_week0371_589824.txt. Сформируйте на основе m- файла PR_Yuma_GPS_Alm1.m файл с именем PR_Yuma_GPS_Alm1_My.m. Введите в сформированный файл имя записанного альманаха, сохраните файл и выполните файл, сравните результат из командного окна с данными альманаха и результат запишите в отчет.
8. Очистите командное окно из папки **ORBITA_GPSv1_My**, откройте m файл с именем Orbita_GPS.m, выполните m-файл, ознакомьтесь с графическим результатом и выполните **задание 4**.
9. **Задание 4.** Присвойте переменной Dat в файле Orbita_GPS.m имя альманаха из п.1 и введите данные о начале отсчета как указано в комментариях к файлу. Установите скорость вращения Земли равную нулю. Убедитесь, что все графики, кроме графика 1 закомментированы. Исполните файл. Результат выполнения график 1 запишите на дискету, как приложение к отчету. В отчет внесите описание графика 1.
10. **Задание 5.** Установите скорость вращения Земли, равную $7.2921151467e-005$. Выберите номер спутник, с которым будете работать. Рекомендация: для генерации вариантов для каждого студента можно задавать номер спутника равный, например, номеру дня рождения, поскольку число спутников GPS, как правило, близко к 30. Разблокируйте график 2, исполните файл. Дайте объяснения зависимостям углов азимута и видимости. График запишите на сменный диск, а интерпретацию результатов в отчет.
11. **Задание 5.** Разблокируйте график 3, исполните файл. Дайте объяснения зависимостям доплеровской частоты, широты от долготы. График запишите на сменный диск, а интерпретацию результатов в отчет.

12. **Задание 6.** Разблокируйте график 4, выполните файл. Объясните поведение проекции орбиты на плоскость XU и изменение дальности до спутника от времени. График запишите на сменный диск, а интерпретацию результатов в отчет.
13. **Задание 7.** Выберите группу спутников, принадлежащих одной из 6 орбит и повторите расчетные процедуры заданий 4- 6. График запишите на сменный диск, а интерпретацию результатов в отчет.
14. Оформите отчет. Графики результатов моделирования можно предъявить на сменном диске, как приложение к отчету.

4.1.3 Вопросы и задания для самоподготовки

1. Объясните, какой смысл вкладывается в содержание составляющих альманаха YUMA: ID, Health, Eccentricity, Time of Applicability(s), Orbital Inclination(rad), Rate of Right Ascen(r/s), SQRT(A) ($m^{1/2}$), Right Ascen at Week(rad), Argument of Perigee(rad), Mean Anom(rad), Af0(s) Af1(s/s), week.
2. Объясните зависимость изменения доплеровской частоты при движении спутника по орбите.
3. Объясните зависимость изменения дальности до спутника от времени для неподвижного наблюдателя.
4. Для каких целей используются данные альманаха в спутниковых навигационных приемниках.
5. Запишите уравнение для расчета дальностей до спутников и найдите это уравнение в текстах программ.
6. С помощью какого фрагмента программного комплекса рассчитываются углы видимости и азимута спутников?
7. Какой содержательный смысл имеет таблица в командном окне, появляющаяся после выполнения файла Orbita_GPS.m.

4.1.4 Функции и файлы из папки ORBITA_GPSv1

Функция ECEFLLH

```
function [Rx,Ry,Rz] = ECEFLLH(lon, lat,hr)
%Имя функции: ECEFLLH.m
%Функция выполняет преобразование координат
%Входные данные:lon-долгота,lat-широта,h-высота;a,b-большая
%i малая полуоси эллипсоида
%Выходные данные:Rx,Ry,Rz- координаты в ECEF
%Для WGS-84
```



```

a=6378137.0;
b=6356752.314;
n=a*a/sqrt(a*a*cos(lat)*cos(lat)+b*b*sin(lat)*sin(lat));
Rx=(n+hr)*cos(lat)*cos(lon);
Ry=(n+hr)*cos(lat)*sin(lon);
Rz=(b*b/(a*a)*n+hr)*sin(lat);

```

Функция LLHECEF

```

function [lons,lats,hrs] = LLHECEF(Xk,Yk,Zk)
%Имя функции: LLHECEF.m
%Функция выполняет преобразование координат.
%Входные данные:Rx,Ry,Rz- координаты в ECEF
%Выходные данные:lon-долгота,lat-широта,h-высота
%a,b-большая и малая полуоси эллипсоида
a=6378137.0;
b=6356752.314;
xy = sqrt(Xk*Xk + Yk*Yk);
thet = atan(Zk*a/(xy*b));
esq = 1.0-b*b/(a*a);
epsq = a*a/(b*b)-1.0;
lats = atan((Zk+epsq*b*(sin(thet)^3))/(xy-esq*a*(cos(thet)^3)));
lons = atan2(Yk,Xk);%!
if lons < 0
lons = 2*pi + lons;
end ;
n = a*a/sqrt(a*a*cos(lats)*cos(lats) + b*b*sin(lats)*sin(lats));
hrs = xy/cos(lats)-n;
end

```

Функция Tim

```

function [week,modeweek,d,dweek,weeks]=Tim(d2,h,min,s)
%Имя функции:Tim.m
%Функция Tim.m работает совместно со стандартной функцией MatLab DAYSDIF.m и рассчитывает:
%week- текущую GPS-неделю, modeweek- модифицированную GPS неделю, d- количество дней,
%dweek- день недели ,weeks- время GPS
%Входные данные:d2='10/23/2007' - 'номер месяца/номер дня месяца/номер года', h=23.0 -часы,
%;min=59.0- минуты, s=59.0- секунды на которые рассчитываются выходные данные
%d2='10/35/2003';h=23.0;min=59.0;s=59.0;
week = floor(DAYSDIF('1/6/1980',d2,3)/7);% текущая GPS-неделя
modeweek=week-1024;% модифицированная GOS-неделя
d = DAYSDIF('1/6/1980',d2,3);%количество дней
dweek=fix(d-week*7);%номер дня недели (нулевой день-воскресенье)
weeks=(dweek)*24*60*60+h*60*60+min*60+s;% время GPS в неделе (секунды)

```

Файл PR_Tim.m

```

%Пример применения функции Tim.m
%Имя m- файла: PR_Tim.m
d2= '10/13/2006';
h=22.0;
min=40.0;
s=11.0;
[week,modeweek,d,dweek,weeks]=Tim(d2,h,min,s);
[week,modeweek,d,dweek,weeks] %=1396    372    9777    5    513611-результат расчета;
%1396-неделя GPS, отсчитываемая с ночи с 5 на 6 января 1980 года, 372=1396-1024- модифициро-
ванная
% неделя GPS, 9777- количество дней прошедших с 6 января 1980 года, 5-пятый день недели (пятни-
ца),
%считая с понедельника, 513611-количество секунд от начала текущей недели.
d = DAYSDIF('1/6/1980',d2,3);%функция MatLab
Функция Yuma_GPS_Alm1
function [alm,max_kol] = Yuma_GPS_Alm1(Dat)
%Имя функции:Yuma_GPS_Alm1.m
%Функция читает данные альманаха, записанные в формате YUMA
%Входные данные записываются в переменную Dat, например,
%Dat='Имя файла альманаха в формате YUMA'
%Выходные данные:1. Численные значения альманаха спутников GPS, представляемые
%в виде структуры в переменной alm =[%alm(ID).ID(1); alm(ID).Health(2); alm(ID).e(3);
%alm(ID).TOA(4); alm(ID).deltai(5);%alm(ID).OMEGADOT(6); alm(ID).A05(7); alm(ID).omega0(8);
%alm(ID).omega(9);%alm(ID).M0(10); alm(ID).Af0(11); alm(ID).Af1(12); alm(ID).Week(13)], где
% цифра в скобках обозначает порядковый номер параметра альманаха в формате YUMA .
%Для чтения альманаха в m-файл функция записывается в виде [alm,max_kol] =
Yuma_GPS_Alm1(Dat).
%2. Количество спутников GPS записывается в переменную max_kol.
for i=1:32% цикл
    alm(i).ID = 0;% обнуление массива
    alm(i).Health=63;% обнуление массива
end;

fid =fopen(Dat,'rt');% открыть файл для чтения
%чтение данных из файла
max_kol = 0;
while notfeof(fid))

    s1=fscanf(fid,'%s',6);
    if notfeof(fid))
        lenstr = length(s1);

```

```

while (fscanf(fid,'%s',1) == '*')
end
str1 = fscanf(fid,'%s',1);
lenstr = length(str1);
n_sv = sscanf(str1,'%d');
strID = str1(1:lenstr);
ID = sscanf(strID,'%d');
alm(ID).ID = ID;

t_2=fscanf(fid,'%s',1);
alm(ID).Health=fscanf(fid,'%d',1);

t_3=fscanf(fid,'%s',1);
alm(ID).e = fscanf(fid,'%g',1);

t_4=fscanf(fid,'%s',3);
alm(ID).TOA =fscanf(fid,'%g',1);
t_5=fscanf(fid,'%s',2);
alm(ID).deltai=fscanf(fid,'%g',1);%i0
t_6=fscanf(fid,'%s',4);
alm(ID).OMEGADOT=fscanf(fid,'%g',1);
while not(fscanf(fid,'%c',1) == ':')
end
alm(ID).A05=fscanf(fid,'%g',1);
t_8=fscanf(fid,'%s',4);
alm(ID).omega0 =fscanf(fid,'%g',1);
t_9=fscanf(fid,'%s',3);
alm(ID).omega=fscanf(fid,'%g',1);
t_10=fscanf(fid,'%s',2);
alm(ID).M0=fscanf(fid,'%g',1);
t_11=fscanf(fid,'%s',1);
alm(ID).Af0=fscanf(fid,'%g',1);
t_12=fscanf(fid,'%s',1);
alm(ID).Af1=fscanf(fid,'%g',1);
t_13=fscanf(fid,'%s',1);
alm(ID).Week=fscanf(fid,'%g',1);
max_kol = max_kol+1 ;
end
end
fclose(fid)
Файл Orbita_GPS.m
clear all

```

```

%Имя m-файла:Orbita_GPS.m
%Программа рассчитывает орбиты навигационных спутников GPS
%Входные данные:
%файл альманаха в формате Yuma,имя файла альманаха присваивается
%переменной "Dat",например,Dat = 'имя файла альманаха';
%данные о начале отсчета "d2",d2='месяц/день/год';h=час;min=минута;s=секунда;
%координаты позиции приемника –lat (широта в радианах),lon (долгота в радианах,
%hr (высота в метрах);
%шаг, с каким будут рассчитываться параметры орбит (step,секунды);
%количество точек (L), в которых будут рассчитываться параметры орбит
%L=12*3600/step,L читается так: количество часов (например,12)
%число секунд в часе (3600) деленное на шаг (step)
%В программе применяются функции: Yuma_GPS_Alm1.m- считывание данных альманаха,
%заданного в формате YUMA; ECEFLLH.m, LLHECEF.m - преобразование координат;Tim.m- расчет
времени;
%Постоянные:
%скорость вращения Земли
OMEGAeDOT=7.2921151467e-005;
%или
%OMEGAeDOT=0;
mu=398600500000000;
F_CONST      = 4.442807633E-10;
%Задание цветов для графики
j_color = 0;
color6(1:16) = ['k' 'r' 'g' 'c' 'm' 'r' 'g' 'b' 'k' 'h'];
%Входные данные
Dat = 'almanac_yuma_week0371_589824.txt';
d2='10/06/2006';h=13.0;min=8.0;s=55.0;
lat = 0.88032730015257; %50 град; 26 мин.; 20.54 с
lon = 0.53109641675259; %30 град; 25 мин.; 46.4995 с
hr=184; %высота в метрах
step=300;
L=(10*3600)/step;

%Чтение альманаха

[alm,max_kol] = Yuma_GPS_Alm1(Dat);
kol = 0;
for i = 1 : max_kol
id=alm(i).ID;
if id > 0
kol = kol + 1;

```

```

nom_ns(kol) = id;
end
nom_ns;%номер навигационного спутника
end

%Преобразование координат

[Rx,Ry,Rz] = ECEFLH(lon, lat,hr);
%Rx=0;Ry=0;Rz=0;%центр масс Земли
%Выбор спутников:
%для выбора спутников вводится параметры kol-количество спутников для
%исследования и номера спутников, например, kol =4; nom_ns(1:kol) = [3 6 7 31],
%такая запись обозначает, что исследуются (рассчитываются орбиты 4 спутников
%с номерами 3,6,7,31; количество номеров спутников должно совпадать с kol
%Варианты (можно любые другие)
%kol =9
%nom_ns(1:kol) = [1 3 4 5 6 7 8 9 10];
%kol =5
%nom_ns(1:kol) = [1 13 14 26 29];    %1 спутники орбиты 1
%kol =5
%nom_ns(1:kol) = [2 5 22 28 30];    %2 спутники орбиты 2
%kol =4
%nom_ns(1:kol) = [3 6 7 31];        %3 спутники орбиты 3
%kol =5
%nom_ns(1:kol) = [4 11 15 17 24 ];   %4 спутники орбиты 4
%kol =4
%nom_ns(1:kol) = [8 9 25 27];        %5 спутники орбиты 5
%kol =5
%nom_ns(1:kol) = [10 18 20 21 23];   %6 спутники орбиты 6
%kol =29
%nom_ns(1:kol) = [1 3 4 5 6 7 8 9 10 11 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31];
%kol =14;
%nom_ns(1:kol) = [1 3 4 5 6 7 8 9 10 11 13 14 15 16];% 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31];

kol =2;
nom_ns(1:kol)=[1 3 ];
str1 = num2str(nom_ns(1:kol));
for k = 1 : kol
    i = nom_ns(k);
%Начало отсчета текущего времени
[week,modeweek,d,dweek,weeks]=Tim(d2,h,min,s);
%Расчет орбит спутников по формулам ( )

```

```

for j = 1:L % 0:L
    t(j)=weeks+step*j; %-step;
    %t1(j) = t(j)/60; %изменение дискретности текущего времени
    %d_wn = (week - alm(i).Week);%если в альманахе учтены 1024
    %d_wn = 0;
    d_wn=(modeweek-alm(i).Week);%если в альманахе не учтено 1024
    tk = t(j) + d_wn * 604800 - alm(i).TOA;
    d_wn = abs(modeweek - alm(i).Week);
    dd = 302400.0 + d_wn * 604800;
while (abs(tk) > dd)
    if tk > dd
        tk = tk - 604800;
    else
        if tk < -dd
            tk = tk + 604800;
        end
    end % if
end % while

%Справочник по альманаху- цифра в скобках обозначает порядковый номер
%параметра альманаха в формате YUMA
%alm(ID).ID(1); alm(ID).Health(2); alm(ID).e(3); alm(ID).TOA(4); alm(ID).deltai(5);
%alm(ID).OMEGADOT(6); alm(ID).A05(7); alm(ID).omega0(8); alm(ID).omega(9);
%alm(ID).M0(10); alm(ID).Af0(11); alm(ID).Af1(12); alm(ID).Week(13);

n0=sqrt((mu)/(alm(i).A05^6));
n=n0;
Mk = alm(i).M0+n*tk;
e=alm(i).e;

%Решение уравнения Кеплера
eps = 1.0E-15;
y = e * sin(Mk);
x1 = Mk;
x = y;
for k = 0 : 15
    x2 = x1;
    x1 = x;
    y1 = y;
    y = Mk - (x - e * sin(x));
    if (abs(y - y1) < eps)
        break

```

```

        end
        x = (x2 * y - x * y1) / (y - y1);
    end %k
    Ek = x;
    deltr = F_CONST * alm(i).e * alm(i).A05 * sin(Ek);
    dt1 = alm(i).Af0 + alm(i).Af1 * tk + deltr;
    tk = tk - dt1;
    nuk = atan2(sqrt(1-alm(i).e^2)*sin(Ek),(cos(Ek)-alm(i).e));
    Ek = acos((alm(i).e+cos(nuk))/(1+alm(i).e*cos(nuk)));
    Fk = nuk + alm(i).omega;
    uk = Fk;
    ik = alm(i).deltai;
    rk = (alm(i).A05^2)*(1.0-alm(i).e*cos(Ek));
    xkk = rk*cos(uk);
    ykk = rk*sin(uk);
    OMEGAk = alm(i).omega0+(alm(i).OMEGADOT-OMEGAeDOT)*tk-OMEGAeDOT*alm(i).TOA;
    %Координаты спутников
    Xk(j) = xkk *cos(OMEGAk)-ykk*cos(ik)*sin(OMEGAk);
    Yk(j) = xkk*sin(OMEGAk)+ykk*cos(ik)*cos(OMEGAk);
    Zk(j) = ykk*sin(ik);
    %Дальности до спутников
    PR(j) = sqrt((Xk(j) - Rx)^2 + (Yk(j) - Ry)^2 + (Zk(j) - Rz)^2);

    %Перевод в географическую систему если требуется
    %[lons,lats,hrs] = LLHECEF(Xk,Yk,Zk);
    %(Llon(j),Llat(j),Hhr(j)) = [lons,lats,hrs];

    xls = Xk(j) - Rx;
    yls = Yk(j) - Ry;
    zls = Zk(j) - Rz;
    range1 = sqrt(xls*xls+yls*yls+zls*zls);
    if j>1
    doppler(j-1) = (range1 - range2) * 5.2514 / step;%расчет доплеровской частоты
    end
    range2 = range1;
    P = sqrt(Rx * Rx + Ry * Ry + Rz * Rz);
    tdot = ( Rx*xls+Ry*yls+Rz*zls)/range1/P;
    xll = xls/range1;
    yll = yls/range1;
    zll = zls/range1;
    %Расчет угла видимости
    if tdot >= 1.00

```

```

b = 0.0;
elseif tdot <= -1.00
    b = pi;
else
b = acos( tdot);
end
satang = pi/2.0 - b;
TT(j) =satang;

%Расчет угла азимута
xn =-cos(lon)*sin(lat);
yn =-sin(lon)*sin(lat);
zn = cos(lat);
xe =-sin(lon);
ye = cos(lon);
xaz  = xe*xll + ye*yll;
yaz  = xn*xll + yn*yll + zn*zll;
if (xaz == 0) or (yaz == 0)
az(j)= 0;
else
az(j) = atan2(xaz,yaz);
end
if az(j) < 0
az(j) = az(j) + pi*2;
end
    end % j
for j = 1:L
    [Llon(j),Llat(j),Hhr(j)] = LLHECEF(Xk(j),Yk(j),Zk(j));%преобразование координат
if j > 1
if abs(Llon(j)-Llon(j-1)) > pi
Llon(j) = Llon(j) + 2*pi;
        end
        end
end

j_color = j_color + 1;
if (j_color > 14 )
j_color = 1;
end
%F_ont=get(gcf,'CurrentAxes'),'FontSize',16,'FontName','TimesNewRoman';%формат текста на графиках
S = color6(j_color);
%Графика

```



```

%График 1 для вывода графика убрать символы %{ и %} относящиеся к данному графику
%{
h_F1=gca;
plot3( Xk(:),Yk(:),Zk(:),S),
hold on,
set(get(gcf,'CurrentAxes'),'FontSize',12,'FontName','TimesNewRoman');
set(h_F1,'Position',[0.1 0.1 0.85 0.9]) ;
xlabel('Координата X')
ylabel('Координата Y'),
zlabel ('Координата Z'),grid on
%}
%{
%График 2 для вывода графика убрать символы %{ и %} относящиеся к данному графику
subplot(2,1,1),plot(t,az(:),S),
set(get(gcf,'CurrentAxes'),'FontSize',12,'FontName','TimesNewRoman')
hold on,
xlabel('Время'),
ylabel('Угол азимута,радиан')
grid on

subplot(2,1,2),plot(t,TT(:),S),
set(get(gcf,'CurrentAxes'),'FontSize',12,'FontName','TimesNewRoman')
hold on,xlabel ('Время'),
ylabel('Угол видимости'),
grid on
%}
%{
%График 3 для вывода графика убрать символы %{ и %} относящиеся к данному графику
subplot(2,1,1),plot(t(1:(j-1)),doppler(:),S),
set(get(gcf,'CurrentAxes'),'FontSize',12,'FontName','TimesNewRoman')
hold on,
xlabel('Время '),
ylabel('Доплеровская частота'), grid on
subplot(2,1,2), plot(Llon(1:j),Llat(1:j),S),
set(get(gcf,'CurrentAxes'),'FontSize',12,'FontName','TimesNewRoman')
hold on, xlabel('Долгота'),ylabel('Широта')
grid on
%}
%График 4 для вывода графика убрать символы %{ и %} относящиеся к данному графику
%hF=figure('Color','w','MenuBar','none')

subplot(1,2,1) ,plot(Xk(:),Yk(:),S),

```

```

set(get(gcf,'CurrentAxes'),'FontSize',12,'FontName','Times New Roman')
hold on,
xlabel('Проекция орбит на плоскость XY')
grid on

subplot(1,2,2), plot(t,PR(:),S),
set(get(gcf,'CurrentAxes'),'FontSize',12,'FontName','Times New Roman')
hold on,
xlabel('Время'),
ylabel('Дальность,метр '),grid on
end % i
clear

```

Для работы с приведенными программами необходимо их скопировать в отдельную папку и изменяя входные данные выполнить желательные расчеты.

4.1.5 Примеры расчетов с помощью m-файла :Orbita_GPS.m.

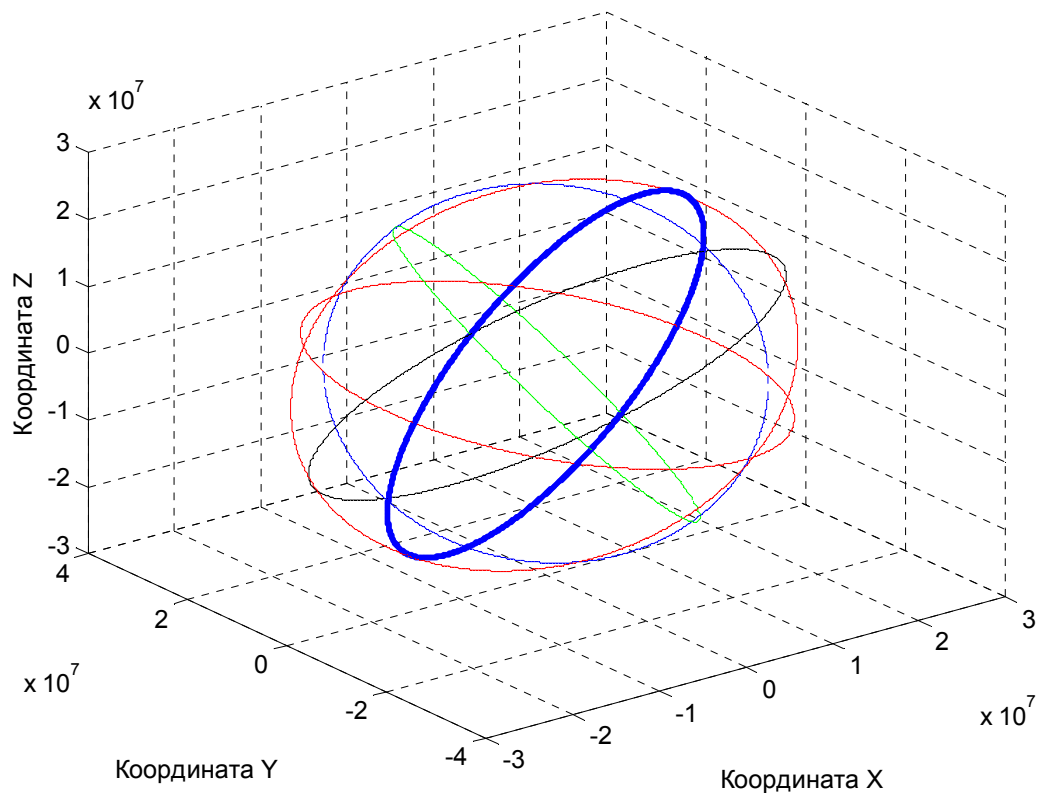


Рис. 4.1. Шесть орбит спутников GPS (из графика 1 файла:Orbita_GPS.m)

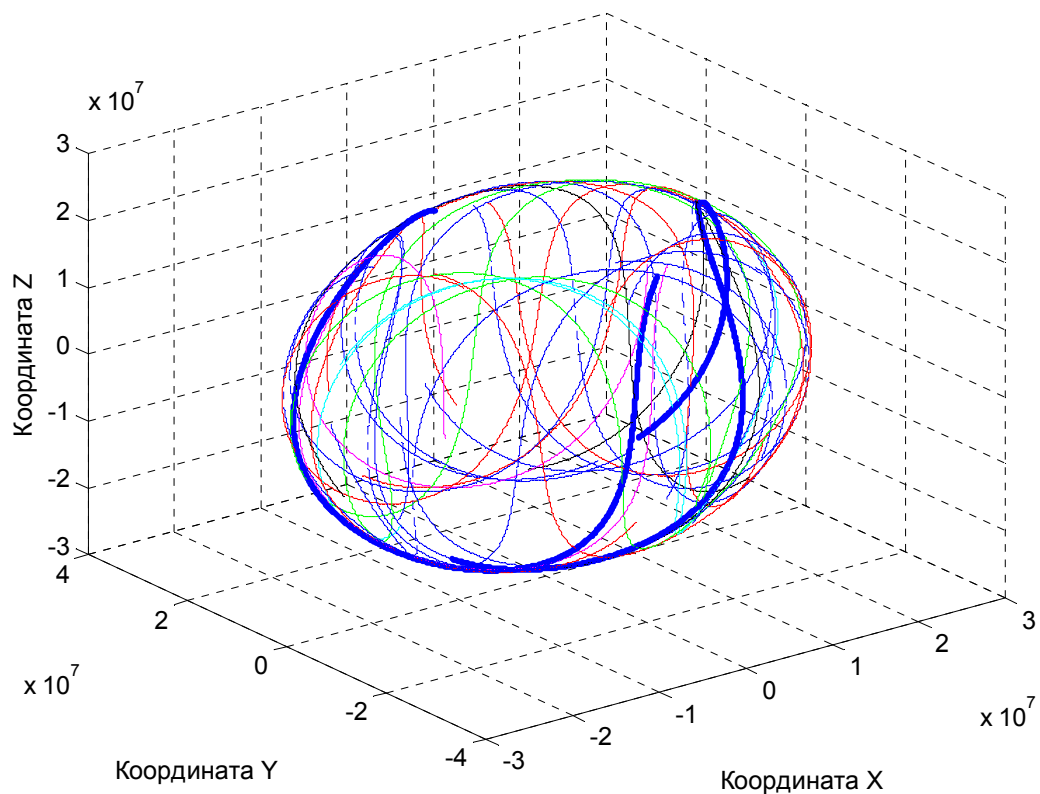


Рис. 4.2. Орбиты 29 спутников за 12 часов (из графика 1 файла:Orbita_GPS.m)

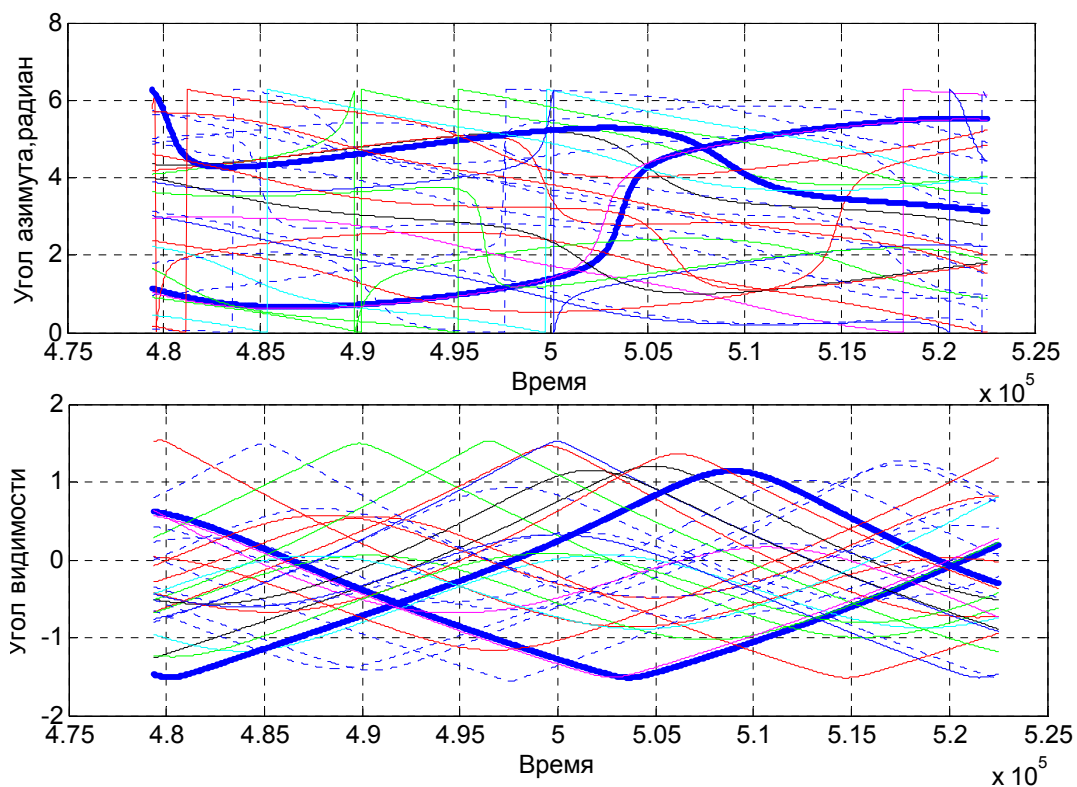


Рис. 4.3. Углы азимута и видимости 29 спутников за 12 часов (из графика 2 файла:Orbita_GPS.m)

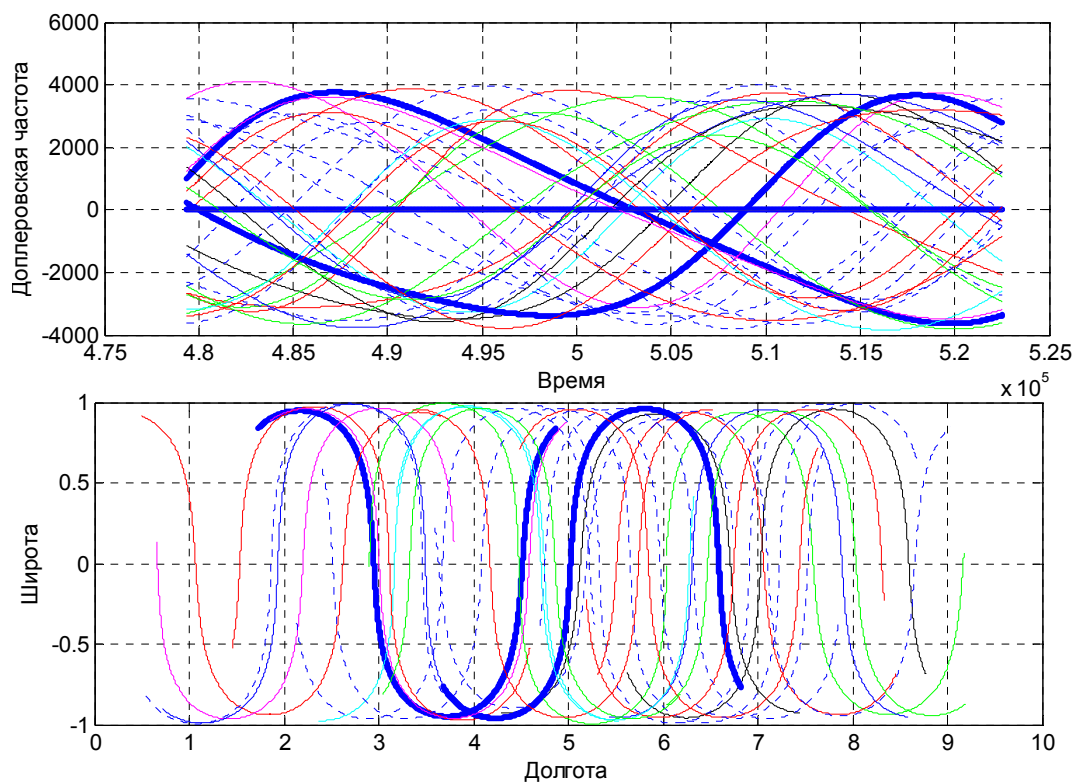


Рис. 4.4. Доплеровская частота и следы 29 спутников за 12 часов (из графика 3 файла:Orbita_GPS.m)

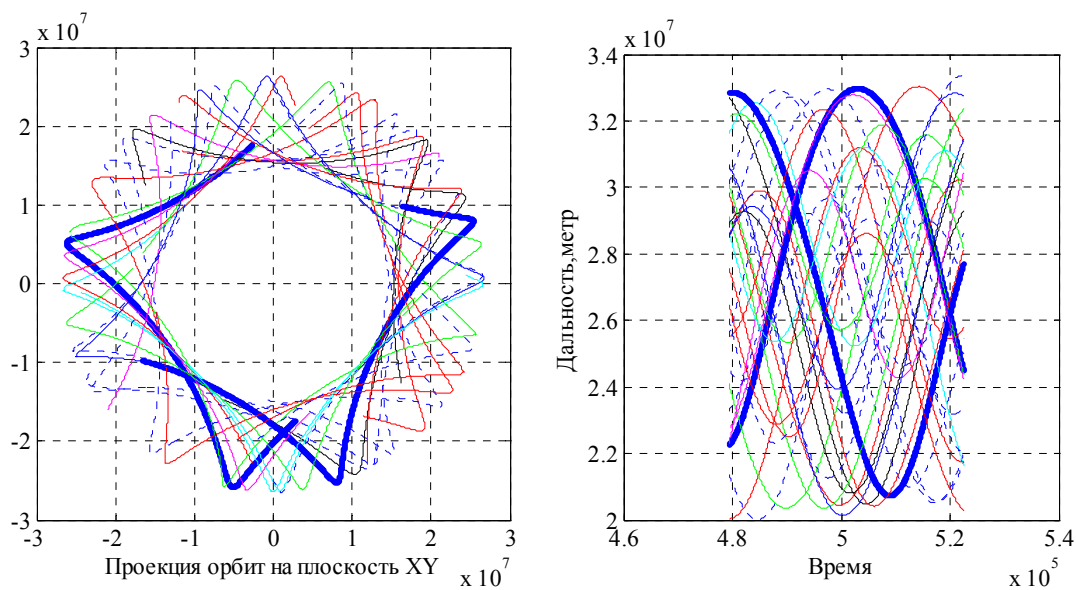


Рис. 4.5. Проекция орбит и дальности до 29 спутников за 12 часов (из графика 4 файла:Orbita_GPS.m)

Файл Orbita_GPS.m предоставляет широкие возможности для исследования самых различных вариантов орбитального движения навигационных спутников GPS исследователю, работающему с данной программой.

Приведем еще один вариант файла- программы, решающей задачу фильтрации спутников по здоровью и видимости из конкретной точки в заданное время.

4.1.6 Лабораторная работа 4. 2 «Наблюдение спутников GPS»

Пакет программ для выполнения лабораторной работы нужно сформировать самостоятельно. Для выполнения работы в качестве входных данных потребуется альманах спутников GPS в формате YUMA, который можно получить с нескольких сайтов или с навигационного приемника, если приемник имеется в лаборатории.

Рекомендуется следующий порядок действий для выполнения лабораторной работы.

1. **Задание 1.** Запишите альманах в формате YUMA на жесткий диск компьютера. По умолчанию в комплексе программ установлен альманах с именем `almanac_yuma_week0371_589824.txt` (альманах приведен в приложении). Имя альманаха внесите в отчет.
2. **Задание 2.** Создайте папку **ORBITA_GPSv2_My**.
3. В папку **ORBITA_GPSv2_My** запишите: файл альманаха, с которым будете работать, скопируйте функции ECEFULLH, LLHECEF, Yuma_GPS_Alm1, файл Tim.m из папки **ORBITA_GPSv1** и файл Orbita_GPS_1.m из папки **ORBITA_GPSv2**. Эти данные внесите в отчет в соответствующий раздел **задания 2**
4. **Задание 3.** Пользуясь комментариями к файлу Orbita_GPS_1.m установите время, в которое должны наблюдаться спутники. Выполните файл и результаты, полученные на графике, занесите в отчет.
5. **Задание 4.** Во входных данных файла Orbita_GPS_1.m измените время на 6 часов, выполните файл и полученную конфигурацию наблюдаемых спутников занести в отчет. Объяснить разницу между результатами задания 3 и задания 4.
6. **Задание 5.** Во входных данных файла Orbita_GPS_1.m, пользуясь комментариями установить вывод наблюдаемых спутников в таблицу, появляющуюся в командном окне после исполнения файла и сопоставить эту таблицу с данными, полученными с навигационного приемника на установленное в файле время.

Примечание: практически все навигационные приемники имеют опцию для отображения на компьютере наблюдаемых спутников в полярной системе координат.

4.1.7 Вопросы и задания для самоподготовки

1. Повторите вопросы к лабораторной работе 4. 1.
2. Рассчитайте, сколько времени спутник GPS находится в зоне видимости неподвижного наблюдателя?
3. В каких пределах изменяется угол видимости спутника?
4. В каких пределах изменяется азимут спутника?
5. Сколько спутников GPS могут одновременно находиться в зоне видимости?

4.1.8 Файл из папки ORBITA_GPSv2

Файл Orbita_GPS_1.m

clear all

```
%Имя m-файла:Orbita_GPS_1.m
%Программа рассчитывает углы видимости, азимута и положение видимых спутников на заданный
момент времени навигационных спутников GPS
%Входные данные:
%файл альманаха в формате Yuma,имя файла альманаха присваивается
%переменной "Dat",например,Dat = 'имя файла альманаха';
%данные о начале отсчета "d2",d2='месяц/день/год';h=час;min=минута;s=секунда;
%координаты позиции приемника -lat(широта в радианах),lon (долгота в радианах,
%hr (высота в метрах);
%шаг с каким будут рассчитываться параметры (step,секунды);
%количество точек (L), в которых будут рассчитываться параметры орбит
%L=12*3600/step,L читается так: количество часов (например,12)
%число секунд в часе (3600) деленное на шаг (step)
%Постоянные:
%скорость вращения Земли
OMEGAeDOT=7.2921151467e-005;
A_WGS84=6378137.0;
B_WGS84=6356752.314;

%или
%OMEGAeDOT=0;
mu=3986005000000000;
F_CONST      = 4.442807633E-10;
kt=3;%установка времени на титульной надписи графика, определяется параметрами d2'; h; min; s и j
или L;
%Задание цветов для графики
j_color = 0;
color6(1:14) = ['k' 'b' 'g' 'r' 'c' 'm' 'r' ':' 'g' ':' 'b' ':' 'k' 'h'];
%Входные данные
Dat = 'almanac_yuma_week0371_589824.txt';
d2='10/05/2006'; h=23.0; min=0.0; s=0.0;
lat = 0.88032730015257;%50 градусов 26минут 20.54 секунд
lon = 0.53109641675259;%30 градусов 25 минут 46.4995секунд
hr=187.488;% метров
X_label=['Широта' ':' num2str(lat) ':' 'долгота' ':' num2str(lon) ':' 'высота' ':' num2str(hr)];
```

```

step=3600;%шаг отсчета времени в секундах (300=5 минутам);шаг можно изменять
L=(24*3600) / step;% убрать % перед L для вывода таблицы улов видимости и азимута
%L=1;% установить % перед L для вывода таблицы улов видимости и азимута
%Чтение альманаха

```

```

[alm,max_kol] = Yuma_GPS_Alm1(Dat);

```

```

nom = 1;
i = 0;
k = 0;
for i = 1 : max_kol
    id = alm(nom).ID;
    Health = alm(nom).Health;
    % fprintf('1: i=%i k=%i nom=%i id=%i Health=%i \n', i, k, nom, id, Health);
    if ( id > 0)
        if ( Health == 0)
            k = k + 1;
            nom_ns(k) = id;
            % fprintf('2: i=%i k=%i nom=%i id=%i Health=%i \n', i, k, nom, id, Health);
            nom = nom + 1;
        else
            nom = nom + 1;
        end;
    else
        nom = nom + 1;
    end;
end; % i
kol = k;
fprintf('kol=%i \n', kol);

```

```

% nom_ns%номер навигационного спутника
[Rx,Ry,Rz] = ECEFLLH(lon, lat,hr);
%Rx=0;Ry=0;Rz=0;%центр масс Земли
%Начало отсчета текущего времени
[week,modeweek,d,dweek,weeks]=Tim(d2,h,min,s);

```

```

for j = 1:L % 0:L
    t(j)=weeks+step*(j); %-step;
    %t1(j) = t(j)/60; %изменение дискретности текущего времени
    %d_wn = (week - alm(i).Week);
    %d_wn = 0;
    d_wn=(modeweek-alm(i).Week);%если в альманахе не учтено 1024

```

```

tk = t(j) + d_wn * 604800 - alm(i).TOA;
d_wn = abs(modeweek - alm(i).Week);
dd = 302400.0 + d_wn * 604800;

for k = 1 : kol
    i = nom_ns(k) ;

%Расчет орбит спутников по формулам
if ( ( alm(i).A05 > 0 ) & ( alm(i).Health == 0 ) )

while (abs(tk) > dd)
    if tk > dd
        tk = tk - 604800;
    else
        if tk < -dd
            tk = tk + 604800;
        end
    end % if
end % while

%Справочник по альманаху- цифра в скобках обозначает порядковый номер
%параметра альманаха в формате YUMA
%alm(ID).ID(1); alm(ID).Health(2); alm(ID).e(3); alm(ID).TOA(4); alm(ID).deltai(5);
%alm(ID).OMEGADOT(6); alm(ID).A05(7); alm(ID).omega0(8); alm(ID).omega(9);
%alm(ID).M0(10); alm(ID).Af0(11); alm(ID).Af1(12); alm(ID).Week(13);

n0=sqrt((mu) / (alm(i).A05^6));
j2 = 1082.68E-6;

re = (A_WGS84 + B_WGS84) / 2.;
sin55 = sin(55.0 * pi / 180.0);
dn = 1.5 * j2 * re * re / (alm(i).A05^4 ) * (1. - 1.5 * sin55 * sin55);
%dn = 0;
n=n0 * (1 + dn);
Mk = alm(i).M0 + n*tk;
e=alm(i).e;

%решение уравнения Кеплера
eps = 1.0E-15;
y = e * sin(Mk);
x1 = Mk;
x = y;

```



```

for k = 0 : 15 % количество итераций
    x2 = x1;
    x1 = x;
    y1 = y;
    y = Mk - (x - e * sin(x));
    if (abs(y - y1) < eps)
        break
    end
    x = (x2 * y - x * y1) / (y - y1);
end % kepler
Ek = x;
deltr = F_CONST * alm(i).e * alm(i).A05 * sin(Ek);
dt1 = alm(i).Af0 + alm(i).Af1 * tk + deltr;
tk = tk - dt1;
vd = 1. - alm(i).e * cos(Ek);
nuk = atan2(sqrt(1-alm(i).e^2)*sin(Ek) / vd,(cos(Ek)-alm(i).e) / vd);
Ek = acos((alm(i).e+cos(nuk))/(1+alm(i).e*cos(nuk)));

Fk =nuk + alm(i).omega;
uk =Fk;

ik=alm(i).deltai;

rk =(alm(i).A05^2)*(1.0-alm(i).e*cos(Ek));
xkk =rk*cos(uk);
ykk =rk*sin(uk);

OMEGAk =alm(i).omega0+(alm(i).OMEGADOT-OMEGAeDOT)*tk-OMEGAeDOT*alm(i).TOA;
Xk(j,i) = xkk *cos(OMEGAk)-ykk*cos(ik)*sin(OMEGAk);
Yk(j,i) = xkk*sin(OMEGAk)+ykk*cos(ik)*cos(OMEGAk);
Zk(j,i) = ykk*sin(ik);
%Dальности до спутников
PR(j,i) = sqrt((Xk(j,i) - Rx)^2 + (Yk(j,i) - Ry)^2 + (Zk(j,i) - Rz)^2);
%Перевод в географическую систему если требуется
%[lons,lats,hrs] = LLHECEF(Xk,Yk,Zk);
%(Llon(j),Llat(j),Hhr(j)) = [lons,lats,hrs];
%расчет угла видимости спутника
xls = Xk(j,i) - Rx;
yls = Yk(j,i) - Ry;
zls = Zk(j,i) - Rz;
range1 = sqrt(xls*xls+yls*yls+zls*zls);
if j>1

```

```

    doppler(j-1) = (range1 - range2) * 5.2514 / step;
end
range2 = range1;
P = sqrt(Rx * Rx + Ry * Ry + Rz * Rz);
tdot = ( Rx*xls+Ry*yIs+Rz*zIs)/range1/P;

xll = xls/range1;
yll = yIs/range1;
zll = zIs/range1;

if tdot >= 1.00
    b = 0.0;
elseif tdot <= -1.00
    b = pi;
else
    b = acos( tdot);
end
satang = pi/2.0 - b;
TT = satang;
TT(j,i) = TT;%угол видимости спутников
%расчет угла азимута спутников
xn = -cos(lon)*sin(lat);
yn = -sin(lon)*sin(lat);
zn = cos(lat);
xe = -sin(lon);
ye = cos(lon);
xaz = xe*xll + ye*yll;
yaz = xn*xll + yn*yll + zn*zll;
if (xaz == 0) or (yaz == 0)
    az(j) = 0;
else
az(j,i) = atan2(xaz,yaz);
end
if az(j,i) < 0
    az(j,i) = az(j,i) + pi*2;
end

AZ(j,i) = az(j,i) *180/pi;%угла азимута спутников в градусах
EL(j,i) = TT(j,i) *180/pi;%угла видимости спутников в градусах

% ПЕРЕСЧЕТ ВРЕМЕНИ
A(j)=mod(t(j),86400);

```

```

her(j)=floor(A(j)/3600);
m(j)=floor((A(j)-her(j)*3600)/60);
sek(j)=A(j)-her(j)*3600-m(j)*60;

%Построение полярной системы координат
if EL(j,i) < 0
    elp = 180;
else
    elp = (EL(j,i)-90);
end;
azp = (AZ(j,i) + 90.0);
rad = pi / 180;
x0 = 0; y0 = 0;
xt(j,i) = (elp * cos(azp * rad));
yt(j,i) = -(elp * sin(azp * rad));
end % i = ns
end; % if ( alm(i).A05 > 0 )
    j_color = j_color + 1;
if (j_color > 14 )
    j_color = 1;
end
S = color6(j_color);

end % j = time
%ВНИМАНИЕ. Для вывода времени визуализации спутников на график установите kt
t_title=[d2 ' ' num2str(her(kt)) ':' num2str(m(kt)) ':' num2str(sek(kt))];
%X_label=['Широта' ':' num2str(lat) ';' 'долгота' ':' num2str(lon) ';' 'высота' ':' num2str(hr)];
%num2ctr(lat)
%num2str(her(kt))
%X_label=['66' ':'];
n = 6;
max_n = max(nom_ns);
n_end = mod(max(nom_ns),n);
n_end = mod(kol, n);
n2 = fix(kol / n) * n - n + 1;
%Формирование таблицы вывода времени UTC (Time), GPS (Tgps в секундах), номера спутника
(nns),
% углов видимости и азимута от времени и номера спутника
for i=1:n:kol
    fprintf(' Time      Tgps ');
    for k=1: n
        nns = nom_ns(i+k-1);

```

```

fprintf('    %2i    ', nns);
end;
fprintf('\n');
for j=1:L
    fprintf('%2i:%2i:%2i  %i  ',her(j),m(j),sek(j), t(j));
    for k=1: n
        nns =  nom_ns(i+k-1);
        fprintf('%6.1f*%6.1f  ', EL(j,nns), AZ(j,nns) );
    end;
    fprintf('\n');
end ; % J=1:L
if (i) == (n2)
    n = n_end;
end;
end% i
hold on
%Окружности уровней на круговой диаграмме видимости спутников
k1 = 10;
k2 = 30;
k3 = 50;
k4 = 70;
k5 = 85;
k6=90;
n=0;
for k=1:5:365
    n=n+1;
    m1 = pi / 180;
    x(n)=cos(k*m1);
    y(n)=sin(k*m1);
end;
%График круговой диаграммы
plot(k1*x(:),k1*y(:),'k:', k2*x(:),k2*y(:),'k:', k3*x(:),k3*y(:),'k:',k4*x(:),k4*y(:),'k:', k5*x(:),k5*y(:),'r',
k6*x(:),k6*y(:),'r:');
text(5, 10,'80','FontSize',12,'FontName','TimesNewRoman');
text(18, 23,'60','FontSize',12,'FontName','TimesNewRoman');
text(32, 37,'40','FontSize',12,'FontName','TimesNewRoman');
text(45, 50,'20','FontSize',12,'FontName','TimesNewRoman');
text(55, 60,'5','FontSize',12,'FontName','TimesNewRoman');
text(62, 67,'0','FontSize',12,'FontName','TimesNewRoman');
grid on;
%Построение изображений видимых спутников на круговой диаграмме
i=1;

```

```

for k=1:kol
i = nom_ns(k) ;
plot(xt(kt,i),yt(kt,i), 'Marker' , 'd', 'MarkerSize', 20 )
title(t_title);
xlabel(X_label, 'FontSize', 12, 'FontName', 'TimesNewRoman')
set(get(gcf, 'CurrentAxes'), 'FontSize', 14, 'FontName', 'TimesNewRoman')
hold on
str1 = num2str(i, 2);
text(xt(kt,i), yt(kt,i), str1, 'FontSize', 14, 'FontName', 'TimesNewRoman', 'HorizontalAlignment', 'center' );
axis([-100 100 -100 100]);
%axis([-90 90 -90 90]);
end
clear

```

4.1.9 Пример выполнения файла Orbita_GPS_1.m

Расчет углов видимости и азимута с интервалом 1 час для 28 здоровых спутников, выполненный m-файлом Orbita_GPS_1.m приведен в таблице. Знак "-" (минус) обозначает невидимые в указанное время спутники.

Таблица

Углы видимости и азимута спутников GPS 6 октября 2006 года

Time	Tgps	1	2	3	4	5	6
0:0:0	432000	72.9*100.5	-46.1*321.4	-13.0*175.9	-16.4*313.7	-0.8*27.4	-53.0*75.4
1:0:0	435600	52.0*58.3	-25.7*328.2	-27.7*162.0	6.1*317.2	-0.0*5.2	-38.8*53.7
2:0:0	439200	26.6*60.6	-3.9*329.0	-32.9*141.1	28.5*309.9	-11.0*346.3	-21.6*38.6
3:0:0	442800	4.0*71.8	16.8*320.1	-27.1*118.8	42.2*283.5	-28.3*332.7	-6.8*22.4
4:0:0	446400	-16.0*86.0	29.1*296.8	-13.6*99.8	35.2*249.6	-46.7*319.6	-0.0*2.2
5:0:0	450000	-34.6*100.8	23.9*267.4	2.0*81.1	15.4*229.1	-61.5*296.6	-5.1*341.6
6:0:0	453600	-52.7*114.0	6.8*246.0	12.6*58.4	-4.9*215.1	-63.7*256.6	-19.2*324.9
7:0:0	457200	-70.8*120.3	-11.3*229.4	10.3*34.4	-20.3*199.3	-51.4*233.3	-36.6*310.1
8:0:0	460800	-84.6*49.6	-24.2*211.1	-4.1*17.8	-26.9*178.9	-33.3*230.4	-51.8*289.9
9:0:0	464400	-68.4*353.7	-27.6*189.7	-23.9*10.2	-22.2*157.6	-13.3*236.8	-57.4*257.5
10:0:0	468000	-47.7*352.9	-20.0*170.2	-44.5*8.0	-8.1*140.1	8.1*248.1	-48.6*230.5
11:0:0	471600	-26.1*352.4	-3.6*155.9	-64.3*5.9	11.5*124.4	31.1*262.4	-31.2*221.6
12:0:0	475200	-5.0*346.7	18.1*144.0	-81.0*337.1	30.4*103.0	56.0*278.8	-10.3*223.8
13:0:0	478800	11.4*331.9	40.6*125.4	-74.8*252.1	36.5*70.8	82.8*306.0	12.5*232.3
14:0:0	482400	16.0*308.0	50.7*86.9	-57.8*249.5	23.4*46.1	68.6*120.0	36.7*245.2
15:0:0	486000	6.2*284.5	36.2*55.6	-40.2*259.5	2.1*38.2	40.8*134.6	61.8*263.1
16:0:0	489600	-10.8*266.1	12.6*48.8	-21.9*272.2	-19.7*40.7	14.9*144.1	85.3*335.1
17:0:0	493200	-27.0*247.9	-10.3*53.7	-1.9*284.8	-39.8*49.3	-8.6*148.9	65.4*95.7
18:0:0	496800	-36.0*225.2	-30.7*64.4	20.6*294.7	-58.1*62.6	-28.9*147.0	39.9*114.6
19:0:0	500400	-33.0*201.2	-49.0*78.6	45.7*296.6	-74.9*82.6	-44.1*134.3	15.3*127.8
20:0:0	504000	-18.7*184.9	-65.8*95.3	67.5*266.4	-86.4*204.9	-49.5*108.7	-7.9*136.5
21:0:0	507600	3.0*176.9	-81.9*118.7	58.1*204.2	-71.6*272.3	-42.3*82.0	-29.2*139.1
22:0:0	511200	29.5*172.5	-81.1*291.1	31.7*188.9	-54.5*291.2	-27.5*62.8	-47.0*131.3
23:0:0	514800	58.2*161.8	-63.7*310.7	6.5*183.1	-35.7*305.0	-11.4*45.9	-56.7*106.0
Time	Tgps	7	8	9	10	11	13
0:0:0	432000	-36.9*144.7	-49.9*251.3	-16.8*0.2	-74.1*251.9	68.9*226.5	-14.8*220.0
1:0:0	435600	-42.3*120.7	-45.2*224.0	-37.5*353.1	-57.7*273.4	47.4*189.5	7.9*228.7

2:0:0	439200	-35.6*95.9	-29.9*210.4	-57.8*346.7	-40.1*290.0	21.1*181.8	32.2*242.9
3:0:0	442800	-21.4*76.8	-9.3*208.3	-74.7*323.0	-20.7*303.1	-2.3*176.2	57.3*264.1
4:0:0	446400	-6.0*59.4	14.3*213.4	-74.0*255.0	1.2*311.6	-20.6*166.4	79.9*324.4
5:0:0	450000	4.2*39.2	40.3*223.8	-58.4*239.5	25.1*311.9	-31.3*149.2	66.9*79.8
6:0:0	453600	3.0*17.6	67.7*241.6	-40.8*245.2	46.2*294.2	-31.8*126.8	42.1*106.9
7:0:0	457200	-9.4*0.8	82.7*31.9	-22.6*256.9	48.5*252.7	-22.5*105.9	17.7*123.8
8:0:0	460800	-27.8*350.2	56.8*74.6	-3.1*270.7	29.2*225.3	-8.2*87.4	-5.4*135.3
9:0:0	464400	-47.8*341.8	31.9*93.6	18.8*284.5	5.9*211.8	5.1*67.5	-27.0*140.9
10:0:0	468000	-66.0*324.8	8.8*109.5	44.1*295.2	-13.8*199.3	9.8*44.4	-46.0*136.8
11:0:0	471600	-72.6*272.9	-13.0*122.3	72.2*288.5	-26.3*181.9	1.9*23.9	-58.5*114.6
12:0:0	475200	-59.7*240.4	-33.5*130.2	70.8*179.4	-28.2*159.9	-14.7*11.4	-56.5*78.5
13:0:0	478800	-41.3*239.1	-52.2*128.9	41.1*169.9	-18.8*139.5	-34.5*5.5	-41.9*55.1
14:0:0	482400	-21.9*247.3	-65.7*106.0	13.7*170.5	-1.7*122.8	-54.3*1.9	-23.3*40.4
15:0:0	486000	-2.0*259.5	-63.8*62.2	-9.6*168.4	17.8*104.8	-72.5*349.4	-6.2*24.7
16:0:0	489600	19.0*273.1	-48.3*38.9	-27.7*159.9	30.9*78.0	-78.7*279.1	3.0*3.8
17:0:0	493200	41.9*285.7	-29.1*27.1	-38.1*142.0	26.3*48.7	-64.3*249.6	-1.0*341.5
18:0:0	496800	67.2*291.3	-10.8*14.6	-37.4*118.0	7.8*33.4	-46.9*253.2	-15.6*323.7
19:0:0	500400	80.9*182.9	1.6*356.3	-26.4*96.8	-14.2*30.8	-28.7*263.5	-34.1*309.3
20:0:0	504000	53.1*159.6	2.0*333.5	-10.5*78.8	-35.5*35.9	-9.5*275.5	-50.9*290.7
21:0:0	507600	24.6*162.0	-9.6*313.0	4.2*58.7	-55.1*46.5	11.6*286.5	-58.8*258.6
22:0:0	511200	-1.0*162.8	-26.6*296.0	9.2*34.3	-73.1*63.8	35.2*293.0	-51.4*228.5
23:0:0	514800	-22.5*157.9	-42.4*276.7	0.0*12.6	-86.8*161.9	59.9*284.1	-34.3*218.3
Time	Tgps	14	16	17	18	19	20
0:0:0	432000	32.0*51.0	-27.1*150.4	21.4*314.6	-29.0*86.9	8.0*191.3	43.9*274.1
1:0:0	435600	7.8*52.7	-15.8*130.6	24.4*287.5	-47.2*101.0	-13.6*182.3	69.5*293.5
2:0:0	439200	-13.9*62.5	1.8*113.9	12.0*263.0	-64.8*112.6	-28.2*166.8	82.3*103.7
3:0:0	442800	-33.2*76.3	19.8*94.3	-6.0*245.0	-82.2*105.9	-32.3*144.6	54.2*130.7
4:0:0	446400	-50.8*92.5	28.4*66.9	-21.7*227.4	-77.6*344.4	-25.0*122.4	26.9*142.5
5:0:0	450000	-67.7*110.3	19.9*41.5	-29.5*206.1	-58.2*342.6	-10.1*104.1	1.7*149.5
6:0:0	453600	-84.5*137.5	0.6*29.8	-26.1*184.3	-37.7*345.2	6.8*85.8	-20.6*150.7
7:0:0	457200	-77.3*309.4	-21.2*28.7	-12.3*168.0	-16.6*344.3	18.0*62.5	-38.7*142.8
8:0:0	460800	-58.4*324.3	-42.2*34.1	8.5*156.9	3.2*336.4	15.1*37.6	-48.9*121.5
9:0:0	464400	-37.9*332.9	-61.8*44.6	33.2*145.0	16.4*318.0	-0.4*21.7	-46.3*92.9
10:0:0	468000	-16.2*336.3	-80.0*62.6	54.3*116.9	15.8*292.6	-21.1*15.7	-33.3*71.1
11:0:0	471600	5.6*332.1	-82.4*235.2	51.7*69.6	2.7*270.4	-42.4*16.0	-16.7*54.3
12:0:0	475200	22.7*315.9	-65.8*259.5	29.6*52.6	-14.3*252.4	-63.0*19.0	-2.5*36.1
13:0:0	478800	25.8*288.2	-48.8*276.6	5.8*54.5	-27.9*232.9	-82.0*10.5	2.5*14.2
14:0:0	482400	12.8*263.3	-30.7*291.6	-15.7*63.9	-32.4*209.8	-78.2*246.0	-5.1*353.6
15:0:0	486000	-6.0*245.5	-10.8*303.4	-34.9*77.2	-25.3*188.9	-60.6*249.7	-21.4*338.7
16:0:0	489600	-22.4*228.1	11.6*310.0	-52.4*92.5	-8.7*175.5	-43.1*262.6	-40.5*326.9
17:0:0	493200	-30.8*206.5	35.2*306.5	-69.3*108.5	14.0*167.6	-25.0*276.8	-58.1*309.7
18:0:0	496800	-27.5*184.0	52.5*279.1	-86.3*126.8	40.3*158.8	-5.5*290.1	-66.5*271.9
19:0:0	500400	-13.4*167.4	45.7*236.5	-75.9*314.8	63.3*129.3	16.4*300.1	-57.8*237.6
20:0:0	504000	8.0*156.5	22.9*215.8	-56.9*325.9	58.5*70.8	40.8*302.1	-40.6*229.7
21:0:0	507600	33.4*145.0	-0.2*204.1	-36.6*333.1	34.4*57.7	62.7*277.8	-20.9*234.4
22:0:0	511200	55.4*117.0	-18.5*190.6	-15.1*335.4	10.4*62.9	57.7*218.9	-0.1*244.9
23:0:0	514800	52.9*67.5	-28.3*171.4	6.1*330.2	-11.0*74.1	32.1*198.7	21.9*259.0
Time	Tgps	21	22	23	24	25	26
0:0:0	432000	-48.3*124.6	2.8*84.1	14.5*215.6	-11.0*311.8	30.7*126.9	-65.7*19.8
1:0:0	435600	-65.0*113.1	-17.1*97.8	40.5*226.5	11.5*315.2	46.4*96.8	-84.2*14.0
2:0:0	439200	-70.3*67.1	-35.9*110.3	67.9*244.8	33.6*306.9	40.7*61.0	-77.5*238.2
3:0:0	442800	-57.8*33.5	-54.4*118.5	82.5*33.8	45.9*277.6	20.4*47.0	-60.5*247.6
4:0:0	446400	-39.2*21.2	-71.9*111.3	57.0*77.0	36.3*243.2	-1.9*47.8	-43.4*262.3
5:0:0	450000	-20.1*11.7	-77.5*41.6	32.4*95.9	15.0*224.2	-22.8*56.0	-25.3*277.7
6:0:0	453600	-4.0*358.1	-61.3*9.3	9.4*111.6	-6.0*210.8	-41.8*68.7	-5.2*291.9
7:0:0	457200	3.6*338.1	-40.9*3.4	-12.3*124.0	-21.6*194.8	-59.4*84.8	18.1*302.6
8:0:0	460800	-1.2*316.0	-20.1*358.0	-32.8*131.3	-27.7*173.6	-76.3*106.5	44.5*304.3
9:0:0	464400	-15.1*296.8	-1.6*347.3	-51.4*129.3	-22.3*151.8	-85.4*255.1	66.9*271.6
10:0:0	468000	-31.1*278.7	9.3*328.4	-64.6*106.0	-7.4*134.1	-68.4*297.5	55.7*210.6
11:0:0	471600	-42.6*255.9	7.2*305.1	-62.5*63.7	12.2*117.8	-49.6*312.4	28.5*196.1

12:0:0	475200	-43.3*229.0	-5.7*284.8	-47.0*40.5	29.8*95.5	-29.4*322.0	3.2*190.1
13:0:0	478800	-32.1*209.9	-21.9*267.1	-28.0*28.0	33.7*64.4	-7.8*326.0	-17.2*181.4
14:0:0	482400	-13.4*202.3	-34.6*246.8	-10.0*14.8	19.9*42.2	13.6*321.9	-30.5*165.6
15:0:0	486000	9.6*202.5	-38.1*222.2	1.6*356.0	-1.1*35.4	29.9*304.5	-33.2*143.2
16:0:0	489600	35.5*207.3	-29.5*201.4	1.0*333.5	-22.6*38.1	31.2*275.2	-24.4*121.4
17:0:0	493200	63.5*213.9	-11.8*190.0	-11.1*313.7	-42.8*46.9	16.8*250.8	-8.2*103.3
18:0:0	496800	86.8*100.4	11.6*185.9	-28.0*297.2	-61.3*60.6	-2.5*233.8	9.6*84.0
19:0:0	500400	60.1*70.1	39.0*184.3	-43.6*277.9	-78.4*84.1	-19.0*217.0	20.0*58.2
20:0:0	504000	34.7*83.1	67.7*172.5	-50.9*250.1	-83.4*233.9	-27.6*196.0	14.1*32.3
21:0:0	507600	11.5*97.5	72.8*79.7	-44.9*223.6	-67.3*270.9	-24.6*173.6	-4.0*17.7
22:0:0	511200	-10.0*110.8	47.4*63.9	-28.8*211.3	-49.9*289.0	-10.9*155.8	-25.6*13.5
23:0:0	514800	-30.4*120.9	23.0*72.2	-7.8*210.3	-31.0*302.9	9.5*142.1	-47.0*15.4
Time	Tgps	27	28	29	30		
0:0:0	432000	-43.0*223.5	-8.6*276.9	-71.4*16.1	-7.2*60.3		
1:0:0	435600	-28.6*206.6	-24.7*258.7	-86.6*291.0	3.7*39.3		
2:0:0	439200	-7.4*201.5	-35.6*236.7	-70.0*240.4	2.9*16.4		
3:0:0	442800	17.5*203.7	-35.6*211.9	-52.0*251.0	-9.9*358.5		
4:0:0	446400	44.7*210.3	-23.8*193.5	-34.0*265.6	-28.6*347.2		
5:0:0	450000	72.9*218.8	-4.0*184.3	-15.1*280.5	-48.5*338.2		
6:0:0	453600	79.6*60.4	20.8*180.7	5.6*293.9	-66.0*320.2		
7:0:0	457200	54.0*71.5	48.7*176.8	28.9*302.9	-72.0*270.5		
8:0:0	460800	30.4*86.6	74.0*143.9	54.3*299.9	-59.8*239.6		
9:0:0	464400	8.3*101.7	63.4*66.9	69.4*247.7	-42.2*237.9		
10:0:0	468000	-13.1*115.0	37.9*63.4	49.6*201.8	-23.4*246.0		
11:0:0	471600	-33.8*124.4	14.3*73.2	21.9*192.3	-3.6*258.2		
12:0:0	475200	-53.5*125.6	-6.8*86.5	-3.0*186.1	17.7*272.1		
13:0:0	478800	-69.0*103.4	-26.4*100.4	-22.6*175.4	41.5*285.5		
14:0:0	482400	-67.5*50.8	-45.3*112.5	-33.6*156.2	68.5*293.2		
15:0:0	486000	-50.9*27.6	-63.7*118.3	-32.6*131.8	80.1*164.7		
16:0:0	489600	-31.2*17.9	-79.7*89.9	-20.7*110.7	50.6*154.9		
17:0:0	493200	-12.4*7.8	-72.6*10.9	-3.3*93.0	22.5*159.2		
18:0:0	496800	1.2*352.3	-53.2*0.7	12.8*72.7	-2.3*161.1		
19:0:0	500400	4.3*331.4	-32.5*358.0	18.5*47.2	-23.1*157.1		
20:0:0	504000	-4.6*310.9	-12.1*352.5	8.9*25.4	-37.8*143.9		
21:0:0	507600	-20.3*293.4	4.7*339.7	-9.9*14.4	-42.8*120.5		
22:0:0	511200	-36.3*274.9	11.9*318.4	-31.3*11.8	-36.1*96.1		
23:0:0	514800	-45.8*249.4	5.5*295.1	-52.5*14.0	-21.9*76.9		

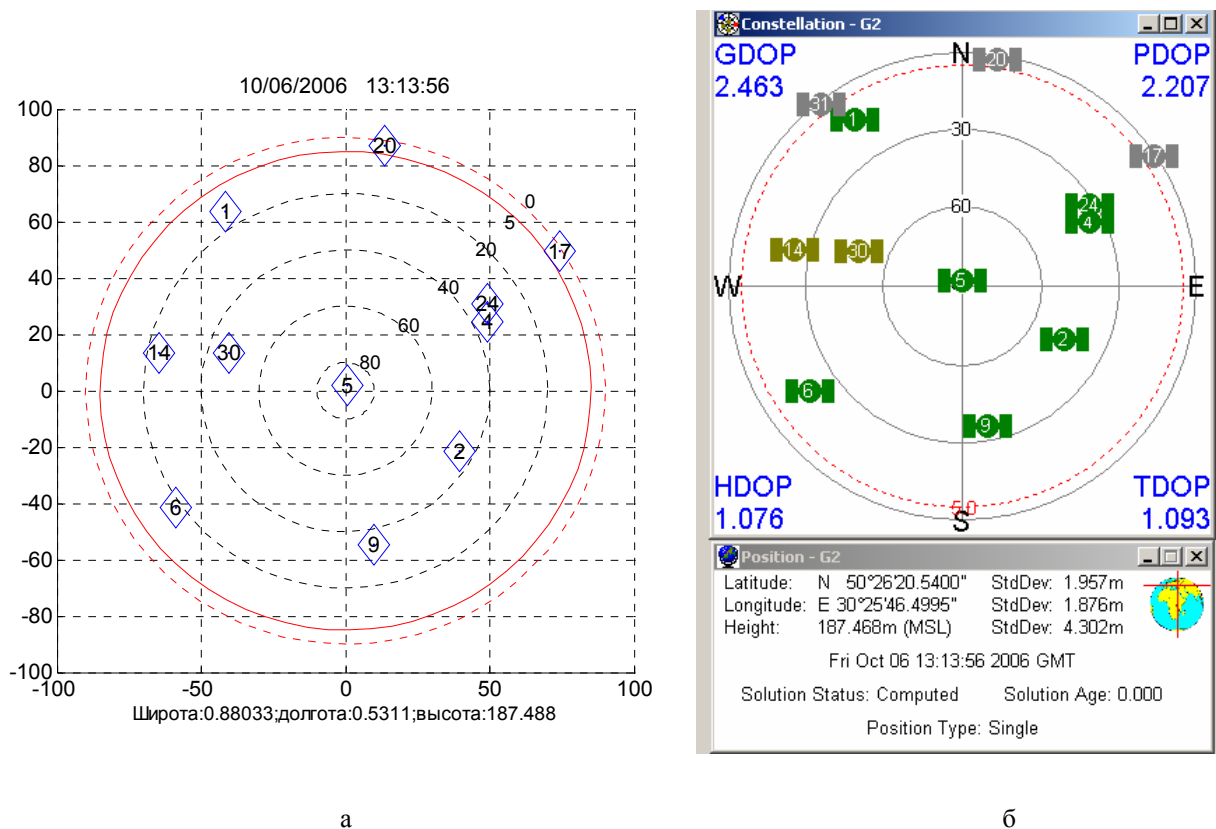


Рис. 4.6. Видимость спутников: а-расчет; б-эксперимент

4.2 Размножения эфемерид спутников ГЛОНАСС (иллюстрация решения системы дифференциальных уравнений)

4.2.1 Краткие сведения из теории

Одним из отличий системы ГЛОНАСС от GPS является то, что размножение эфемерид в системе ГЛОНАСС производится через решение системы дифференциальных уравнений. В полном виде система дифференциальных уравнений для ГЛОНАСС приводится в интерфейсном контрольном документе [], упрощенная принципиальная схема решения дается в книге [1] (параграфы 1. 3. 4, стр. 51- 56, 4. 2.1, стр. 198-201).

Цель лабораторной работы: Овладение методом решения дифференциальных уравнения для размножения эфемерид спутниковой системы ГЛОНАСС.

4.2.2 Лабораторная работа 4. 3 «Решения системы дифференциальных уравнений»

Приведенная программа иллюстрирует применение функции MatLab [7, 8] при решении системы дифференциальных уравнений методом Рунге- Кутта для размножения эфемерид спутников ГЛОНАСС. На CD- диске программа расположена в папке **ОРБИТА_GLONASS**. Для выполнения работы в качестве входных данных потребуются координаты и скорости спутников ГЛОНАСС, которые являются начальными условиями при

решении системы дифференциальных уравнений; время, на которое заданы эти параметры в данных, передаваемых со спутников, время на которое рассчитываются координаты и скорости навигационных спутников. Поскольку приводится существенно упрощенный алгоритм размножения эфемерид, то другие составляющие данных с навигационного спутника не учитываются.

Рекомендуется следующий порядок выполнения лабораторной работы.

1. Создайте папку **ОРБИТА_GLONASS_My** и скопируйте в нее из папки **ОРБИТА_GLONASS** m-файл ORBITA_1.m и функцию orbit_GL.
2. Изучите программные процедуры и комментарии к m-файлу ORBITA_1.m и функции orbit_GL. Выполните **задание 1**.
3. **Задание 1.** С сайта Российского космического агентства или навигационного приемника, работающего с системой ГЛОНАСС, запишите данные координат и скорости работоспособного спутника и включите их в отчет по лабораторной работе.
4. Откройте файл ORBITA_1.m из папки **ОРБИТА_GLONASS_My** и выполните его. Ознакомьтесь с полученным графическим изображением орбиты спутника ГЛОНАСС и выполните **задания 2, 3, 4**.
5. **Задание 2.** Введите координаты и скорости спутника ГЛОНАСС из п. 3 в строки входных данных файла ORBITA_1.m, исполните файл. Проанализируйте полученное графическое изображение орбиты спутника ГЛОНАСС. Результаты анализа внесите в отчет.
6. **Задание 3.** Дополните файл ORBITA_1.m процедурой вывода координат и скорости спутника на одно из значений текущего времени. Исполните файл и запишите полученные значения координат и скоростей спутника в отчет.
7. **Задание 4.** Измените в п. 6 текущее время на 900 секунд (граница интервала размножения эфемерид), исполните файл занесите результат выполнения файла в отчет. Проанализируйте и объясните разницу в результатах, полученных в п. п. 6 и 7.

4.2.3 Вопросы и задания для самоподготовки

1. Для каких целей требуется размножать координаты и скорости навигационных спутников?
2. Какие параметры являются начальными условиями при решении системы дифференциальных уравнений орбитального движения спутников ГЛОНАСС ?
3. В какой системе координат передаются данные о координатах и скорости в спутниковой системе ГЛОНАСС?

4. В какой системе координат решаются дифференциальные уравнения орбитального движения спутников ГЛОНАСС?

4.2.4 Файл ORBITA_1.m

```
%Имя m-файла:ORBITA_1.m
%Программа иллюстрирует процедуру размножения эфемерид и орбиты спутника ГЛОНАСС
%(демонстрация упрощенного варианта решения системы дифференциальных уравнений
%движения спутника)
%Программа выполняется совместно с функцией orbit_GL, использующей функцию MatLab ode45
%для решения дифференциальных уравнений методом Рунге-Кутта
%Входные данные:
%вектор координат x, y, z спутника XYZ (размерность-метр);
%вектор скоростей спутника по осям x, y, z (размерность-м/с) VXYZ;
%текущее время t= "начальное время" : "шаг" : "конечное время=" время в часах"*3600"
%Выходные данные:
%координаты спутника X, Y,Z (x, y, z) в абсолютной (относительной) системах координат;
%скорости спутника Vx, Vy, Vz в абсолютной системе координат;
%вектор текущего времени T;
%вектор текущих координат и скоростей V
%Расчет вектора входных параметров u
omega = 0.7292115*10^(-4);%- скорость вращения Земли
t=0:360:23*3600;
S=omega*3*3600;% угол

%XYZ=[21840.10466;-9006.95351;-9696.59786];%координаты спутника
XYZ=[9795803.22265 ;-7174949.70703;22480344.23828 ];%координаты спутника
mS=[cos(S) -sin(S) 0;sin(S) cos(S) 0;0 0 1]; %матрица преобразования координат

%VXYZ=[-1.19933288;0.58113958;-3.25131421];%скорости спутника
VXYZ=[2773.857116;1295.602798;-814.5313262];
ys1=[mS*XYZ]';%вектор преобразованных координат
ys2=[mS*VXYZ]'+omega*[-ys1(2) ys1(1) 0];% вектор преобразованных скоростей
y=[ys1 ys2];%вектор начальных условий
%Расчет орбиты спутника с помощью функции ode45
%[T,V] = ode45(@orbit_GL,[0:360:23*3600],[y],[]);
[T,V] = ode45(@orbit_GL,[t],[y],[]);
% Координаты и график орбиты спутника

X=V(:,1);
Y=V(:,2);
Z=V(:,3);
```

```

subplot(2,1,1), plot3(X,Y,Z),grid on
set(get(gcf,'CurrentAxes'),'FontSize',14,'FontName','TimesNewRoman');
%Vx=V(:,4);
%Vy=V(:,5);
%Vz=V(:,6) ;
%subplot(1,3,2), plot3(Vx,Vy,Vz)

% Координаты и график орбиты спутника в системе координат ПЗ90
S=omega*T;
x= X.*cos(S)+Y.*sin(S);
y =-X.*sin(S)+Y.*cos(S);
z =Z;
subplot(2,1,2), plot3(x,y,z),grid on
set(get(gcf,'CurrentAxes'),'FontSize',14,'FontName','TimesNewRoman');

function [dy1 y1]= orbit_GL(t,y1)
%Имя функции: orbit_GL
%Функция записи системы дифференциальных уравнений для решения с помощью стандартной
% программы MatLab
dy1 = zeros(6,1);
prom=398600.44*10^9/((y1(1)*y1(1)+y1(2)*y1(2)+y1(3)*y1(3))^1.5);
dy1=[y1(4) y1(5) y1(6) [-y1(1) -y1(2) -y1(3)]*prom]';

```

4.3 Орбитальное движение спутников ГЛОНАСС

4.3.1 Краткие сведения из теории

Приведенный ниже комплекс программ для изучения орбитального движения спутников ГЛОНАСС составлен на основе формул для расчета орбит спутников по данным альманах ГЛОНАСС в полном соответствии с интерфейсным контрольным документом ГЛОНАСС [2]. На CD-диске комплекс размещен в папке 09_ORBITA_GL_NAVIOR. Структура комплекса изображена на рис. 4.7. Экспериментальные данные, использованные при изучении, получены с помощью навигационного приемника «НАВИОР- 14», разработанного Государственным предприятием «Оризон- Навигация» (Украина).

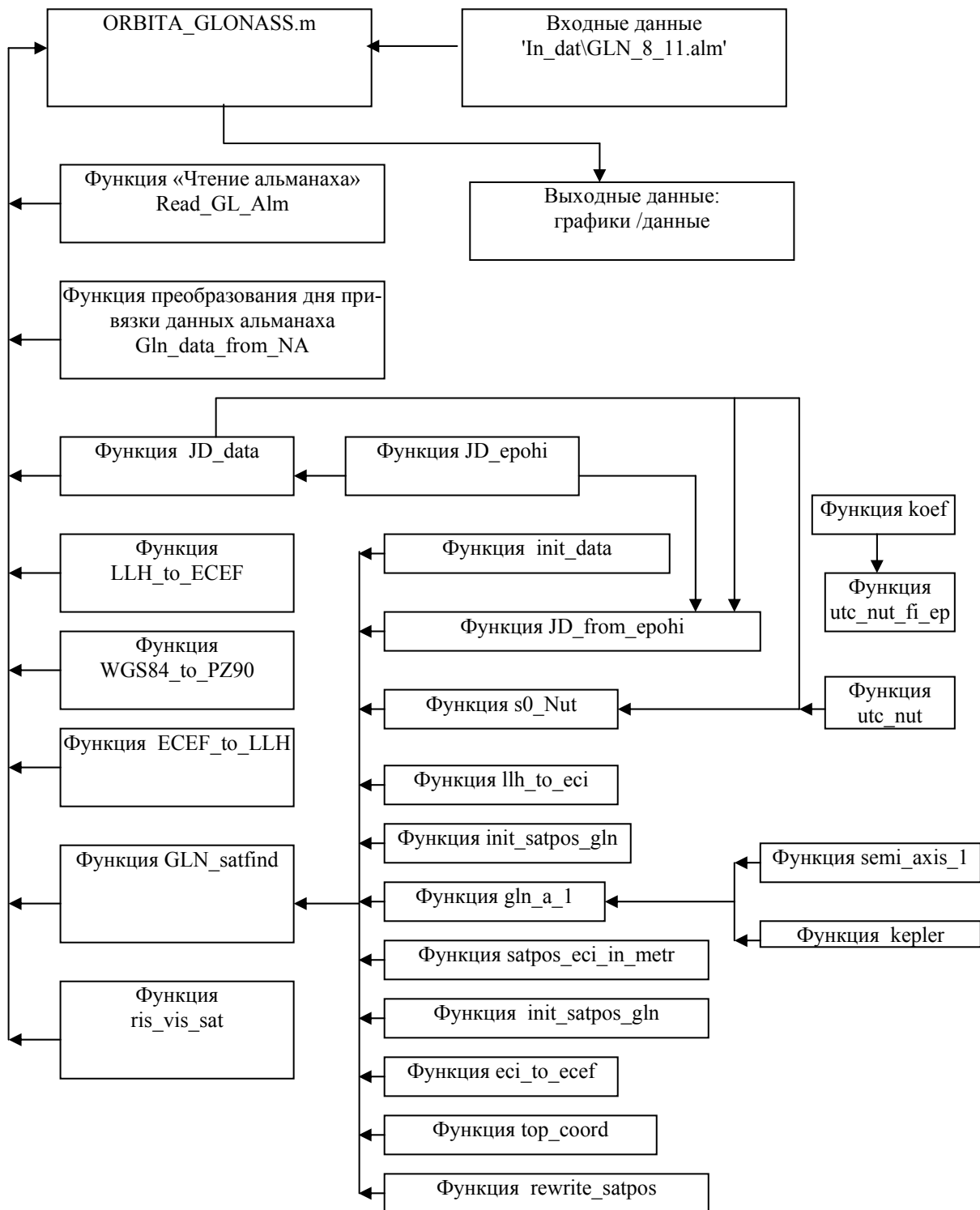


Рис. 4.7. Структура комплекса программ орбитального движения спутников ГЛОНАСС

Цель лабораторной работы: исследование орбитального движения спутников ГЛОНАСС по данным альманаха

4.3.2 Лабораторная работа 4. 4 «Орбитальное движение спутников ГЛОНАСС»

Рекомендуется следующий порядок выполнения лабораторной работы.

1. Создайте папку **ORBITA_GL_NAVIOR_My** и скопируйте в ее из папки **ORBITA_GL_NAVIOR** m-файлы и функции
2. Изучите функции и файл, используя рекомендованную литературу, комментарии к программам и блок –схему, изображенную на рис. 4.7.
3. Откройте m-файл ORBITA_1.m и выполните **задания 1- 3**, руководствуясь комментариями, приведенными в файле.
4. **Задание 1.** Постройте график орбит спутников ГЛОНАСС системе ECEF, проанализируйте их и результаты занесите в отчет.
5. **Задание 2.** Постройте график орбит спутников ГЛОНАСС системе ECI, проанализируйте их и результаты занесите в отчет.
6. **Задание 3.** Постройте графики времени наблюдения спутников ГЛОНАСС , проанализируйте их и результаты занесите в отчет.

4.3.3 Вопросы и задания для самоподготовки

1. Объясните, какой смысл вкладывается в содержание составляющих альманаха ГЛОНАСС: поправка к шкале времени ГЛОНАСС относительно UTC(SU), номер четырех-летнего периода, поправка на расхождение системных шкал времени GPS и ГЛОНАСС, календарный номер суток внутри четырехлетнего периода, номер спутника, номер несущей частоты, долгота восходящего узла орбиты спутника , время прохождения восходящего узла орбиты спутника, поправка к среднему значению наклона орбиты спутника, поправка к среднему значению драконического периода обращения спутника, скорость изменения драконического периода обращения спутника, эксцентриситет орбиты спутника, аргумент перигея орбиты спутника, признак состояния спутника.
2. Какая размерность данных передаваемых со спутника ГЛОНАСС в альманахе?
3. В какой системе координат передаются данные со спутника ГЛОНАСС.

4.3.4 Функции и файлы из папки **ORBITA_GL_NAVIOR**

Файл ORBITA_GLONASS.m

clear all

%Имя файла:ORBITA_GLONASS.m

%Программа рассчитывает орбиты спутников ГЛОНАСС по данным альманаха приемника НАВИОР

14

% разработанного ГП ОРИЗОН-НАВИГАЦИЯ в строгом соответствии с интерфейсным контрольным

%документом ГЛОНАСС

% Входные данные:

```

%Dat - альманах ГЛОНАСС находится в папке In_dat, например, Dat = 'In_dat\GLN_8_11.alm';
%Data_observ.year - год ;
%Data_observ.mon - месяц ;
%Data_observ.day - день;
%current_loc_wgs - координаты точки, из которой проводится наблюдение спутников (pos. receiver
(WGS-84):
% current_loc_wgs.lat - широта, радиан ( DEG_TO_RAD * (50. + 29.0 / 60.0 + 36.78 / 3600.0),например,
% 50 градусов; 29 минут; 36.78 секунд;
% current_loc_wgs.lon - долгота, радиан ( DEG_TO_RAD * (30. + 27.0 / 60.0 + 50.5 / 3600.0),например,
%30 градусов; 27 минут; 50.5 секунд;
%current_loc_wgs.h - высота, метр, например, 120.9;все в WGS 84
%step_t - шаг в секундах (600= 10 минут);
%L =24*1 * 3600- расчетный интервал, например, 24 часа, 1 день, 3600 секунд
%kol - количество спутников, для которых строятся графики орбитального движения
%пом_ns(1:kol) - номера спутников, для которых строятся графики орбитального движения (записыва-
ются
% в квадратных скобках, через пробел, число номеров должно совпадать с количеством спутников
%
%Выходные данные:
%x(j,i)= satpos_gln_ecef(i).x - координата спутника x с номером i на момент времени j в системе
ECEF;
% y(j,i)= satpos_gln_ecef(i).y- координата спутника y с номером i на момент времени j в системе
ECEF;
%z(j,i)= satpos_gln_ecef(i).z- координата спутника z с номером i на момент времени j в системе ECEF;
%x1(j,i)= satpos_gln_a(i).x- координата спутника x с номером i на момент времени j в системе ECI;
%y1(j,i)= satpos_gln_a(i).y- координата спутника y с номером i на момент времени j в системе ECI;
%z1(j,i)= satpos_gln_a(i).z- координата спутника z с номером i на момент времени j в системе ECI;
%x2(j,i)= satpos_gln_ecef(i).vx - скорость спутника с номером i на момент времени j по оси x;
%y2(j,i)= satpos_gln_ecef(i).vy- скорость спутника с номером i на момент времени j по оси y;
%z2(j,i)= satpos_gln_ecef(i).vz- скорость спутника с номером i на момент времени j по оси z;
%x3(j,i)= satvis_gln(i).el*180/pi - угол места спутника с номером i на момент времени j;
%y3(j,i)= satvis_gln(i).az- угол азимута спутника с номером i на момент времени j;
%z3(j,i)= satvis_gln(i).r - дальность до спутника с номером i на момент времени j;
%plot3(x(:,prn),y(:,prn),z(:,prn),S,'LineWidth',0.5)- график орбит спутников ГЛОНАСС системе ECEF;
%plot3(x1(:,prn),y1(:,prn),z1(:,prn),S, 'LineWidth',1)- график орбит спутников ГЛОНАСС системе ECI;
%gis_vis_sat- график видимости спутников
%Примечание: места ввода данных отмечены строками %%%%%%%%%%
%Ввод входных данных
Dat = 'In_dat\GLN_8_11.alm';%%%%%%%%%

```

N=6378136;% радиус Земли (используется, как нормирующий коэффициент
map(N);%функция выводит на графики Землю

```

%Задание цветов для графики
j_color = 0;
color6(1:11) = ['B' 'r' 'D' 'c' 'g' 'k' 'm' '!' 's' 'H' '+'];
%color6(1:16) = ['!' 'k' '!' 'r' 'g' 'r' 'c' 'm' 'r' '!' 'g' '!' 'b' '!' 'k' 'h'];

[alm,max_kol] = Read_GL_Alm(Dat); % Чтение альманаха

nom = 1;
day_from_leap = alm(nom).Na;
while (alm(nom).Health > 0 )
    day_from_leap = alm(nom).Na; % номер дня от ближайшего предшествующего високосного года
    nom = nom + 1;
end;
leap_year = 2004;% високосный год
%alm.Na -(сек) время привязки альманаха от начала предшествующего високосного года
% Время привязки альманаха:
timeUTC = Gln_data_from_NA(leap_year, day_from_leap);
[JD_alm, day_year_alm] = JD_data(timeUTC);
%JD_alm - номер юлианского дня привязки альманаха

% Data_observ - Дата начала обсервации (расчета):
Data_observ = timeUTC;
%{
Data_observ.year = timeUTC.year ;
Data_observ.mon = timeUTC.mon ;
Data_observ.day = timeUTC.day + 2 ;
%}
Data_observ.year = 2006;%%%%%%%%%%%
Data_observ.mon = 11; %%%%%%%%%%%
Data_observ.day =9; %%%%%%%%%%%
Data_observ.ti = 0 ; %%%%%%%%%%%
[JD_observ, day_year_observ] = JD_data(Data_observ);
% JD_observ - номер юлианского дня обсервации
day_observ_alm = JD_observ - JD_alm;
%вывод в командное окно даты, на которую выполняется расчет параметров (год, месяц, день)
fprintf('year=%i mon = %i day = %i \n',Data_observ.year, Data_observ.mon, Data_observ.day);
%Расчет количества и номеров здоровых спутников по данным альманаха:
nom = 1;
i = 0;
k = 0;
i = 0;

```

```

id = alm(nom).ID;
while ( i < 24)
    id = alm(nom).ID;
    Health = alm(nom).Health;
    fprintf('1: i=%i k=%i nom=%i id=%i Health=%i \n', i, k, nom, id, Health);
    if ( id > 0)
        Health = alm(nom).Health;
        if ( Health == 0)
            k = k + 1;
            nom_ns(k) = id;
        % fprintf('2: i=%i k=%i nom=%i id=%i Health=%i \n', i, k, nom, id, Health);
        nom = nom + 1;
    else
        nom = nom + 1;
    end;
else
    nom = nom + 1;
end;
i = i + 1;
end; % i
kol = k;
% fprintf('kol=%i \n', kol); % для вывода номеров здоровых спутников перед fprintf убрать "%"
nom_ns % - номера здоровых навигационных спутников,
%количество спутников, для которых строятся графики орбитального движения
kol = 1; %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
nom_ns(1:kol)=[24]; %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%nom_ns(1:kol)=[1 2 3 4 7 8 19 21 22 23 24 ]; %номера спутников (число номеров должно
%равняться количеству спутников
KOL_GLN = 24;
DEG_TO_RAD = 0.017453292519943; % (PI / 180.00 )
RAD_TO_DEG = 57.295779513082; % (180.00 / PI )
A_WGS84_M = 6378137.0 ; % WGS-84 ellipsoid parameters
B_WGS84_M = 6356752.314; % WGS-84 ellipsoid parameters

A_PZ90_M = 6378136.0; %6 378 136 м - Equatorial radius of the Earth - ,большая полуось эллипсоида
B_PZ90_M = 6356751.36174571344; %AP_LAND (m) Polar radius of the Earth
%A_PZ90_KM = 6378.136; % (Km) Equatorial radius of the Earth
B_PZ90_KM = 6356.75136174571344; % AP_LAND (Km) Polar radius of the Earth
FACTOR_PZ90 = 1.0/298.257839303; % Коэффициент сжатия эллипсоида
C_LIGHT_M = 2.99792458E8; % m/sec Speed of light
RAD_IN_SEC = 7.2722052166430e-5 ; % (PI/43200.0) Number radian in second of time
2*PI/(24*3600)=PI/43200;

```



```

dt_lsf = 14;
% Вводим координаты приемника:
% current_loc_wgs - Pos. reciver (WGS-84):
    current_loc_wgs.lat = DEG_TO_RAD * (50. + 29.0 / 60.0 + 36.78 /
3600.0);%%%%%%%%%%
    current_loc_wgs.lon = DEG_TO_RAD * (30. + 27.0 / 60.0 + 50.5 /
3600.0);%%%%%%%%%%
    current_loc_wgs.h = 120.9; % метр, WGS 84 %%%%%%%%%%
%current_loc_wgs.hae = 0.1229; % Km PZ90
%Преобразование координат
rec_pos_xyz_wgs = LLH_to_ECEF( A_WGS84_M, B_WGS84_M, current_loc_wgs);
rec_pos_xyz_pz90_m = WGS84_to_PZ90(rec_pos_xyz_wgs);
current_loc_pz90 = ECEF_to_LLH(A_PZ90_M, B_PZ90_M, rec_pos_xyz_pz90_m);

step_t = 600;%%%%%%%%%%
%L = 12*24;
L =24*1 * 3600;%%%%%%%%%%
%ti_start = 0;
ti_start = Data_observ.ti + day_observ_alm * 86400;
ti_end = ti_start + L;
j = 0;
for ( ti = ti_start : step_t : ti_end )
    j = j + 1;
%ПЕРЕСЧЕТ ВРЕМЕНИ из секунд в часы, минуты и секунды:
    t(j) = ti;
    A(j) = mod(t(j), 86400);
    hour(j) = floor(A(j) / 3600);
    m(j) = floor((A(j) - hour(j)*3600) / 60);
    sek(j) = A(j) - hour(j) * 3600 - m(j) * 60;

    for (n=1:24) % обнуление массивов в текущий момент времени для всех спутников
        x(j, n) = 0;
        y(j, n) = 0;
        z(j, n)= 0;
        ris_vid(n, j) = 0;
    end;

%timeUTC.ti = mod( ti, 86400); % текущее время обсервации от начала суток
timeUTC.ti = ti;

```

```

[kol_gln_a, satvis_gln, satpos_gln_a, satpos_gln_ecef] = GLN_satfind( A_PZ90_M, B_PZ90_M, timeUTC,
current_loc_pz90, alm);

for (i=1: KOL_GLN)
    if ( satvis_gln(i).el >= 0)
        el = RAD_TO_DEG * satvis_gln(i).el;
        az = RAD_TO_DEG * satvis_gln(i).az;
        AZ(j,i) = az;%уголы азимута спутников в градусах
        EL(j,i) = el;%уголы видимости спутников в градусах
        ris_vid(i, j) = i;
    % fprintf('i=%2i el=%6.2f az=%7.2f x=%12.2f y=%12.2f z=%12.2f\n', i, el, az, satpos_gln_a(i).x, sat-
    pos_gln_a(i).y, satpos_gln_a(i).z);
    else
        AZ(j,i) = 0.0;
        EL(j,i) = 0.0;
    end;
end;

for (i=1:KOL_GLN)
    prn = alm(i).ID;
    health = alm(i).Health;
    if ( (prn > 0) & (health == 0))
    %Координаты спутников в системе ECEF. Для визуализации добавить в вывод графиков и
    % снять блокировку.
        x(j,i)= satpos_gln_ecef(i).x;
        y(j,i)= satpos_gln_ecef(i).y;
        z(j,i)= satpos_gln_ecef(i).z;
    %Координаты спутников в системе ECI. Для визуализации добавить в вывод графиков.
        x1(j,i)= satpos_gln_a(i).x;
        y1(j,i)= satpos_gln_a(i).y;
        z1(j,i)= satpos_gln_a(i).z;
    %Скорости спутников. Для визуализации добавить в вывод графиков.
        x2(j,i)= satpos_gln_ecef(i).vx;
        y2(j,i)= satpos_gln_ecef(i).vy;
        z2(j,i)= satpos_gln_ecef(i).vz;
    %Углы видимости и дальности до спутников. Для визуализации добавить в вывод графиков.
        x3(j,i)= satvis_gln(i).el*180/pi;
        y3(j,i)= satvis_gln(i).az;
        z3(j,i)= satvis_gln(i).r;
    end;
end;
end; % ti (j)

```

```

max_n = 24;
%для блокировки вывода графика "Время наблюдения спутников ГЛОНАСС" перед функцией
%ris_vis_sat в следующей строке поставить %

ris_vis_sat(max_n, j, ti_start, step_t, ris_vid);

for (i=1:kol)
j_color = j_color + 1;
if (j_color > 14 )
    j_color = 1;
    end
    S = color6(j_color);
    prn = nom_ns(i);

hold on
h_F1 = gca;
%{
%ГРАФИКИ ПАРАМЕТРОВ ОРБИТАЛЬНОГО ДВИЖЕНИЯ СПУТНИКОВ ГЛОНАСС
%График орбит спутников ГЛОНАСС системе ECEF
plot3(x(:,prn),y(:,prn),z(:,prn),S,'LineWidth',0.5);
axis([-2.552*10^(7) 2.552*10^(7) -2.552*10^(7) 2.552*10^(7) -2.552*10^(7) 2.552*10^(7)]);
set(get(gcf,'CurrentAxes'),'FontSize',14,'FontName','TimesNewRoman');
set(h_F1,'Position',[0.1 0.1 0.85 0.9]) ;
xlabel('Координата X')
ylabel('Координата Y'),
zlabel ('Координата Z'),grid on
str1 = num2str( prn);
text(x(j,prn),
y(j,prn),z(j,prn),str1,'FontSize',14,'FontName','TimesNewRoman','HorizontalAlignment','center' );
%}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%{
%График орбит спутников ГЛОНАСС системе ECI
plot3(x1(:,prn),y1(:,prn),z1(:,prn),S, 'LineWidth',1);
axis([-2.552*10^(7) 2.552*10^(7) -2.552*10^(7) 2.552*10^(7) -2.552*10^(7) 2.552*10^(7)]);
set(get(gcf,'CurrentAxes'),'FontSize',14,'FontName','TimesNewRoman');
set(h_F1,'Position',[0.1 0.1 0.85 0.9]) ;
xlabel('Координата X')
ylabel('Координата Y'),
zlabel('Координата Z'),grid on
str1 = num2str( prn);

```

```

text(x1(1,prn), y1(1,prn),z1(1,prn),str1,'FontSize',14,'FontName','TimesNewRoman','VerticalAlignment',
'cap');
%}
end

```

Функция Read_GL_Alm

```
function [alm,max_kol] = Read_GL_Alm(Dat);
```

```
%Имя функции: Read_GL_Alm
```

```
%Функция читает данные альманаха спутников ГЛОНАСС.
```

```
%Входные данные:
```

```
%Dat - имя файла, содержащего альманах спутников ГЛОНАСС , например, Dat = 'GLN_all.alm';
```

```
%Выходные данные:
```

```
%alm - данные альманаха спутников ГЛОНАСС (структура) ,
```

```
%max_kol- максимальное количество спутников в альманахе
```

```
fid =fopen(Dat,'rt');%открытие файла
```

```
max_kol = 0;
```

```
%чтение данных из файла
```

```
while notfeof(fid)
```

```
    s1=fscanf(fid,'%s',6);
```

```
    if notfeof(fid)
```

```
        lenstr = length(s1);
```

```
        max_kol = max_kol + 1;
```

```
    % while (fscanf(fid,'%s',1) == '-') end
```

```
        str1 = fscanf(fid,'%s',1);
```

```
% lenstr = length(str1);
```

```
str2 = fscanf(fid,'%s',1);
```

```
lenstr = length(str2);
```

```
    n_sv = sscanf(str2,'%d');
```

```
    strID = str2(1:lenstr);
```

```
    ID = sscanf(strID,'%d');
```

```
    alm(ID).ID = ID; %номер спутника (1...24)
```

```
    %%%%%%%%%%
```

```
    t_1=fscanf(fid,'%s',1);
```

```
    alm(ID).Hn=fscanf(fid,'%d'); % номер частоты спутника
```

```
    t_2=fscanf(fid,'%s',1);
```

```
    alm(ID).Health=fscanf(fid,'%d',1);%здоровье (0- спутник здоров)
```

```
    t_3=fscanf(fid,'%s',1);
```

```
    alm(ID).ecc = fscanf(fid,'%g',1);% эксцентриситет орбиты спутника
```

```
    t_4=fscanf(fid,'%s',1);
```

```

alm(ID).Na =fscanf(fid,'%g',1 );% номер суток к которым относятся данные альманаха, отсчиты-
ваемых
%от ближайшего високосного года
t_5=fscanf(fid,'%s',3);
alm(ID).deltai = fscanf(fid,'%g',1); % наклонение орбиты спутника (радианы)
t_6=fscanf(fid,'%s',2);
alm(ID).LambdaN = fscanf(fid,'%g',1); % долгота восходящего узла орбиты спутника (радианы)
while not(fscanf(fid,'%c',1) == ':')
end
    % t_7=fscanf(fid,'%s',1)
alm(ID).TLambdaN = fscanf(fid,'%g',1) - 10800;% время прохождения восходящего узла орбиты
спутника,
%к которому относятся данные альманаха, приведенное к времени UTC (секунда)
t_8=fscanf(fid,'%s',2);
alm(ID).omegan = fscanf(fid,'%g',1); %аргумент перигея (радианы)
    t_11=fscanf(fid,'%s',1);
alm(ID).Tdr=fscanf(fid,'%g',1); % драконический период (секунда/виток)
t_12=fscanf(fid,'%s',1);
alm(ID).dTdr=fscanf(fid,'%g',1); % скорость изменения драконического периода (секунда/виток в
квдрате)
t_13=fscanf(fid,'%s',2);
alm(ID).tau_n=fscanf(fid,'%g',1) / 1000.0;%коэффициент коррекции шкалы времени (сдвиг времени
спутника
%относительно системного времени ГЛОНАСС в секундах)

end;
end
fclose(fid);

```

Функция Gln_data_from_NA

```
function [time.UTC] = Gln_data_from_NA(Leap_year, day_from_Leap);
```

```
%Имя функции:Gln_data_from_NA
```

```
%Функция предназначена для преобразования номера дня NA (день привязки альманаха
% от ближайшего високосного года) в текущую дату
```

```
%function [time.UTC] = Gln_data_from_NA(Leap_year, day_from_Leap);
```

```
%Входные данные:
```

```
% Leap_year - ближайший високосный год
```

```
%day_from_Leap (NA) - день привязки альманаха
```

```
%Выходные данные:
```

```
%структура timeUTC (год, месяц, день) - текущая дата
```

```
DnMon = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]; %количество дней в месяцах
```

```

n4 = mod(leap_year, 4);
n100 = mod(leap_year, 100);
n400 = mod(leap_year, 400);
if (n4 == 0)
    n_leap = 1;
else
    n_leap = 0;
end
if ((n100 == 0) & (n400 > 0))
    n_leap = 0;
end
if (day_from_leap > (365 + n_leap))
    day = day_from_leap - (365 + n_leap);
    k = fix (day / 365);
    day = mod(day, 365);
else
    day = day_from_leap;
    k = -1;
end;
god = leap_year + k + 1;
if (god > leap_year)
    n_leap = 0;
end;
mon = 1;
mon_day = 31;
while (day > mon_day)
    day = day - mon_day;
    mon = mon + 1; % LK
    mon_day = DnMon(mon);
%   mon = mon+1;
    if (mon == 2)
        mon_day = mon_day + n_leap;
    end
end
time.UTC.year = god;
time.UTC.mon = mon;
time.UTC.day = day;

```

Функция JD_data

```
function [JD, day_year] = JD_data(timeUTC)
```

```
%Имя:JD_data
```

```
% Функция JD_data(timeUTC) вычисляет :
```

%JD - номер юлианского дня, day_year - номер дня года.

%Входные данные:

%Структура timeUTC

%timeUTC.year - год,

% timeUTC.mon - месяц,

% timeUTC.day - день.

%Выходные данные:

%JD - юлианский день;

%day_year- день от начала года.

%количество дней в месяцах

DnMon = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31];

%Вычисление номера юлианского дня опорной эпохи

jd0 = JD_эпохи(timeUTC.year);

%Учет високосного года

nfebr = 0;

if mod(timeUTC.year,4) == 0

 nfebr = 1;

end;

%Расчетномера дня года

k = 0;

for i = 2 : timeUTC.mon

 k = k + DnMon(i - 1);

 if (i == 2)

 k = k + nfebr;

 end;

end;

day_year = k + timeUTC.day;

%Расчет номера юлианского дня

JD = jd0 + day_year;

Функция LLH_to_ECEF

function [XYZ] = LLH_to_ECEF(a, b, llh)

%Имя функции:LLH_to_ECEF.m

%Функция преобразования географических координат в прямоугольную геоцентрическую систему координат (ECEF)

%Входные данные:

%Структура llh

%llh.lon-долгота (радиан),

%llh.lat-широта (радиан),

%llh.h-высота (метр);

%a, b-большая и малая полуоси земного эллипсоида в WGS-84 (метр);

%Выходные данные:

```

%Структура XYZ
%XYZ.x - координата x в ECEF;
%XYZ.y - координата y в ECEF;
%XYZ.z - координата z в ECEF;
a2=a * a;
b2=b * b;
r = a2 / sqrt(a2 * cos(llh.lat) * cos(llh.lat) + b2 * sin(llh.lat) * sin(llh.lat));
XYZ.x = (r + llh.h) * cos(llh.lat) * cos(llh.lon);
XYZ.y = (r + llh.h) * cos(llh.lat) * sin(llh.lon);
XYZ.z = (b2 / a2 * r + llh.h) * sin(llh.lat);

```

Функция WGS84_to_PZ90

```

function [pos_pz90] = WGS84_to_PZ90( pos_wgs84)
%Имя функции: WGS84_to_PZ90.m
%Функция преобразует координаты из системы WGS 84 в систему ПЗ 90
p = (1.0 - 0.12e-6);
x = pos_wgs84.x;
y = pos_wgs84.y;
z = pos_wgs84.z;
pos_pz90.x = (p * x + p * 0.82e-6 * y + 1.4 );
pos_pz90.y = (-p * 0.82e-6 * x + p * y + 1.4);
pos_pz90.z = (p * z + 0.9);

```

Функция ECEF_to_LLH

```

function [llh] = ECEF_to_LLH(a, b, XYZ)
%Имя функции:ECEF_to_LLH
%Функция преобразует координаты из системы ECEF в географическую систему
%Входные данные:
%a, b-большая и малая полуоси земного эллипсоида (метр); ( a=6378137.0; %b=6356752.314; метры)
%Структура XYZ
%XYZ.x - координата x в ECEF;
%XYZ.y - координата y в ECEF;
%XYZ.z- координата z в ECEF;
%Выходные данные:
%Структура llh
%llh.lon - долгота (радиан);
%llh.lat - широта (радиан);
%llh.h-высота (метр);
a2 = a * a;
b2 = b * b;
xy = sqrt(XYZ.x * XYZ.x + XYZ.y * XYZ.y);
thet = atan2(XYZ.z * a , (xy * b));

```



```

%thet = atan(XYZ.z * a / (xy * b));
esq = 1.0 - b2 / a2;
epsq = a2 / b2 - 1.0;
llh.lat = atan((XYZ.z + epsq * b * (sin(thet) ^3)) / (xy - esq * a * (cos(thet) ^3)));
llh.lon = atan2(XYZ.y, XYZ.x);%!
while (llh.lon < 0)
    llh.lon = 2*pi + llh.lon;
end;
%if llh.lon < 0
% llh.lon = 2*pi + llh.lon;
%end ;
r = a2 / sqrt(a2 * cos(llh.lat) * cos(llh.lat) + b2 * sin(llh.lat) * sin(llh.lat));
llh.h = xy / cos(llh.lat) - r;
end

```

Функция GLN_satfind

```

function [kol_gln_a, satvis_gln, satpos_gln_ns, satpos_gln_ecef] = GLN_satfind(a, b, timeUTC, current_loc_pz90, alm_gln);
%Имя функции:GLN_satfind
%Функция вычисляет координаты спутников ГЛОНАСС, углы видимости и количество видимых спутников
%Входные данные:
%a, b- большая и малая полуоси земного эллипсоида;
%timeUTC- параметры времени;
%current_loc_pz90- широта, долгота и высота точки, из которой наблюдаются спутники (структура);
% alm_gln-альманах спутников ГЛОНАСС (структура)
%Выходные данные:
%kol_gln_a- количество видимых спутников ГЛОНАСС,
%satvis_gln- углы видимости спутников (структура),
%satpos_gln_ns-координаты спутников (структура) в абсолютной системе координат (ECI)
%satpos_gln_ecef- координаты спутников (структура) в подвижной системе координат (ECEF)
RAD_TO_DEG = 180.00 / pi ;
KOL_GLN = 24;
nom_ns = 1;
Min_Elev = 0; % минимальный угол видимости
kol_gln_a = 0;
[nom_sat_gln_a, trac_sat_gln_a, sat_pos, satvis_gln] = init_data(KOL_GLN);

% число дней от фундаментальной эпохи 2000г от 12h, 0, January:
% jd_up_epoh = JD_from_epohi( 2000, timeUTC) - 0.5; % from 12h UTC 2000
nut = 0;

```

```

S0 = s0_Nut( timeUTC, nut);
time_s0 = S0.s0_m_mod ; %time_s0 - истинное звездное время в день обсервации, 0h UTC

year = timeUTC.year;
leap_year = fix( year / 4) * 4; % ближайший к текущему (предыдущий) високосный год
ti = timeUTC.ti; % текущее время обсервации от начала дня
n00 = fix(ti / 86400);
% n0 - номер текущих суток внутри 4-х летнего периода (от ближайшего високосного года)
n0 = JD_from_epohi(leap_year, timeUTC) + n00 + 1;
[eci_current_loc, eci_rec_pos_xyz] = llh_to_eci(a, b, ti, time_s0, current_loc_pz90);
satpos_eci = init_satpos_gln();

for ( i = 1 : KOL_GLN)
    prn = alm_gln (i).ID;
    health = alm_gln (i).Health;
    if ( (prn > 0) & (health == 0))
        satpos_eci = gln_a_1(i, n0, ti, time_s0, alm_gln);
        [satpos_eci, satpos_gln_a] = satpos_eci_in_metr(satpos_eci); %Transformation coordinates from
"Km" in "m"
        satpos_gln_ns(prn) = satpos_gln_a;
    %-----
    [satpos_ecef] = eci_to_ecef(time_s0, ti, satpos_eci);
    satpos_gln_ecef(prn) = satpos_ecef;
% calculates angles of visibility by almanac:
    [top] = top_coord(prn, eci_current_loc, eci_rec_pos_xyz, satpos_gln_a);

    satvis_gln(prn) = top;

if (satvis_gln(prn).r > 0.0)
    el = RAD_TO_DEG * satvis_gln(prn).el;
    if ( el >= Min_Elev)
%    fprintf('prn=%2i el=%f az=%f\n ', prn, el, RAD_TO_DEG * satvis_gln(prn).az);
        satpos_gln( prn) = satpos_eci;
        [satpos_ecef] = eci_to_ecef(time_s0, ti, satpos_eci);

        [trac_sat_gln_a] = rewrite_satpos(nom_ns, satpos_ecef);
        trac_sat_gln_a( nom_ns).r = satvis_gln( prn).r;
        trac_sat_gln_a (nom_ns).prn = prn;
        nom_sat_gln_a(nom_ns) = prn;
        nom_ns = nom_ns + 1; % number of visible satellites
    end; % if ( el >= Min_Elev)
end; % if ( satvis_gln[prn-1].r > 0.0)

```

```

end; % f ( (prn > 0) & (health == 1))
end; % for ( i = 1 : KOL_GLN)
kol_gln_a = nom_ns - 1;

```

Функция ris_vis_sat

```

function ris_vis_sat(max_n, nj, ti_start, step_t, ris_vid)
%Имя функции:ris_vis_sat
%Функция строит графики спутников, видимых из точки наблюдения
mm = step_t/3600.0;
j_color = 0;
color7(1:7) = ['b' 'g' 'r' 'c' 'm' 'y' 'k'];

for i = 1 : max_n
    str1 = ris_vid(i, 1: nj);
    k = find(str1);
    if k > 0
        len = length(k);
        if len == 1
            hPlot = plot(k(1):k(1),ris_vid(i,k(1):k(1)), s);
            set(hPlot, 'LineWidth', 3);
            hold on;
        else
            n0 = k(1);
            j_color = j_color + 1;
            if (j_color > 7 )
                j_color = 1;
            end
            s = color7(j_color);
            for j = 2 : len
                if (k(j) > (k(j-1) + 1)) | (j == len)
                    k0 = k(j - 1);
                    hPlot = plot(n0*mm : mm : k0*mm,ris_vid(i,n0:k0), s);
                    set(hPlot, 'LineWidth', 3);
                    hold on;
                    n0 = k(j);
                end; % if
            end; % for j = 2 : len
        end; % else
    end; % if k > 0
end; % for i = 1 : max_n

```

```

grid on
hAxes=gca;
My = [0:2:25];
kol_x = ti_start + nj * step_t;

Mx = [0 : 2 : (kol_x - ti_start) / 3600];
set(hAxes, 'ytick', My, 'xtick', Mx) ;
hText = gca;
set(hText, 'FontSize', 12, 'FontName', 'TimesNewRoman')
xlabel('Время наблюдения спутников ГЛОНАСС, час')
ylabel('Номер навигационного спутника')

```

Функция JD_epochi

```

function jden = JD_epochi(epocha)
%Имя: JD_epochi
%Функция JD_epochi(epocha) рассчитывает номер юлианского дня
%до начала года (epocha) на 12h, 0 день, январь.
%Входные данные: epocha, размерность-год
%Выходные данные:
% jden- номер юлианского дня на 12h, 0 день, январь ( размерность -дни)
rk = mod(epocha,4);
if ( rk == 0 ) rk = 1.0;
else
    rk = 2.0 - rk * 0.25;
end;
n100 = floor(epocha / 100);
n400 = floor(epocha / 400);
jden = (4712 + epocha) * 365.25 + n400 - n100 + rk;
% fprintf('epocha=%d rk=%f jden=%6.2f \n', epocha, rk, jden);

```

Функция init_data

```

function[nom_sat_gln_a, trac_sat, sat_pos, satvis_gln] = init_data(kol)
%Имя функции: init_data
%Функция предназначена для инициализации массивов
sat_pos.prn = 0;
sat_pos.x = -1.0;
sat_pos.y = -1.0;
sat_pos.z = -1.0;
sat_pos.vx = -1.0;
sat_pos.vy = -1.0;
sat_pos.vz = -1.0;

```

```

    sat_pos.dvx = -1.0;
    sat_pos.dvy = -1.0;
    sat_pos.dvz = -1.0;
    sat_pos.range_m = -1.0;

for( i = 1 : kol)
    nom_sat_gln_a(i) = 0;
    satvis_gln(i).el = -1; satvis_gln(i).az = -1; satvis_gln(i).r = 0;
    trac_sat(i).prn = 0;
    trac_sat(i).x = -1.0;
    trac_sat(i).y = -1.0;
    trac_sat(i).z = -1.0;
    trac_sat(i).vx = -1.0;
    trac_sat(i).vy = -1.0;
    trac_sat(i).vz = -1.0;
    trac_sat(i).range_m = -1.0;
end;

```

Функция JD_from_epohi

```

function [jd] = JD_from_epohi( epoha, timeUTC);
%Имя функции: JD_from_epohi.m
%Функция вычисляет jd - количество дней от указанного года (epoha)
% до текущей даты, указанной в структуре timeUTC, представленной в виде
% (timeUTC.year, timeUTC.mon, timeUTC.day)
    jd0 = JD_epohi(epoha) + 1; % 12h, 1 den January
    [day, day_year] = JD_data(timeUTC);
    jd = day - jd0;

```

Функция s0_Nut

```

function [S0] = s0_Nut( timeUTC, nut)
%Имя функции: s0_Nut
%Функция рассчитывает истинное или среднее звездное время на 0ч UTC
%Входные данные:
%timeUTC - дата, на которую требуется рассчитать истинное или среднее звездное время
%nut- признак (если nut= 0, то вычисляется среднее звездное время без учета нутации, иначе вычис-
ляется
%истинное звездное время)
%Выходные данные:
%S0 - истинное или среднее звездное время на 0ч UTC
jd2000 = 2451545; % 12h UTC 1 января
% Применить функцию JD_data
[jd, day_year] = JD_data( timeUTC);
if (jd == NaN)

```

```

s0_mod = NaN; h = NaN; min = NaN; sec = NaN;
fprintf('function s0_m - end0 \n');
return;
end;
jd = jd - 0.5;
d = jd - jd2000;
t = d / 36525.0; % 36525 - юлианский период 100 лет
t2 = t * t;
h1 = 24110.54841;
%h1=6.0*3600.0+41.0*60.0+50.54841;
% h2 = 236.555367908 * d;
h2 = 8640184.812866 * t ;
h3 = 0.093104 * t2;
h4 = t2 * t * 6.2E-6;
if ( nut == 0)
    na = 0;
else
    na = utc_nut(t);
end;
s0_m = h1 + h2 + h3 - h4;
S0.s0_nut = s0_m + na;
S0.s0_m_mod = mod(s0_m, 86400);
s0_day = floor(s0_m / 86400);
S0.s0_m_hour = S0.s0_m_mod / 3600.0;
S0.s0_m_hour = floor(S0.s0_m_mod / 3600);
sec_min = S0.s0_m_mod - S0.s0_m_hour * 3600;
S0.s0_m_min = floor(sec_min / 60);
S0.s0_m_sec = sec_min - S0.s0_m_min * 60;
S0.s0_nut_mod = mod(S0.s0_nut, 86400);
s0_day = floor(S0.s0_nut / 86400);
S0.s0_nut_hour = S0.s0_nut_mod / 3600.0;
S0.s0_nut_hour = floor(S0.s0_nut_mod / 3600);
sec_min = S0.s0_nut_mod - S0.s0_nut_hour * 3600;
S0.s0_nut_min = floor(sec_min / 60);
S0.s0_nut_sec = sec_min - S0.s0_nut_min * 60;

```

Функция llh_to_eci

```
function [eci_llh, eci_xyz] = llh_to_eci(a, b, ti, time_s0, llh_loc) ;
```

```
%Имя функции:llh_to_eci
```

```
%Функция вычисляет позицию приемника в абсолютной геоцентрической системе координат (ECI)
```

```
%Входные данные:
```

```
%a, b-большая и малая полуоси земного эллипсоида (метр);
```

```

%ti- текущее время (секунды) ,
%time_s0- истинное звездное время,
%Структура llh_loc - координаты приемника; {
%llh_loc.lon-долгота (радиан);
%llh_loc.lat-широта (радиан);
%llh_loc.h-высота (метр);
%Выходные данные:
%Структура eci_llh - географические координаты приемника в абсолютной геоцентрической системе
координат (ECI)
%eci_llh.lon - долгота (радиан);
%eci_llh.lat - широта (радиан);
%eci_llh.h - высота (метр);
%Структура eci_xyz- координаты приемника в абсолютной прямоугольной геоцентрической системе
координат (ECI)
% eci_xyz.x - координата x;
% eci_xyz.y - координата y;
% eci_xyz.z - координата z;
OMEGA_Z = 0.7292115e-4; % угловая скорость вращения Земли, (рад/сек)
SEC_IN_RAD = 7.2722052166430e-005; % (PI / 43200.0 ) // Number radian
eci_llh.lon = llh_loc.lon + ti * OMEGA_Z + SEC_IN_RAD * time_s0;
eci_llh.lat = llh_loc.lat;
eci_llh.h = llh_loc.h;
eci_xyz = llh_to_ecef( a, b, eci_llh);

```

Функция init_satpos_gln

```
function [sat_pos] = init_satpos_gln() ;
```

```
%Имя функции: init_satpos_gln
```

```
%Функция предназначена для инициализации структуры sat_pos
```

```

sat_pos.pri = 0;
sat_pos.x = -1.0;
sat_pos.y = -1.0;
sat_pos.z = -1.0;
sat_pos.vx = -1.0;
sat_pos.vy = -1.0;
sat_pos.vz = -1.0;
sat_pos.dvx = -1.0;
sat_pos.dvy = -1.0;
sat_pos.dvz = -1.0;
sat_pos.range_m = -1.0;

```

Функция gln_a_1

```
function [satpos_xyz_gln] = gln_a_1(ns, n0, ti_current, time_s0, alm_gln);
```

```

%Имя функции:gln_a_1
%Функция рассчитывает координаты и скорости спутников ГЛОНАСС в соответствии с интерфей-
НЫМ
%контрольным документом ГЛОНАСС
%Входные данные:
%ns- номер спутника ,
%p0 - номер текущих суток внутри 4-х летнего периода (от ближайшего високосного года),
%ti_current - текущее время обсервации от начала дня,
%time_s0 - истинное звездное время в текущий момент обсервации,
%alm_gln - альманах спутников ГЛОНАСС
%Выходные данные:
%Структура satpos_xyz_gln
%satpos_xyz_gln.x - координата по оси x;
%satpos_xyz_gln.y- координата по оси y;
%satpos_xyz_gln.z- координата по оси z;
%satpos_xyz_gln.vx- скорость по оси x;
%satpos_xyz_gln.vy - скорость по оси y;
%satpos_xyz_gln.vz- скорость по оси z ;
% I_MID - Mean value of an inclination of a plane of orbit of a satellite
I_MID = 1.0995574287564; %(double)(PI * 63.0 / 180.0)
T_DR_MID = 43200.0; %Mean value a dragon of cycle time of a satellite
A_PZ90_KM = 6378.136; %(Km) Equatorial radius of the Earth
MU = 398600.44; %(Km^3/cek^2) constant of a gravitational
OMEGA_Z = 0.7292115e-4; %скорость вращения Земли (Angular speed of rotation of the Earth,
рад/cek)
SEC_IN_RAD = 7.2722052166430e-005; %(PI / 43200.0 ) Number radian in second of time
2*PI/(24*3600)=PI/43200
HALF_PI = pi / 2;
D7_3 = 7.0 / 3.0;
D7_4 = 7.0 / 4.0;
D7_6 = 7.0 / 6.0;
D7_24 = 7.0 / 24.0;
D49_72 = 49.0 / 72.0;
nn = fix(ti_current / 86400);
ti = ti_current - nn * 86400;
% i_incl = I_MID + alm_gln(ns).deltai;

i_incl = alm_gln(ns).deltai;

ecc2_1 = 1.0 - alm_gln(ns).ecc * alm_gln(ns).ecc;
% t_dr = T_DR_MID + alm_gln(ns).Tdr;
t_dr = alm_gln(ns).Tdr;

```



```

n_dr = pi * 2 / t_dr;
%1. Вычисление полуоси a_n:
a_n = semi_axis_1(t_dr, i_incl, alm_gln(ns).ecc, alm_gln(ns).omegan);
alm_gln(ns).a = a_n;

%2. Вычисление t_lambda_k - время пересечения восходящего узла орбиты спутника
sin_i = sin(i_incl);
cos_i = cos(i_incl);
sin_i2 = sin_i * sin_i;
cos_i2 = cos_i * cos_i;
v = - alm_gln(ns).omegan; % omega_n - angle of a perigee
J = -0.00162393855 ; % J = 3/2 * C20; C20=1082.6257 * 10-6;
ae_a = A_PZ90_KM / a_n;
ae_a2 = ae_a * ae_a;
j_ae_a2 = J * ae_a2;

omega1 = j_ae_a2 * n_dr * cos_i / (ecc2_1 * ecc2_1);
n0_na = (n0 - alm_gln(ns).Na);
tz = ti - alm_gln(ns).TLambdaN + 86400.0 * n0_na;

wk = tz / t_dr;
wi = fix(wk);
wi2 = wi * wi;
t_lambda_kk = alm_gln(ns).TLambdaN + t_dr * wi + alm_gln(ns).dTdr * wi2;
t_lambda_k = t_lambda_kk - n0_na * 86400.0 ;
lambda_k = alm_gln(ns).LambdaN + (omega1 - OMEGA_Z) * (wi * t_dr + alm_gln(ns).dTdr * wi2) ;
%time_s0 - a true sidereal time to Greenwich midnight of date n0 to which time ti concerns
time_s = time_s0 * SEC_IN_RAD + OMEGA_Z * t_lambda_k ;
omega_0 = lambda_k + time_s;

% Auxiliary values:
d1 = 1.0 - 1.5 * sin_i2;
j_ae_a2_d1 = j_ae_a2 * d1;
j_ae_a2_d = j_ae_a2 * (1.0 - 1.5 * sin_i);
j_ae_a2_sin_i = j_ae_a2 * sin_i;
j_ae_a2_sin_i2 = j_ae_a2 * sin_i2;
j_ae_a2_cos_i2 = j_ae_a2 * cos_i2;

% 3. Calculation of constants:
tau = 0;
l = cos(alm_gln(ns).omegan) * alm_gln(ns).ecc;
h = alm_gln(ns).ecc * sin(alm_gln(ns).omegan);
dop_x = (sqrt(1.0 - alm_gln(ns).ecc) * tan(v / 2.0));
dop_y = sqrt(1.0 + alm_gln(ns).ecc);

```

```

ee = 2.0 * atan2(dop_x, dop_y);
ma = ee - alm_gln(ns).ecc * sin(ee);
ll = ma + alm_gln(ns).omegan;

% for (j = 0; j <= 1; j++)
for (j = 1 : 2)

sin_1 = sin(ll);
sin_2l = sin(2 * ll);
sin_3l = sin(3 * ll);
sin_4l = sin(4 * ll);
cos_1 = cos(ll);
cos_2l = cos(2 * ll);
cos_3l = cos(3 * ll);
cos_4l = cos(4 * ll);
l_cosl = 1 * cos_1;
delta_a(j) = 2.0 * a_n * j_ae_a2_d1 * l_cosl +...
            j_ae_a2 * sin_i * (0.5 * h * sin_1 - 0.5 * l_cosl +...
            cos(2.0 * alm_gln(ns).LambdaN) + 3.5 * l * cos_3l + 3.5 * h * sin_3l);
delta_h(j) = j_ae_a2_d1 * (1 * n_dr * tau + sin_3l +...
            1.5 * l * sin_2l - 1.5 * h * cos_2l) -...
            0.25 * j_ae_a2_sin_i2 * (sin_1 - D7_3 * sin_3l +...
            5.0 * l * sin_2l - 8.5 * l * sin_4l +...
            8.5 * h * cos_4l + h * cos_2l) + j_ae_a2_cos_i2 *...
            (tau * l * n_dr - 0.5 * l * sin_2l);
delta_l(j) = j_ae_a2_d1 * (- tau * h * n_dr + cos_1 +...
            1.5 * l * cos_2l + 1.5 * h * sin_2l) -...
            0.25 * j_ae_a2_sin_i2 * (- cos_1 - D7_3 * cos_3l -...
            5.0 * h * sin_2l - 8.5 * l * cos_4l -...
            8.5 * h * sin_4l + l * cos_2l) + j_ae_a2_cos_i2 *...
            (- tau * h * n_dr + 0.5 * h * sin_2l);
d_omega(j) = j_ae_a2 * cos_i * (tau * n_dr + 3.5 * l * sin_1 -...
            3.5 * h * cos_1 - 0.5 * sin_2l - D7_6 * sin_3l + ...
            D7_6 * h * cos_3l);
delta_i(j) = 0.5 * j_ae_a2_sin_i * cos_i * (- l * cos_1 +...
            h * sin_1 + cos_2l + D7_3 * l * cos_3l + D7_3 * h * sin_3l);
delta_ll(j) = 2.0 * j_ae_a2_d * ... %           2.0 * j_ae_a2_d1 *
            (tau * n_dr + D7_4 * l * sin_1 -...
            D7_4 * h * cos_1) + 3 * j_ae_a2_sin_i *...
            (- D7_24 * h * cos_1 - D7_24 * l * sin_1 -...
            D49_72 * h * cos_3l + D49_72 * l * sin_3l +...
            0.25 * sin_2l) + j_ae_a2 * cos_i *...

```

```

        (tau * n_dr + 3.5 * l * sin_l - 2.5 * h * cos_l - ...
        0.5 * sin_2l - D7_6 * l * sin_3l + D7_6 * h * cos_3l);
tau = ti - t_lambda_k;
ll = ma + alm_gln(ns).omegan + n_dr * tau;
end;
% 4. Corrections to orbit elements of a satellite by second zone harmonic J20
% influence at the moments of time ti
dda = delta_a(2) - delta_a(1);
ddh = delta_h(2) - delta_h(1);
ddl = delta_l(2) - delta_l(1);
dd_omega = d_omega(2) - d_omega(1);
ddi = delta_i(2) - delta_i(1);
dd_ll = delta_ll(2) - delta_ll(1);

% 5. calculation of influenced elements of orbits at the moment of ti time
ai = a_n + dda;
hi = h + ddh;
li = l + ddl;
% ecc_i = alm_gln[ns].ecc;
ecc_i = sqrt(hi * hi + li * li);
if (ecc_i == 0)
    w_i = 0;
else
    if (li == ecc_i)
        w_i = HALF_PI;
    else
        if (li == -ecc_i)
            w_i = -HALF_PI;
        else
            if (li ~= 0)
                w_i = atan2(hi,li);
            else
                w_i = HALF_PI;
            end;
        end;
    end;
end;

omega_i = omega_0 + dd_omega;

ii_incl = i_incl + ddi;
ll_z = ma + alm_gln(ns).omegan + n_dr * tau + dd_ll;

```

```

% ll_z = ma + alm_gln(ns).omegan + n_dr * (ti - t_lambda_k) + dd_ll;
mm_i = ll_z - w_i;

% 6. Coordinates and components of satellite velocity vector
% in absolute coordinate system OXaYaZa at the moment of ti time :
%  $E_i(n) = M_i + e_i * \sin E_i(n-1)$ ,  $E_i(0) = M_i$ ,  $|E_i(n) - E_i(n-1)| \leq 10^{-8}$ ,
eps_kepler = 1E-8;
ee_i = kepler( mm_i, ecc_i, eps_kepler); % ee_i - эксцентрическая аномалия
% alm_mca(ns).ek = ee_i;
v_i = 2.0 * atan2(sqrt(1.0 + ecc_i) * tan(ee_i / 2.0), sqrt(1.0 - ecc_i));
u_i = v_i + w_i;
ri = ai * (1.0 - ecc_i * cos(ee_i));
ecc_i2 = 1.0 - ecc_i * ecc_i;
sqrt_mu_ai = sqrt( MU / ai);
vr_i = sqrt_mu_ai * (ecc_i - sin(v_i)) * ecc_i2;
vu_i = sqrt_mu_ai * (1.0 + ecc_i * cos(v_i)) * ecc_i2;

cos_ui = cos(u_i);
sin_ui = sin(u_i);
cos_ii = cos(ii_incl);
sin_ii = sin(ii_incl);
cos_omega_i = cos(omega_i);
sin_omega_i = sin(omega_i);
x_dop = (cos_ui * cos_omega_i - sin_ui * sin_omega_i * cos_ii);
y_dop = (cos_ui * sin_omega_i + sin_ui * cos_omega_i * cos_ii);

satpos_xyz_gln.x = ri * x_dop;
satpos_xyz_gln.y = ri * y_dop;
satpos_xyz_gln.z = ri * sin_ui * sin_ii;
satpos_xyz_gln.vx = vr_i * x_dop - vu_i * (sin_ui * cos_omega_i + ...
cos_ui * sin_omega_i * cos_ii);
satpos_xyz_gln.vy = vr_i * y_dop - vu_i * (sin_ui * sin_omega_i - ...
cos_ui * cos_omega_i * cos_ii);
satpos_xyz_gln.vz = vr_i * sin_ui * sin_ii + vu_i * cos_ui * sin_ii;

```

Функция satpos_eci_in_metr

```
function [satpos_eci, satpos_gln] = satpos_eci_in_metr(satpos_eci);
```

```
%Имя функции: satpos_eci_in_metr
```

```
%Функция преобразует координаты satpos_eci (структура), заданные в километрах в координаты
```

```
%satpos_eci, satpos_gln в метрах
```

```
satpos_eci.x = satpos_eci.x * 1000.0;
```

```
satpos_eci.y = satpos_eci.y * 1000.0;
```

```

satpos_eci.z = satpos_eci.z * 1000.0;
satpos_eci.vx = satpos_eci.vx * 1000.0;
satpos_eci.vy = satpos_eci.vy * 1000.0;
satpos_eci.vz = satpos_eci.vz * 1000.0;
satpos_gln.x = satpos_eci.x;
satpos_gln.y = satpos_eci.y;
satpos_gln.z = satpos_eci.z;

```

Функция init_satpos_gln

```
function [sat_pos] = init_satpos_gln() ;
```

```
%Имя функции: init_satpos_gln
```

```
%Функция предназначена для инициализации структуры sat_pos
```

```

sat_pos.prn = 0;
sat_pos.x = -1.0;
sat_pos.y = -1.0;
sat_pos.z = -1.0;
sat_pos.vx = -1.0;
sat_pos.vy = -1.0;
sat_pos.vz = -1.0;
sat_pos.dvx = -1.0;
sat_pos.dvy = -1.0;
sat_pos.dvz = -1.0;
sat_pos.range_m = -1.0;

```

Функция eci_to_ecef

```
function [satpos_ecef] = eci_to_ecef(s0, ti, satpos_eci)
```

```
%Имя функции: eci_to_ecef
```

```
%Функция преобразования координат
```

```
%Входные данные: s0 - истинное звездное время в текущий момент обсервации ,
```

```
%ti - текущее время; satpos_eci
```

```
%Структура satpos_eci
```

```
%satpos_eci.x - координата x в абсолютной неподвижной системе координат (ECI);
```

```
%satpos_eci.y - координата y в абсолютной неподвижной системе координат (ECI);
```

```
%satpos_eci.z - координата z в абсолютной неподвижной системе координат (ECI);
```

```
%satpos_eci.vx - скорость vx в абсолютной неподвижной системе координат (ECI);
```

```
%satpos_eci.vy - скорость vy в абсолютной неподвижной системе координат (ECI);
```

```
% satpos_eci.vz - скорость vz в абсолютной неподвижной системе координат (ECI);
```

```
%Выходные данные:
```

```
% Структура satpos_ecef
```

```
%satpos_ecef.x - координата x в подвижной системе координат (ECEF);
```

```
%satpos_ecef.y - координата y в подвижной системе координат (ECEF);
```

```
%satpos_ecef.z - координата z в подвижной системе координат (ECEF);
```

```

%satpos_ecef.vx - скорость по оси x в подвижной системе координат (ECEF);
%satpos_ecef.vy - скорость по оси y в подвижной системе координат (ECEF);
%satpos_ecef.vz - скорость по оси z в подвижной системе координат (ECEF);
%Коэффициенты
% SEC_IN_RAD - коэффициент преобразования секунд в радианы
% s0(radian) = s0 (sek) * SEC_IN_RAD, where
% SEC_IN_RAD = 2 * pi / (24 * 3600) = pi / 43200
SEC_IN_RAD = pi / 43200;
OMEGA_Z = 0.7292115e-4; %( скорость вращения Земли (angular speed of rotation of the Earth,
рад/сек)
s_zv = s0 * SEC_IN_RAD + OMEGA_Z * ti;
cos_s = cos(s_zv);
sin_s = sin(s_zv);

satpos_ecef.x = satpos_eci.x * cos_s + satpos_eci.y * sin_s;
satpos_ecef.y = -satpos_eci.x * sin_s + satpos_eci.y * cos_s;
satpos_ecef.z = satpos_eci.z;

satpos_ecef.vx = satpos_eci.vx * cos_s + satpos_eci.vy * sin_s + OMEGA_Z * satpos_ecef.y;
satpos_ecef.vy = -satpos_eci.vx * sin_s + satpos_eci.vy * cos_s - OMEGA_Z * satpos_ecef.x;
satpos_ecef.vz = satpos_eci.vz;

```

Функция top_coord

```

function [satvis] = top_coord(prn, rec_llh, rec_xyz, nlo_xyz)
% Имя функции: top_coord
% Функция выполняет расчет топоцентрических координат объекта по заданным
%географическим (долгота, широта, высота) и геоцентрическим (x, y, z)
%координатам приемника, а также геоцентрическим координатам объекта(x, y, z)
%Входные данные:
%prn - номер спутника
%структура
% rec_llh.lat - широта (рад) приемника
%rec_llh.lon - долгота (рад) приемника
%rec_llh.h - высота (м) приемника
%структура
%прямоугольные геоцентрические координаты приемника (м)
% rec_xyz.x
%rec_xyz.y
%rec_xyz.z
%структура

```

```

% прямоугольные геоцентрические координаты объекта (м)
%nlo_xyz.x
%nlo_xyz.y
%nlo_xyz.z
%Выходные данные:
%структура satvis
% top.s - проекция вектора дальности на ось, направленную на Юг (South)
% top.e - проекция вектора дальности на ось, направленную на Восток (East)
% top.z - проекция вектора дальности на ось, направленную в Зенит
% top.daln - дальность до объекта
% top.az - угол азимута объекта
% top.el - угол видимости объекта
rx = nlo_xyz.x - rec_xyz.x;
ry = nlo_xyz.y - rec_xyz.y;
rz = nlo_xyz.z - rec_xyz.z;
r_sat = sqrt( rx * rx + ry * ry + rz * rz);
r_rec = sqrt((rec_xyz.x)^2 + (rec_xyz.y)^2 + (rec_xyz.z)^2);
top.r = r_sat;
rx1 = rx;
ry1 = ry;
rz1 = rz;
sin_lat = sin(rec_llh.lat);
cos_lat = cos(rec_llh.lat);
sin_lon = sin(rec_llh.lon);
cos_lon = cos(rec_llh.lon);
% Projections of vector of range in topocentric coordinate system:
top.e = -sin_lon * rx1 + cos_lon * ry1;
top.s = cos_lon * sin_lat * rx1 + sin_lon * sin_lat * ry1 - cos_lat * rz1;
top.z = cos_lat * cos_lon * rx1 + cos_lat * sin_lon * ry1 + sin_lat * rz1;
% azimuth: отсчет по часовой стрелке от оси направленной на Север (N or -S) (-top.s)
eps = 10e-10;
if ( (abs(top.e) < eps) || (abs(top.s) < eps))
    top.az = 0.0;
else
    top.az = atan2(top.e,-top.s);
end;
if (top.az < 0.0)
    top.az = top.az + pi * 2;
end;
% elevation:
cos_el_top = (rec_xyz.x * rx + rec_xyz.y * ry + rec_xyz.z * rz) / (r_sat * r_rec);
if ( cos_el_top >= 1.00 )

```

```

    el = 0.0;
else
    if ( cos_el_top <= -1.00 )
        el = pi;
    else
        el = acos(cos_el_top);
    end;
end;
top.el = pi / 2.0 - el;
satvis.el = top.el;
satvis.az = top.az;
satvis.r = r_sat;

```

Функция *rewrite_satpos*

```
function [result] = rewrite_satpos(nom, satpos)
```

```
%Имя функции:rewrite_satpos
```

```
%Функция предназначена для перезаписи структуры satpos в массив структур result
```

```

result(nom).x = satpos.x;
result(nom).y = satpos.y;
result(nom).z = satpos.z;
result(nom).vx = satpos.vx;
result(nom).vy = satpos.vy;
result(nom).vz = satpos.vz;

```

Функция *utc_nut*

```
function nut = utc_nut(t)
```

```
%Имя: utc_nut
```

```
%Функция предназначена для расчета нутации
```

```
%Входные данные:
```

```
%t=6.023472005475702e+002;
```

```
%Выходные данные:
```

```
%nut - нутация
```

```
R = 1296000; % ( 1r=360grad=1 296 000 сек)
```

```
RAD_SEK_ANGL = pi/(3600*180);
```

```
t2 = t * t;
```

```
t3 = t2 * t;
```

```
l = 485866.733 + (1325.0 * R + 715922.633) * t + 31.310 * t2 + 0.064 * t3;%1.034807679476340e+012
```

```
l1 = 1287099.804 + (99 * R + 1292581.224) * t - 0.577 * t2 - 0.012 * t3;
```

```
f = 335778.877 + (1342 * R + 295263.137) * t - 13.257 * t2 + 0.011 * t3;
```

```
dd = 1072261.307 + (1236 * R + 1105601.328) * t - 6.891 * t2 + 0.019 * t3;
```

```
omega = 450160.280 - (5 * R + 482890.539) * t + 7.455 * t2 + 0.008 * t3;
```

```
eps0 = 84381.448 - 46.8150 * t - 0.00059 * t2 + 0.001813 * t3;
```

```
% eps0 = 84381.447996 - 46.8150 * t - 0.00059 * t2 + 0.001813 * t3;
```



```

eps_d = utc_nut_fi_eps(t, l, l1, f, dd, omega, 'd','e');
eps_k = utc_nut_fi_eps(t, l, l1, f, dd, omega, 'k','e');
eps = eps0 + eps_d + eps_k;
cos_eps = cos(RAD_SEK_ANGL * eps ) / 15.0;
d_fi = utc_nut_fi_eps(t, l, l1, f, dd, omega, 'd', 'f');
k_fi = utc_nut_fi_eps(t, l, l1, f, dd, omega, 'k', 'f');
nut1 = d_fi * cos_eps;
nut2 = k_fi * cos_eps;
% nut3 = 0.00264 * sin(omega) + 0.000063 * sin(2.0 * omega)
nut3 = 0;
nut1_dop = nut1;
nut2_dop = nut2;
nut3_dop = nut3;
nut = nut1 + nut2 + nut3;

```

Функция *utc_nut_fi_ep*

```
function nut_fi_eps = utc_nut_fi_eps(t, l, l1, f, dd, omega, typ_nut, fi_eps)
```

```
% Имя функции:utc_nut_fi_eps
```

```
%Функция предназначена для расчета нутации по епсилон и фи
```

```
%Входные данные:
```

```
%t - значение параметра приведено в функции utc_nut;
```

```
%l - значение параметра приведено в функции utc_nut;
```

```
%l1 - значение параметра приведено в функции utc_nut;
```

```
%f -значение параметра приведено в функции utc_nut;
```

```
%dd - значение параметра приведено в функции utc_nut;
```

```
% omega - значение параметра приведено в функции utc_nut;
```

```
%typ_nut - признак параметра приведен в функции utc_nut;
```

```
%fi_eps - признак параметра приведен в функции utc_nut;
```

```
%Выходные данные:
```

```
%nut_fi_eps - значение нутации по по епсилон и фи
```

```
% применить функцию koef
```

```
[koef_id, koef_abd, koef_ik, koef_abk] = koef;
```

```
RAD_SEK_ANGL = pi/(3600*180);
```

```
if (typ_nut == 'd')
```

```
    n = 30;
```

```
else
```

```
    n = 76;
```

```
end;
```

```
sum_a = 0;
```

```
sum_b = 0;
```

```
for i = 1 : n
```

```
if (typ_nut == 'd')
```

```

s1 = koef_id(i,1) * l + koef_id(i,2) * l1 + koef_id(i,3) * f + koef_id(i,4) * dd + koef_id(i,5) * omega;
    if (fi_eps == 'f')
        a = koef_abd(i,1) * 1e-4;
        bt = koef_abd(i,2) * 1e-4;
    else
        a = koef_abd(i,3) * 1e-4;
        bt = koef_abd(i,4) * 1e-4;
    end;
else
s1 = koef_ik(i,1) * l + koef_ik(i,2) * l1 + koef_ik(i,3) * f + koef_ik(i,4) * dd + koef_ik(i,5) * omega;
    if (fi_eps == 'f')
        a = koef_abk(i,1) * 1e-4;
        bt = koef_abk(i,2) * 1e-4;
    else
        a = koef_abk(i,3) * 1e-4;
        bt = koef_abk(i,4) * 1e-4;
    end;
end;
if (fi_eps == 'f')
    sin_s1 = sin(RAD_SEK_ANGL * s1);
    sa = a * sin_s1;
    sb = bt * sin_s1;
else
    cos_s1 = cos(RAD_SEK_ANGL * s1);
    sa = a * cos_s1;
    sb = bt * cos_s1;
end;
arg = RAD_SEK_ANGL * s1;
sum_a = sum_a + sa;
sum_b = sum_b + sb;
end;
nut_fi_eps = sum_a + sum_b * t;

```

Функция koef

```
function[koef_id, koef_abd, koef_ik, koef_abk] = koef
```

```
% Имя функции:koef
```

```
% Функция предназначена для ввода коэффициентов при расчете нутации
```

```
koef_id = [ 0, 0, 0, 0, 1 ; % 1
```

```
0, 0, 0, 0, 2 ; % 2
```

```
-2, 0, 2, 0, 1 ; % 3
```

```
2, 0,-2, 0, 0 ; % 4
```

```
-2, 0, 2, 0, 2 ; % 5
```

1,-1, 0,-1, 0 ; % 6
 0,-2, 2,-2, 1 ; % 7
 2, 0,-2, 0, 1 ; % 8
 0, 0, 2,-2, 2 ; % 9
 0, -1, 0, 0, 0 ; % 10 % исправлено
 0, 1, 2,-2, 2 ; % 11
 0,-1, 2,-2, 2 ; % 12
 0, 0, 2,-2, 1 ; % 13
 -2, 0, 0,2, 0 ; % 14 % исправлено
 0, 0, 2,-2, 0 ; % 15
 0, 2, 0, 0, 0 ; % 16
 0, 1, 0, 0, 1 ; % 17
 0, 2, 2,-2, 2 ; % 18
 0,-1, 0, 0, 1 ; % 19
 -2, 0, 0, 2, 1 ; % 20
 0,-1, 2,-2, 1 ; % 21
 2, 0, 0,-2, 1 ; % 22
 0, 1, 2,-2, 1 ; % 23
 1, 0, 0,-1, 0 ; % 24
 2, 1, 0,-2, 0 ; % 25
 0, 0,-2, 2, 1 ; % 26
 0, 1,-2, 2, 0 ; % 27
 0, 1, 0, 0, 2 ; % 28
 -1, 0, 0, 1, 1 ; % 29
 0, 1, 2,-2, 0 ; % 30

koef_abd = [-171996.0,-174.2, 92025.0, 8.9; % 1
 2062.0, 0.2, -895.0, 0.5; % 2
 46.0, 0.0, -24.0, 0.0; % 3
 11.0, 0.0, 0.0, 0.0; % 4
 -3.0, 0.0, 1.0, 0.0; % 5
 -3.0, 0.0, 0.0, 0.0; % 6
 -2.0, 0.0, 1.0, 0.0; % 7
 1.0, 0.0, 0.0, 0.0; % 8
 -13187.0, -1.6, 5736.0,-3.1; % 9
 -1426.0, 3.4, 54.0,-0.1; % 10 % исправлено
 -517.0, 1.2, 224.0,-0.6; % 11
 217.0, -0.5, -95.0, 0.3; % 12
 129.0, 0.1, -70.0, 0.0; % 13
 -48.0, 0.0, 1.0, 0.0; % 14 % исправлено
 -22.0, 0.0, 0.0, 0.0; % 15
 17.0, -0.1, 0.0, 0.0; % 16

-15.0, 0.0, 9.0, 0.0; % 17
-16.0, 0.1, 7.0, 0.0; % 18
-12.0, 0.0, 6.0, 0.0; % 19
-6.0, 0.0, 3.0, 0.0; % 20
-5.0, 0.0, 3.0, 0.0; % 21
4.0, 0.0, -2.0, 0.0; % 22
4.0, 0.0, -2.0, 0.0; % 23
-4.0, 0.0, 0.0, 0.0; % 24
1.0, 0.0, 0.0, 0.0; % 25
1.0, 0.0, 0.0, 0.0; % 26
-1.0, 0.0, 0.0, 0.0; % 27
1.0, 0.0, 0.0, 0.0; % 28
1.0, 0.0, 0.0, 0.0; % 29
-1.0, 0.0, 0.0, 0.0]; % 30
koef_ik = [0, 0, 2, 0, 2; % 31
1, 0, 0, 0, 0; % 32
0, 0, 2, 0, 1; % 33
1, 0, 2, 0, 2; % 34
1, 0, 0, -2, 0; % 35
-1, 0, 2, 0, 2; % 36
0, 0, 0, 2, 0; % 37
1, 0, 0, 0, 1; % 38
-1, 0, 0, 0, 1; % 39
-1, 0, 2, 2, 2; % 40
1, 0, 2, 0, 1; % 41
0, 0, 2, 2, 2; % 42
2, 0, 0, 0, 0; % 43
1, 0, 2, -2, 2; % 44
2, 0, 2, 0, 2; % 45
0, 0, 2, 0, 0; % 46
-1, 0, 2, 0, 1; % 47
-1, 0, 0, 2, 1; % 48
1, 0, 0, -2, 1; % 49
-1, 0, 2, 2, 1; % 50
1, 1, 0, -2, 0; % 51
0, 1, 2, 0, 2; % 52
0, -1, 2, 0, 2; % 53
1, 0, 2, 2, 2; % 54
1, 0, 0, 2, 0; % 55
2, 0, 2, -2, 2; % 56
0, 0, 0, 2, 1; % 57
0, 0, 2, 2, 1; % 58

1, 0, 2,-2, 1; % 59
0, 0, 0,-2, 1; % 60
1,-1, 0, 0, 0; % 61
2, 0, 2, 0, 1; % 62
0, 1, 0,-2, 0; % 63
1, 0,-2, 0, 0; % 64
0, 0, 0, 1, 0; % 65
1, 1, 0, 0, 0; % 66
1, 0, 2, 0, 0; % 67
1,-1, 2, 0, 2; % 68
-1,-1, 2, 2, 2; % 69
-2, 0, 0, 0, 1; % 70
3, 0, 2, 0, 2; % 71
0,-1, 2, 2, 2; % 72
1, 1, 2, 0, 2; % 73
-1, 0, 2,-2, 1; % 74
2, 0, 0, 0, 1; % 75
1, 0, 0, 0, 2; % 76
3, 0, 0, 0, 0; % 77
0, 0, 2, 1, 2; % 78
-1, 0, 0, 0, 2; % 79
1, 0, 0,-4, 0; % 80
-2, 0, 2, 2, 2; % 81
-1, 0, 2, 4, 2; % 82
2, 0, 0,-4, 0; % 83
1, 1, 2,-2, 2; % 84
1, 0, 2, 2, 1; % 85
-2, 0, 2, 4, 2; % 86
-1, 0, 4, 0, 2; % 87
1,-1, 0,-2, 0; % 88
2, 0, 2,-2, 1; % 89
2, 0, 2, 2, 2; % 90
1, 0, 0, 2, 1; % 91
0, 0, 4,-2, 2; % 92
3, 0, 2,-2, 2; % 93
1, 0, 2,-2, 0; % 94
0, 1, 2, 0, 1; % 95
-1,-1, 0, 2, 1; % 96
0, 0,-2, 0, 1; % 97
0, 0, 2,-1, 2; % 98
0, 1, 0, 2, 0; % 99
1, 0,-2,-2, 0; % 100

0,-1, 2, 0, 1; % 101
1, 1, 0,-2, 1; % 102
1, 0,-2, 2, 0; % 103
2, 0, 0, 2, 0; % 104
0, 0, 2, 4, 2; % 105
0, 1, 0, 1, 0]; % 106
koef_abk = [-2274.0, -0.2, 977.0, -0.5; % 31
712.0, 0.1, -7.0, 0.0; % 32
-386.0, -0.4, 200.0, 0.0; % 33
-301.0, 0.0, 129.0, -0.1; % 34
-158.0, 0.0, -1.0, 0.0; % 35
123.0, 0.0, -53.0, 0.0; % 36
63.0, 0.0, -2.0, 0.0; % 37
63.0, 0.1, -33.0, 0.0; % 38
-58.0, -0.1, 32.0, 0.0; % 39
-59.0, 0.0, 26.0, 0.0; % 40
-51.0, 0.0, 27.0, 0.0; % 41
-38.0, 0.0, 16.0, 0.0; % 42
29.0, 0.0, -1.0, 0.0; % 43
29.0, 0.0, -12.0, 0.0; % 44
-31.0, 0.0, 13.0, 0.0; % 45
26.0, 0.0, -1.0, 0.0; % 46
21.0, 0.0, -10.0, 0.0; % 47
16.0, 0.0, -8.0, 0.0; % 48
-13.0, 0.0, 7.0, 0.0; % 49
-10.0, 0.0, 5.0, 0.0; % 50
-7.0, 0.0, 0.0, 0.0; % 51
7.0, 0.0, -3.0, 0.0; % 52
-7.0, 0.0, 3.0, 0.0; % 53
-8.0, 0.0, 3.0, 0.0; % 54
6.0, 0.0, 0.0, 0.0; % 55
6.0, 0.0, -3.0, 0.0; % 56
-6.0, 0.0, 3.0, 0.0; % 57
-7.0, 0.0, 3.0, 0.0; % 58
6.0, 0.0, -3.0, 0.0; % 59
-5.0, 0.0, 3.0, 0.0; % 60
5.0, 0.0, 0.0, 0.0; % 61
-5.0, 0.0, 3.0, 0.0; % 62
-4.0, 0.0, 0.0, 0.0; % 63
4.0, 0.0, 0.0, 0.0; % 64
-4.0, 0.0, 0.0, 0.0; % 65
-3.0, 0.0, 0.0, 0.0; % 66

3.0, 0.0, 0.0, 0.0; % 67
-3.0, 0.0, 1.0, 0.0; % 68
-3.0, 0.0, 1.0, 0.0; % 69
-2.0, 0.0, 1.0, 0.0; % 70
-3.0, 0.0, 1.0, 0.0; % 71
-3.0, 0.0, 1.0, 0.0; % 72
2.0, 0.0, -1.0, 0.0; % 73
-2.0, 0.0, 1.0, 0.0; % 74
2.0, 0.0, -1.0, 0.0; % 75
-2.0, 0.0, 1.0, 0.0; % 76
2.0, 0.0, 0.0, 0.0; % 77
2.0, 0.0, -1.0, 0.0; % 78
1.0, 0.0, -1.0, 0.0; % 79
-1.0, 0.0, 0.0, 0.0; % 80
1.0, 0.0, -1.0, 0.0; % 81
-2.0, 0.0, 1.0, 0.0; % 82
-1.0, 0.0, 0.0, 0.0; % 83
1.0, 0.0, -1.0, 0.0; % 84
-1.0, 0.0, 1.0, 0.0; % 85
-1.0, 0.0, 1.0, 0.0; % 86
1.0, 0.0, 0.0, 0.0; % 87
1.0, 0.0, 0.0, 0.0; % 88
1.0, 0.0, -1.0, 0.0; % 89
-1.0, 0.0, 0.0, 0.0; % 90
-1.0, 0.0, 0.0, 0.0; % 91
1.0, 0.0, 0.0, 0.0; % 92
1.0, 0.0, 0.0, 0.0; % 93
-1.0, 0.0, 0.0, 0.0; % 94
1.0, 0.0, 0.0, 0.0; % 95
1.0, 0.0, 0.0, 0.0; % 96
-1.0, 0.0, 0.0, 0.0; % 97
-1.0, 0.0, 0.0, 0.0; % 98
-1.0, 0.0, 0.0, 0.0; % 99
-1.0, 0.0, 0.0, 0.0; % 100
-1.0, 0.0, 0.0, 0.0; % 101
-1.0, 0.0, 0.0, 0.0; % 102
-1.0, 0.0, 0.0, 0.0; % 103
1.0, 0.0, 0.0, 0.0; % 104
-1.0, 0.0, 0.0, 0.0; % 105
1.0, 0.0, 0.0, 0.0]; % 106

Функция semi_axis_1

```

% double semi_axis(double t_dr, double i_incl, double ecc, double omega_n)
function [a_npp] = semi_axis_1(t_dr, i_incl, ecc, omega_n);
%Имя функции:semi_axis_1
%Функция вычисляет радиус орбиты спутника ГЛОНАСС в соответствии с интерфейсным контроль-
НЫМ
%документом ГЛОНАСС
%Входные данные:
%t_dr - драконический период обращения спутника ГЛОНАСС (секунды)
%i_incl - наклонение орбиты спутника ГЛОНАСС (радиан)
%ecc - эксцентриситет
%omega_n - аргумент перигея орбиты спутника ГЛОНАСС (радиан)
%Выходные данные:
%a_npp - большая полуось орбиты спутника ГЛОНАСС (километр)
A_PZ90_KM = 6378.136; % (Km) Equatorial radius of the Earth
J20 = -1082625.7e-9; % Factor at the second zone harmonic

% B_PZ90_KM = 6356.75136174571344; % AP_LAND (Km) Polar radius of the Earth */
%a_npp = 1;
MU = 398600.44; % (Km^3/cek^2) constant of a gravitational
epsilon = 1.0e-3;
sin_i = sin(i_incl);
sin_i2 = sin_i * sin_i;
v = -omega_n;% omega_n - angle of a perigee
ecc2_1 = 1.0 - ecc * ecc;
b1 = 2.0 - 2.5 * sin_i2;
b2 = sqrt(ecc2_1 * ecc2_1 * ecc2_1 );
b3 = 1.0 + ecc * cos(omega_n);
b3 = b3 * b3;
b4 = 1.0 + ecc * cos(v);
b5 = b4 * b4 * b4 / ecc2_1;
b = b1 * b2 / b3 + b5;
t_ock = t_dr;
tock_2pi = t_ock / (pi * 2);
p1_3 = 1.0 / 3.0;
a_dop = MU * tock_2pi * tock_2pi;
a_n = a_dop^(1/3);
% a_n = pow(a_dop, p1_3);
nn = 0;

dda = epsilon + 1;
while ( (dda > epsilon) & (nn < 50) )
    p = a_n * ecc2_1; % Focal parameter

```



```

ae_p = A_PZ90_KM / p;
b0 = 1.0 + (1.5 * J20 * ae_p * ae_p) * b;
t_ock = t_dr / b0;
tock_2pi = t_ock / (pi * 2);
a_dop = MU * tock_2pi * tock_2pi;
a_npp = a_dop^(1/3);
dda = abs(a_n - a_npp);
a_n = a_npp;
nn = nn + 1;
end;

```

Функция kepler

```

function [Ek] = kepler(Mk, ecc, eps);
%Имя функции:kepler
%Функция предназначена для решения уравнения Кеплера
% eps = 1.0E-15;
y = ecc * sin(Mk);
x1 = Mk;
x = y;
for k = 0 : 15
    x2 = x1;
    x1 = x;
    y1 = y;
    y = Mk - (x - ecc * sin(x));
    if (abs(y - y1) < eps)
        break
    end
    x = (x2 * y - x * y1) / (y - y1);
end %k
Ek = x;

```

Функция тар

```

function тар(N)
%Имя функции:тар
%Применена функция MatLab для внесения в графики орбитального движения изображения Земли
load('topo.mat','topo','topomar1');
[x,y,z] = sphere(50);
cla reset
%axis square off
props.AmbientStrength = 0.1;

```

```

props.DiffuseStrength = 1;
props.SpecularColorReflectance = .5;
props.SpecularExponent = 20;
props.SpecularStrength = 1;
props.FaceColor= 'texture';
props.EdgeColor = 'none';
props.FaceLighting = 'phong';
props.Cdata = topo;
surface(x*N,y*N,z*N,props);
light('position',[-1 0 1]);
light('position',[-1.5 0.5 -0.5], 'color', [.6 .2 .2]);
view(3) ;
grid on

```

4.3.5 Примеры выполнения комплекса программ **ORBITA_GL_NAVIOR**

Некоторые результаты, иллюстрирующие работу программ, изображены на рис. 4.8 - рис. 4.11.

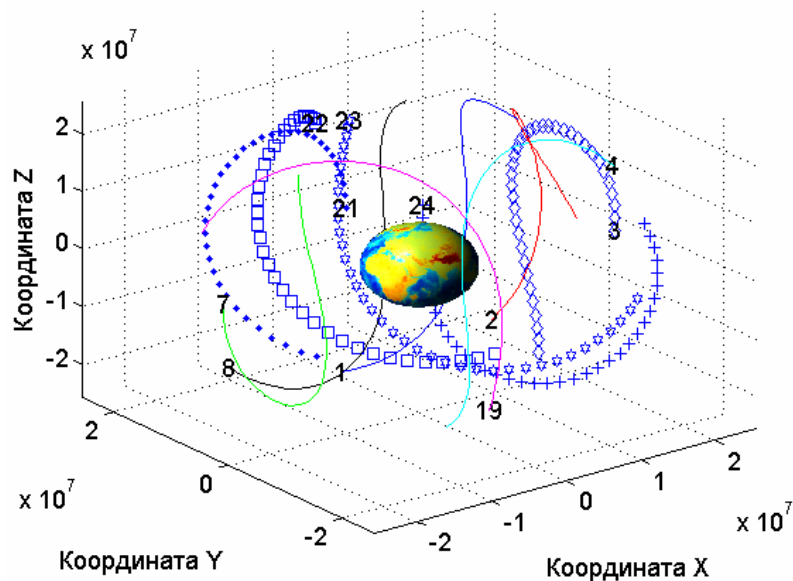


Рис. 4.8. Орбиты спутников ГЛОНАСС в системе координат ECEF

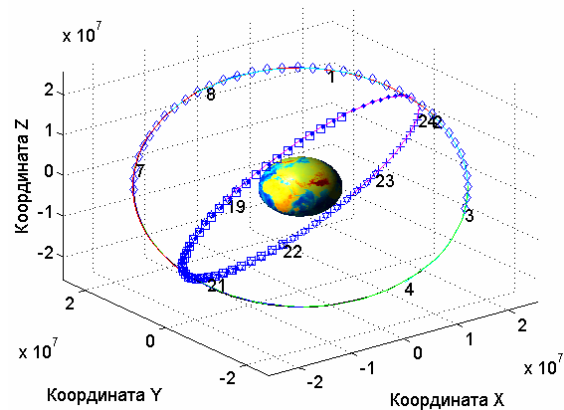


Рис. 4.9. Орбиты спутников ГЛОНАСС в системе координат ECI

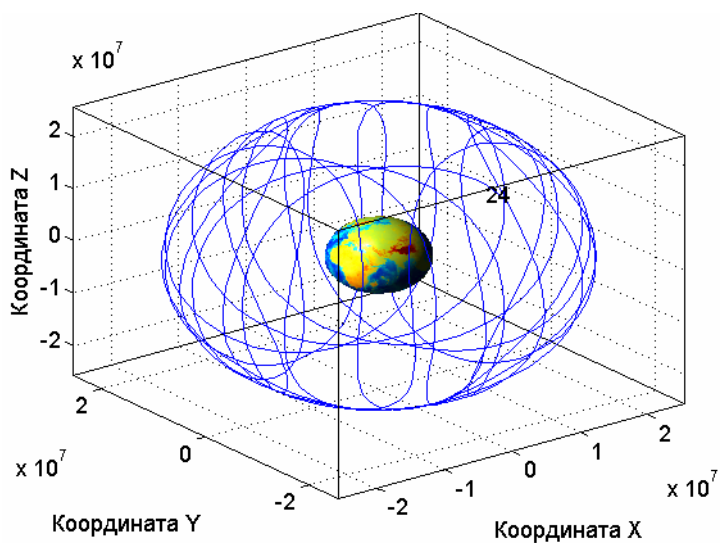


Рис. 4.10. Орбита спутника ГЛОНАСС № 24 за 7 суток

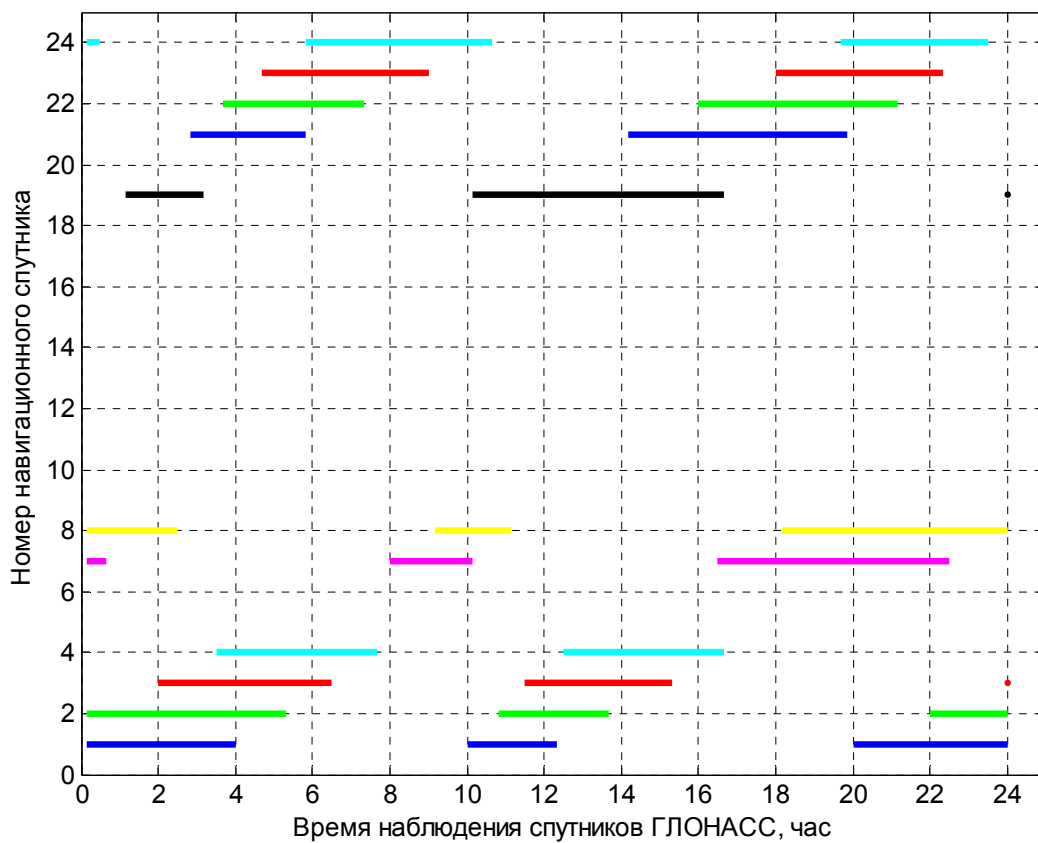


Рис. 4.11. Видимость спутников ГЛОНАСС

Программы имеют и другие возможности в зависимости от задач, которые может поставить специалист, исследующий орбитальное движение спутников.

РАЗДЕЛ 5 Преобразование данных навигационных спутников

5.1 Преобразование данных альманаха приемника СН 4701 в формат YUMA

5.1.1 Краткие сведения из теории

В спутниковых радионавигационных системах ГЛОНАСС и GPS эфемериды рассчитываются по принципиально разным формулам. В то же время весьма целесообразным является применение в задачах прогнозирования навигационных сеансов использовать одинаковый формат данных альманаха. Отсюда вытекает задача преобразования (конвертирования) альманаха системы ГЛОНАСС в широко распространенный формат YUMA без потери данных об альманахе спутников ГЛОНАСС.

Программы представленные в данном подразделе выполняют преобразование альманаха спутников GPS и ГЛОНАСС в формат YUMA, который воспринимается программой PLANNING и обеспечивает возможность планирования сеансов навигационных определений при применении спутниковых систем GPS и ГЛОНАСС. Программный комплекс подготовлен в точном соответствии с интерфейсными контрольными документами [

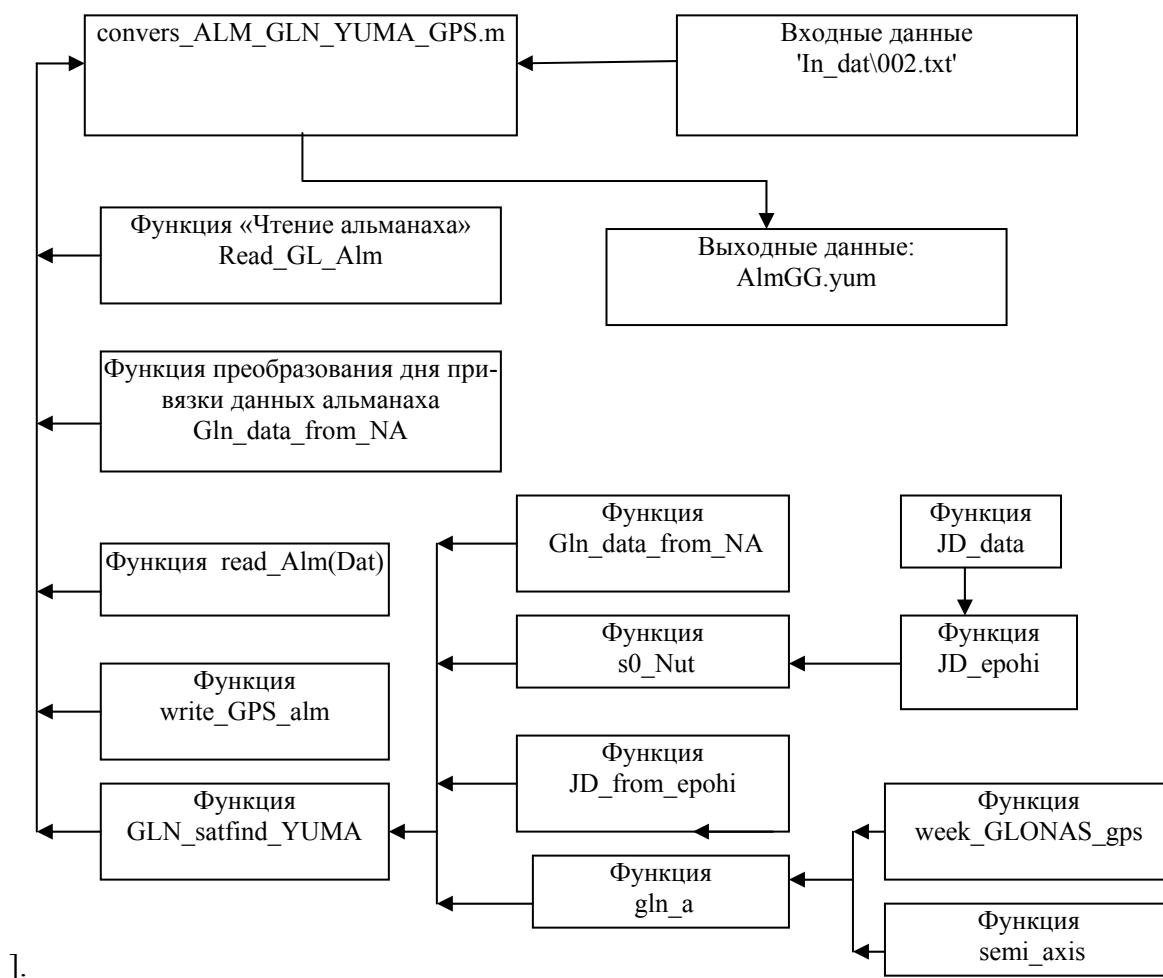


Рис. 5.1. Блок- схема программы преобразования альманаха ГЛОНАСС/GPS в формат YUMA

Для системы ГЛОНАСС Программы, записанные на CD-диск, находятся в папке ALM_CH4701_V3. Схема построения программного комплекса изображена на рис. 5.1. Основные комментарии и пояснения к программам приведены в листингах.

Цель лабораторной работы преобразование (конвертирование) данных альманаха навигационных спутников в формат YUMA.

5.1.2 Лабораторная работа 5. 1 «Конвертирование данных альманаха GPS и ГЛОНАСС в формат » YUMA

Пакет программ для выполнения лабораторной работы расположен в папке ALM_CH4701_V3. Для выполнения работы в качестве входных данных потребуется альманах спутников GPS и ГЛОНАСС, который можно получить с нескольких сайтов или с навигационного приемника, если приемник имеется в лаборатории.

Рекомендуется следующий порядок выполнения лабораторной работы.

1. Запишите альманах, с приемника **CH4701** в папку **In_dat**. Если не имеется возможности получить данные альманаха с приемника, то можно воспользоваться альманахом с именем 002.txt из приложения к данной книге.
2. Создайте папку **ALM_CH4701_V3_My** и скопируйте в нее все программы из папки **ALM_CH4701_V3** и папку **In_dat**.
3. Запустите MatLab.
4. Обратитесь к папке **ALM_CH4701_V3_My**, откройте ее, изучите функции, файлы и программные процедуры по комментариям и выполните **задания 1 и 2**.
5. **Задание 1.** Откройте файл **convers_ALM_GLN_YUMA_GPS.m** в папке **ALM_CH4701_V3_My** и присвойте переменной **Dat** имя альманаха из папки **In_dat**. Выполните файл. Результат выполнения запишется в файл **AlmGG.yum**. Это и будет преобразованный в формат YUMA альманах спутников GPS и ГЛОНАСС.
6. **Задание 2.** Загрузите альманах **AlmGG.yum** в программу прогнозирования **PLANNING** и убедитесь в его работоспособности.

5.1.3 Задание для самоподготовки

Ознакомьтесь и освоите работу с программой **PLANNING**

5.1.4 Функции и файлы из папки **ALM_CH4701_V3**

Файл *convers_ALM_GLN_YUMA_GPS.m*

```

%Имя файла:convers_ALM_GLN_YUMA_GPS.m
%от conversion- преобразование; программа преобразует данные альманаха GPS и ГЛОНАСС,
полученные
%с приемника "СН 4701" (производитель ГП "ОРИЗОН-НАВИГАЦИЯ" в формат YUMA
%Входные данные: альманах- записан в папке с именем In_dat, например,
%Dat = 'In_dat\GLN_all_23_10_06_17_45.alm';
%Выходные данные:записываются в файл с именем 'AlmGG.yum'
%Dat = 'In_dat\GLN_all8_11_06.alm';
%Dat = 'In_dat\GLN06_11_06.alm';
%Dat = 'In_dat\GG_24_03_03.alm';% GG24
Dat = 'In_dat\002.txt';% CN
%Имя файла для записи
Name = 'AlmGG.yum';
%Открытие файла для записи
fid =fopen(Name,'wt');
% Чтение альманаха
[alm_GPS,max_kol_GPS,alm_gln,max_kol_GLN] = read_Alm(Dat);
%Запись в файл альманаха GPS
write_GPS_alm(fid, alm_GPS);
%Полуоси земного эллипсоида
A_WGS84_M = 6378137.0 ;      % WGS-84 ellipsoid parameters
A_PZ90_M = 6378136.0;  %6 378 136 м - Equatorial radius of the Earth - ,большая полуось эллипсоида
A_PZ90_M = A_WGS84_M;
%Смещение времени GPS от UTC в секундах
%dt_Isf = 13;
leap_year = 2004;% Високосный год
timeUTC_leap.year = leap_year;
timeUTC_leap.mon = 0;
timeUTC_leap.day = 0;
GLN_satfind_YUMA( A_PZ90_M, timeUTC_leap, alm_gln, fid);

```

Функция read_Alm

```

function [alm,max_kol_GPS,alm_gln,max_kol_GLN] = read_Alm(Dat)
%Имя функции:read_Alm
%Функция читает данные альманаха GPS и ГЛОНАСС, полученные
%с приемника "СН 4701" (производитель ГП "ОРИЗОН-НАВИГАЦИЯ"
%Dat = '001.txt';
i=0;
for i=1:61
alm(i).ID = 0;
alm(i).Health=63;

```

```

alm_gln(i).ID=0;
alm_gln(i).Health=255;
end;
fid =fopen(Dat,'rt');
max_kol_GLN = 0;
max_kol_GPS = 0;

while notfeof(fid)
    str1=fscanf(fid,'%s',1); % GPS | GLN
    len1 = length(str1);
    if len1 == 4 % GPS
        if str1 == 'GPS:'
            max_kol_GPS = max_kol_GPS + 1;
            str1=fscanf(fid,'%c',4);
            ID=fscanf(fid,'%d',1);
            alm(ID).ID = ID;
            str2=fscanf(fid,'%c',13);
            Health=fscanf(fid,'%d',1);
            alm(ID).Health=Health;
            str3=fscanf(fid,'%c',19);
            eccentricity=fscanf(fid,'%g',1);
            alm(ID).ecc =eccentricity;

            str4=fscanf(fid,'%c',27);
            deltai=fscanf(fid,'%g',1) ; %i0
            alm(ID).deltai= deltai ; %i0

            str5=fscanf(fid,'%c',30);
            OMEGADOT=fscanf(fid,'%g',1)*1000;
            alm(ID).OMEGADOT=OMEGADOT;

            str6=fscanf(fid,'%c',37);
            A05=fscanf(fid,'%g',1);
            alm(ID).A05=A05;

            str7=fscanf(fid,'%c',32);
            omega0 =fscanf(fid,'%g',1);
            alm(ID).omega0 =omega0;

            str8=fscanf(fid,'%c',26);
            omega =fscanf(fid,'%g',1);
            alm(ID).omega=omega;

```



```

str9=fscanf(fid,'%c',19);
M0=fscanf(fid,'%g',1);
alm(ID).M0=M0;

str10=fscanf(fid,'%c',11);
Af0m=fscanf(fid,'%g',1)/1000;
alm(ID).Af0m=Af0m;

str11=fscanf(fid,'%c',10);
Af01=fscanf(fid,'%g',1);
alm(ID).Af01=Af01;

str12=fscanf(fid,'%c',11);
Af0=fscanf(fid,'%g',1);
alm(ID).Af0=Af0;

str13=fscanf(fid,'%c',19);
Week=fscanf(fid,'%g',1)+1024;
alm(ID).Week=Week;

str14=fscanf(fid,'%c',27);
TOA=fscanf(fid,'%d')/1000;
alm(ID).TOA =TOA;
    end;% if str1 == 'GPS'
else % GLONASS

    len1 = length(str1);
if len1 == 8

    max_kol_GLN = max_kol_GLN+1;
    str2=fscanf(fid,'%c',4);
    ID=fscanf(fid,'%d',1);
    alm_gln(ID).ID = ID;

    str3=fscanf(fid,'%c',15);
    Health=fscanf(fid,'%d',1);
    alm_gln(ID).Health=Health;

    str4=fscanf(fid,'%c',19);
    Hn=fscanf(fid,'%d',1);
    alm_gln(ID).Hn=Hn;

```

```

str5=fscanf(fid,'%c',34);
TaUGL=fscanf(fid,'%g',1);
alm_gln(ID).tau_n=TaUGL/1000;

str6=fscanf(fid,'%c',52);
LambdaN=fscanf(fid,'%g',1);
alm_gln(ID).LambdaN=LambdaN;

str7=fscanf(fid,'%c',24);
deltai=fscanf(fid,'%g',1);
alm_gln(ID).deltai=deltai;

str8=fscanf(fid,'%c',21);
eccentricity=fscanf(fid,'%g',1);
alm_gln(ID).ecc =eccentricity;

str9=fscanf(fid,'%c',39);
omegan=fscanf(fid,'%g',1);
alm_gln(ID).omegan =omegan;

str10=fscanf(fid,'%c',48);
TLambdaN=fscanf(fid,'%g',1);
alm_gln(ID).TLambdaN =TLambdaN/1000-10800;

str11=fscanf(fid,'%c',27);
Tdr=fscanf(fid,'%g',1);
alm_gln(ID).Tdr =Tdr/1000;

str12=fscanf(fid,'%c',48);
dTdr=fscanf(fid,'%g',1);
alm_gln(ID).dTdr=dTdr;%?единицы измерения

str13=fscanf(fid,'%c',16);
Na=fscanf(fid,'%g',1);
alm_gln(ID).Na =Na;

end % if len1 == 4 % GPS
end % while
end
fclose(fid);

```

Функция *Gln_data_from_NA*

```
function [time.UTC] = Gln_data_from_NA(leap_year, day_from_leap);
%Имя функции:Gln_data_from_NA
%function [time.UTC] = Gln_data_from_NA(leap_year, day_from_leap);
% преобразует NA - день привязки альманаха
% от ближайшего високосного года - leap_year
% в текущую дату: timeUTC (год, месяц, день)

DnMon = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]; %количество дней в месяцах
n4 = mod(leap_year, 4);
n100 = mod(leap_year, 100);
n400 = mod(leap_year, 400);
if (n4 == 0)
    n_leap = 1;
else
    n_leap = 0;
end
if ((n100 == 0) & (n400 > 0))
    n_leap = 0;
end
if (day_from_leap > (365 + n_leap))
    day = day_from_leap - (365 + n_leap);
    k = fix (day / 365);
    day = mod(day, 365);
else
    day = day_from_leap;
    k = -1;
end;
god = leap_year + k + 1;
if (god > leap_year)
    n_leap = 0;
end;
mon = 1;
mon_day = 31;
while (day > mon_day)
    day = day - mon_day;
    mon = mon + 1; % LK
    mon_day = DnMon(mon);
%    mon = mon+1;
if (mon == 2)
    mon_day = mon_day + n_leap;
```

```

    end
end
time_UTC.year = god;
time_UTC.mon = mon;
time_UTC.day = day;

```

Функция :read_Alm

```

function [alm,max_kol_GPS,alm_gln,max_kol_GLN] = read_Alm(Dat)
%Имя функции:read_Alm
%Функция читает данные альманаха GPS и ГЛОНАСС, полученные
%с приемника "СН 4701" (производитель ГП "ОРИЗОН-НАВИГАЦИЯ"
%Dat = '001.txt';
i=0;
for i=1:61
    alm(i).ID = 0;
    alm(i).Health=63;
    alm_gln(i).ID=0;
    alm_gln(i).Health=255;
end;
fid =fopen(Dat,'rt');
max_kol_GLN = 0;
max_kol_GPS = 0;

while notfeof(fid)
    str1=fscanf(fid,'%s',1); % GPS | GLN
    len1 = length(str1);
    if len1 == 4 % GPS
        if str1 == 'GPS:'
            max_kol_GPS = max_kol_GPS + 1;
            str1=fscanf(fid,'%c',4);
            ID=fscanf(fid,'%d',1);
            alm(ID).ID = ID;
            str2=fscanf(fid,'%c',13);
            Health=fscanf(fid,'%d',1);
            alm(ID).Health=Health;
            str3=fscanf(fid,'%c',19);
            eccentricity=fscanf(fid,'%g',1);
            alm(ID).ecc =eccentricity;

            str4=fscanf(fid,'%c',27);
            deltai=fscanf(fid,'%g',1) ; %i0
            alm(ID).deltai= deltai ; %i0

```

```
str5=fscanf(fid,'%c',30);
OMEGADOT=fscanf(fid,'%g',1)*1000;
alm(ID).OMEGADOT=OMEGADOT;
```

```
str6=fscanf(fid,'%c',37);
A05=fscanf(fid,'%g',1);
alm(ID).A05=A05;
```

```
str7=fscanf(fid,'%c',32);
omega0 =fscanf(fid,'%g',1);
alm(ID).omega0 =omega0;
```

```
str8=fscanf(fid,'%c',26);
omega =fscanf(fid,'%g',1);
alm(ID).omega=omega;
str9=fscanf(fid,'%c',19);
M0=fscanf(fid,'%g',1);
alm(ID).M0=M0;
```

```
str10=fscanf(fid,'%c',11);
Af0m=fscanf(fid,'%g',1)/1000;
alm(ID).Af0m=Af0m;
```

```
str11=fscanf(fid,'%c',10);
Af01=fscanf(fid,'%g',1);
alm(ID).Af01=Af01;
```

```
str12=fscanf(fid,'%c',11);
Af0=fscanf(fid,'%g',1);
alm(ID).Af0=Af0;
```

```
str13=fscanf(fid,'%c',19);
Week=fscanf(fid,'%g',1)+1024;
alm(ID).Week=Week;
```

```
str14=fscanf(fid,'%c',27);
TOA=fscanf(fid,'%d')/1000;
alm(ID).TOA =TOA;
```

```
end;% if str1 == 'GPS'
else % GLONASS
```

```

len1 = length(str1);
if len1 == 8

    max_kol_GLN = max_kol_GLN+1;
    str2=fscanf(fid,'%c',4);
    ID=fscanf(fid,'%d',1);
    alm_gln(ID).ID = ID;

    str3=fscanf(fid,'%c',15);
    Health=fscanf(fid,'%d',1);
    alm_gln(ID).Health=Health;

    str4=fscanf(fid,'%c',19);
    Hn=fscanf(fid,'%d',1);
    alm_gln(ID).Hn=Hn;

str5=fscanf(fid,'%c',34);
    TaUGL=fscanf(fid,'%g',1);
    alm_gln(ID).tau_n=TaUGL/1000;

    str6=fscanf(fid,'%c',52);
    LambdaN=fscanf(fid,'%g',1);
    alm_gln(ID). LambdaN=LambdaN;

str7=fscanf(fid,'%c',24);
    deltai=fscanf(fid,'%g',1);
    alm_gln(ID).deltai=deltai;

str8=fscanf(fid,'%c',21);
    eccentricity=fscanf(fid,'%g',1);
    alm_gln(ID).ecc =eccentricity;

str9=fscanf(fid,'%c',39);
    omegan=fscanf(fid,'%g',1);
    alm_gln(ID).omegan =omegan;

str10=fscanf(fid,'%c',48);
    TLambdaN=fscanf(fid,'%g',1);
    alm_gln(ID).TLambdaN =TLambdaN/1000-10800;

str11=fscanf(fid,'%c',27);
    Tdr=fscanf(fid,'%g',1);

```

```

alm_gln(ID).Tdr =Tdr/1000;

str12=fscanf(fid,'%c',48);
dTdr=fscanf(fid,'%g',1);
alm_gln(ID).dTdr =dTdr;%?единицы измерения

str13=fscanf(fid,'%c',16);
Na=fscanf(fid,'%g',1);
alm_gln(ID).Na =Na;

end % if len1 == 4 % GPS
end % while
end
fclose(fid);

```

Функция write_GPS_alm

```

function [] = write_GPS_alm(fw, alm)
%Имя функции:write_GPS_alm
%Функция записывает альманах GPS в совместный альманах спутников GPS и ГЛОНАСС в формате YUMA
i=0;
format long e;
for i=1:31
    if alm(i).ID > 0

%Заголовок альманаха
        fprintf(fw,'**** Week %i almanAU for PRN-0%i *****\n',alm(i).Week, alm(i).ID);
%Номер спутника GPS
        fprintf(fw,'ID:                %i\n',alm(i).ID);

%Здоровье спутника GPS
        fprintf(fw,'Health:            %i\n', alm(i).Health);

%Эксцентриситет орбиты спутника GPS

        strdop = e_norm(alm(i).ecc, 9);
        fprintf(fw,'Eccentricity:      %s\n', strdop);
%Время от начала недели GPS, на которое заданы параметры альманаха
        fprintf(fw,'Time of Applicability(s):  %6.4f\n',alm(i).TOA);
%Наклонение орбиты спутника GPS

```

```

fprintf(fw,'Orbital Inclination(rad):  %0.10f\n', alm(i).deltai);
%Скорость изменения восходящего узла орбиты спутника GPS
strdop = e_norm(alm(i).OMEGADOT, 9);
fprintf(fw,'Rate of Right Ascen(r/s):  %s\n', strdop);
%Корень квадратный из большой полуоси орбиты спутника GPS
fprintf(fw,'SQRT(A) (m^1/2):  %4.7f\n',alm(i).A05);
%Долгота восходящего узла орбиты спутника GPS
strdop = e_norm(alm(i).omega0, 9);
fprintf(fw,'Right Ascen at Week(rad):  %s\n', strdop);
%Аргумент перигея орбиты спутника GPS
strdop = e_norm(alm(i).omega, 9);
    fprintf(fw,'Argument of Perigee(rad):  %s\n',strdop);
%Средняя аномалия спутника GPS
strdop = e_norm(alm(i).M0, 9);
fprintf(fw,'Mean Anom(rad):  %s\n', strdop);
%Коэффициенты полинома для учета поправок времени
strdop = e_norm(alm(i).Af0m, 9);
fprintf(fw,'Af0(s):  %s\n', strdop);

strdop = e_norm(alm(i).Af01, 9);
fprintf(fw,'Af1(s/s):  %s\n', strdop);
%Номер недели
fprintf(fw,'week:  %i\n',alm(i).Week);
%Запись альманаха GPS в файл
fprintf(fw,' \n');

    end; %if alm(i).ID > 0
end; %i

```

Функция GLN_satfind_YUMA

```

function [ ] = GLN_satfind_YUMA(a, timeUTC_leap, alm_gln, fid);
%Имя функции:GLN_satfind_YUMA
%функция обрабатывает входные данные
KOL_GLN = 24;
A_PZ90_KM = a / 1000;
    for ( i = 1 :KOL_GLN)
%alm_gln.Na -(сек) время привязки альманаха от начала предшествующего високосного года
day_from_leap = alm_gln(i).Na;
timeUTC = Gln_data_from_NA(timeUTC_leap.year, day_from_leap);
timeUTC.ti = 0.0;
nut = 0;
S0 = s0_Nut( timeUTC, nut);

```



```

time_s0 = S0.s0_m_mod ; %time_s0 - истинное звездное время в текущий момент обсервации

year = timeUTC.year;
% leap_year = fix( year / 4 ) * 4; % ближайший к текущему (предыдущий) високосный год
leap_year = timeUTC_leap.year; % ближайший к текущему (предыдущий) високосный год
%-----
% ti = timeUTC.ti; % текущее время обсервации от начала дня
% n00 = fix(ti / 86400);
% n0 - номер текущих суток внутри 4-х летнего периода (от ближайшего високосного года)
% n0 = JD_from_epohi(leap_year, timeUTC) + n00 + 1;
% [eci_current_loc, eci_rec_pos_xyz] = llh_to_eci(a, b, ti, time_s0, current_loc_pz90);
% satpos_eci = init_satpos_gln();
ti = alm_gln (i).TLambdaN; % текущее время обсервации от начала дня
n00 = fix(ti / 86400);
% n0 - номер текущих суток внутри 4-х летнего периода (от ближайшего високосного года)
n0 = JD_from_epohi(leap_year, timeUTC) + n00 + 1;
    prn = alm_gln (i).ID;
    health = alm_gln (i).Health;
    Hn = alm_gln (i).Hn;

    if ((prn > 0) & (Hn > 0))
        gln_a(A_PZ90_KM, i, n0, ti, time_s0, alm_gln, timeUTC_leap, fid);

    end; % f ( (prn > 0) & (health == 1))

end; % for ( i = 1 : KOL_GLN)

fclose(fid);

```

Функция Gln_data_from_NA

```

function [time.UTC] = Gln_data_from_NA(leap_year, day_from_leap);
%Имя функции:Gln_data_from_NA
%function [time.UTC] = Gln_data_from_NA(leap_year, day_from_leap);
% преобразует NA - день привязки альманаха
% от ближайшего високосного года - leap_year
% в текущую дату: timeUTC (год, месяц, день)

DnMon = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]; %количество дней в месяцах
n4 = mod(leap_year, 4);
n100 = mod(leap_year, 100);
n400 = mod(leap_year, 400);

```

```

if (n4 == 0)
    n_leap = 1;
else
    n_leap = 0;
end
if ((n100 == 0) & (n400 > 0))
    n_leap = 0;
end
if (day_from_leap > (365 + n_leap))
    day = day_from_leap - (365 + n_leap);
    k = fix (day / 365);
    day = mod(day, 365);
else
    day = day_from_leap;
    k = -1;
end;
god = leap_year + k + 1;
if (god > leap_year)
    n_leap = 0;
end;
mon = 1;
mon_day = 31;
while (day > mon_day)
    day = day - mon_day;
    mon = mon + 1; % LK
    mon_day = DnMon(mon);
%   mon = mon+1;
    if (mon == 2)
        mon_day = mon_day + n_leap;
    end
end
time.UTC.year = god;
time.UTC.mon = mon;
time.UTC.day = day;

```

Функция s0_Nut

```

function [S0] = s0_Nut( timeUTC, nut)
%Имя функции: s0_Nut
%функция рассчитывает истинное и среднее звездное время по формулам ( )
% среднее звездное время s0 на 0ч UTC
%year = 1993; mon = 1; day = 0;
%fprintf('function s0_m - start \n');

```

```

jd2000 = 2451545; % 12h UTC 1 января
% Применить функцию JD_data
[jd, day_year] = JD_data( timeUTC);
if (jd == NaN)
    s0_mod = NaN; h = NaN; min = NaN; sec = NaN;
    fprintf('function s0_m - end0 \n');
    return;
end;
jd = jd - 0.5;
d = jd - jd2000;
t = d / 36525.0; % 36525 - юлианский период 100 лет
    t2 = t * t;
h1 = 24110.54841;
%h1=6.0*3600.0+41.0*60.0+50.54841;
% h2 = 236.555367908 * d;
    h2 = 8640184.812866 * t ;
h3 = 0.093104 * t2;
h4 = t2 * t * 6.2E-6;
if ( nut == 0)
    na = 0;
else
    na = utc_nut(t);
end;
s0_m = h1 + h2 + h3 - h4;
S0.s0_nut = s0_m + na;
S0.s0_m_mod = mod(s0_m, 86400);
s0_day = floor(s0_m / 86400);
S0.s0_m_hour = S0.s0_m_mod / 3600.0;
S0.s0_m_hour = floor(S0.s0_m_mod / 3600);
sec_min = S0.s0_m_mod - S0.s0_m_hour * 3600;
S0.s0_m_min = floor(sec_min / 60);
S0.s0_m_sec = sec_min - S0.s0_m_min * 60;
S0.s0_nut_mod = mod(S0.s0_nut, 86400);
s0_day = floor(S0.s0_nut / 86400);
S0.s0_nut_hour = S0.s0_nut_mod / 3600.0;
S0.s0_nut_hour = floor(S0.s0_nut_mod / 3600);
sec_min = S0.s0_nut_mod - S0.s0_nut_hour * 3600;
S0.s0_nut_min = floor(sec_min / 60);
S0.s0_nut_sec = sec_min - S0.s0_nut_min * 60;

```

Функция :JD_from_epohi

```
function [jd] =JD_from_epohi( epoha, timeUTC);  
%Имя функции:JD_from_epohi  
%Функция вычисляет jd - количество дней от указанного года (epoha)  
% до текущей даты, указанной в структуре timeUTC представленной в виде  
% (timeUTC.year, timeUTC.mon, timeUTC.day)  
jd0 = JD_epohi(epoha) + 1; % 12h, 1 den January  
[day, day_year] = JD_data(timeUTC);  
jd = day - jd0;
```

Функция gln_a

```
function [ ] = gln_a(A_PZ90_KM, ns, n0, ti_current, time_s0, alm_gln, timeUTC_leap, fid);  
%Имя функции:gln_a  
%Функция формирует альманах ГЛОНАСС в формате YUMA. Данные альманаха ГЛОНАСС соот-  
ветствуют  
%данному альманаху ГЛОНАСС, записанным приемником СН 4701 и преобразованным в строгом со-  
ответствии  
%интерфейсному контрольному документу ГЛОНАСС  
%Постоянные:  
I_MID = 1.0995574287564; %(double)(PI * 63.0 / 180.0). I_MID - Mean value of an inclination of a plane  
of orbit of a satellite  
T_DR_MID = 43200.0; %Mean value a dragon of cycle time of a satellite  
  
MU = 398600.44; % (Km^3/cek^2) гравитационная постоянная (constant of a gravitational)  
OMEGA_Z = 0.7292115e-4; %(рад/cek)- Угловая скорость вращения Земли (Angular speed of rotation  
of the Earth )  
  
SEC_IN_RAD = 7.2722052166430e-005; %(PI / 43200.0 ) Number radian in second of time  
2*PI/(24*3600)=PI/43200  
HALF_PI = pi / 2;  
D7_3 = 7.0 / 3.0;  
D7_4 = 7.0 / 4.0;  
D7_6 = 7.0 / 6.0;  
D7_24 = 7.0 / 24.0;  
D49_72 = 49.0 / 72.0;  
nn = fix(ti_current / 86400);  
  
ti = ti_current - nn * 86400;%время в секундах  
i_incl = I_MID + alm_gln(ns).deltai; %наклонение орбиты спутника ГЛОНАСС с номером ns
```

```

ecc2_1 = 1.0 - alm_gln(ns).ecc * alm_gln(ns).ecc;
% t_dr = T_DR_MID + alm_gln(ns).Tdr; % GG24
t_dr = alm_gln(ns).Tdr; % драконический период спутника ГЛОНАСС (данные приемника СН 4701)
n_dr = pi * 2 / t_dr;
%1. Вычисление полуоси a_n:
a_n = semi_axis( A_PZ90_KM, t_dr, i_incl, alm_gln(ns).ecc, alm_gln(ns).omegan);
alm_gln(ns).a = a_n;

%2. Вычисление t_lambda_k - время пересечения восходящего узла орбиты спутника ГЛОНАСС
sin_i = sin(i_incl);
cos_i = cos(i_incl);
sin_i2 = sin_i * sin_i;
cos_i2 = cos_i * cos_i;
v = - alm_gln(ns).omegan; % omega_n - angle of a perigee
J = -0.00162393855 ; % J = 3/2 * C20; C20=1082.6257 * 10-6;
ae_a = A_PZ90_KM / a_n;
ae_a2 = ae_a * ae_a;
j_ae_a2 = J * ae_a2;

omega1 = j_ae_a2 * n_dr * cos_i / (ecc2_1 * ecc2_1);
n0_na = (n0 - alm_gln(ns).Na) ;
tz = ti - alm_gln(ns).TLambdaN + 86400.0 * n0_na;
wk = tz / t_dr;
wi = fix(wk);
wi2 = wi * wi;
t_lambda_kk = alm_gln(ns).TLambdaN + t_dr * wi + alm_gln(ns).dTdr * wi2;
t_lambda_k = t_lambda_kk - n0_na * 86400.0 ;
lambda_k = alm_gln(ns).LambdaN + (omega1 - OMEGA_Z) * (wi * t_dr + alm_gln(ns).dTdr * wi2) ;
%time_s0 - a true sidereal time to Greenwich midnight of date n0 to which time ti concerns
%time_s0 - истинное звездное время в текущий момент обсервации
time_s0 = 0;
time_s = time_s0 * SEC_IN_RAD + OMEGA_Z * t_lambda_k ;
omega_0 = lambda_k + time_s;
% Auxiliary values:
d1 = 1.0 - 1.5 * sin_i2;
j_ae_a2_d1 = j_ae_a2 * d1;
j_ae_a2_d = j_ae_a2 * (1.0 - 1.5 * sin_i);
j_ae_a2_sin_i = j_ae_a2 * sin_i;
j_ae_a2_sin_i2 = j_ae_a2 * sin_i2;
j_ae_a2_cos_i2 = j_ae_a2 * cos_i2;
% 3. Calculation of constants:
tau = 0;

```

```

l = cos(alm_gln(ns).omegan) * alm_gln(ns).ecc;
h = alm_gln(ns).ecc * sin(alm_gln(ns).omegan);
dop_x = (sqrt(1.0 - alm_gln(ns).ecc) * tan( v / 2.0));
dop_y = sqrt(1.0 + alm_gln(ns).ecc);
ee = 2.0 * atan2(dop_x, dop_y);
ma = ee - alm_gln(ns).ecc * sin(ee);
ll = ma + alm_gln(ns).omegan;

% for (j = 0; j <= 1; j++)
for (j = 1 : 2)

sin_1 = sin(ll);
sin_2l = sin(2 * ll);
sin_3l = sin(3 * ll);
sin_4l = sin(4 * ll);
cos_1 = cos(ll);
cos_2l = cos(2 * ll);
cos_3l = cos(3 * ll);
cos_4l = cos(4 * ll);
l_cosl = 1 * cos_1;
delta_a(j) = 2.0 * a_n * j_ae_a2_d1 * l_cosl + ...
            j_ae_a2 * sin_i * (0.5 * h * sin_1 - 0.5 * l_cosl + ...
            cos(2.0 * alm_gln(ns).LambdaN) + 3.5 * l * cos_3l + 3.5 * h * sin_3l);
delta_h(j) = j_ae_a2_d1 * (1 * n_dr * tau + sin_3l + ...
            1.5 * l * sin_2l - 1.5 * h * cos_2l) - ...
            0.25 * j_ae_a2_sin_i2 * (sin_1 - D7_3 * sin_3l + ...
            5.0 * l * sin_2l - 8.5 * l * sin_4l + ...
            8.5 * h * cos_4l + h * cos_2l) + j_ae_a2_cos_i2 * ...
            (tau * l * n_dr - 0.5 * l * sin_2l);
delta_l(j) = j_ae_a2_d1 * (- tau * h * n_dr + cos_1 + ...
            1.5 * l * cos_2l + 1.5 * h * sin_2l) - ...
            0.25 * j_ae_a2_sin_i2 * (- cos_1 - D7_3 * cos_3l - ...
            5.0 * h * sin_2l - 8.5 * l * cos_4l - ...
            8.5 * h * sin_4l + l * cos_2l) + j_ae_a2_cos_i2 * ...
            (- tau * h * n_dr + 0.5 * h * sin_2l);
d_omega(j) = j_ae_a2 * cos_i * (tau * n_dr + 3.5 * l * sin_1 - ...
            3.5 * h * cos_1 - 0.5 * sin_2l - D7_6 * sin_3l + ...
            D7_6 * h * cos_3l);
delta_i(j) = 0.5 * j_ae_a2_sin_i * cos_i * (- l * cos_1 + ...
            h * sin_1 + cos_2l + D7_3 * l * cos_3l + D7_3 * h * sin_3l);
delta_ll(j) = 2.0 * j_ae_a2_d * ... %           2.0 * j_ae_a2_d1 *
            (tau * n_dr + D7_4 * l * sin_1 - ...

```

```

        D7_4 * h * cos_1) + 3 * j_ae_a2_sin_i *...
        (- D7_24 * h * cos_1 - D7_24 * l * sin_1 -...
        D49_72 * h * cos_3l + D49_72 * l * sin_3l +...
        0.25 * sin_2l) + j_ae_a2 * cos_i *...
        (tau * n_dr + 3.5 * l * sin_1 - 2.5 * h * cos_1 -...
        0.5 * sin_2l - D7_6 * l * sin_3l + D7_6 * h * cos_3l);
tau = ti - t_lambda_k;
ll = ma + alm_gln(ns).omegan + n_dr * tau;
end;
% 4. Corrections to orbit elements of a satellite by second zone harmonic J20
% influence at the moments of time ti
dda = delta_a(2) - delta_a(1);
ddh = delta_h(2) - delta_h(1);
ddl = delta_l(2) - delta_l(1);
dd_omega = d_omega(2) - d_omega(1);
ddi = delta_i(2) - delta_i(1);
dd_ll = delta_ll(2) - delta_ll(1);

% 5. calculation of influenced elements of orbits at the moment of ti time
ai = a_n + dda;
hi = h + ddh;
li = l + ddl;
% ecc_i = alm_gln[ns].ecc;
ecc_i = sqrt(hi * hi + li * li);
if (ecc_i == 0)
    w_i = 0;
else
    if (li == ecc_i)
        w_i = HALF_PI;
    else
        if (li == -ecc_i)
            w_i = -HALF_PI;
        else
            if (li ~= 0)
                w_i = atan2(hi,li);
            else
                w_i = HALF_PI;
            end;
        end;
    end;
end;
end;
end;
end;

```

```

omega_i = omega_0 + dd_omega;

ii_incl = i_incl + ddi;
ll_z = ma + alm_gln(ns).omegan + n_dr * tau + dd_ll;
% ll_z = ma + alm_gln(ns).omegan + n_dr * (ti - t_lambda_k) + dd_ll;
mm_i = ll_z - w_i;
%-----

%timeUTC_leap
Na = alm_gln(ns).Na;
[wGPS,weekday] = week_GLONAS_gps(Na, timeUTC_leap);
Yuma(ns).week = wGPS;% - 1024 ;%варианты записи недели
Yuma(ns).id = alm_gln(ns).ID + 37;
%Запись альманаха ГЛОНАСС в формате YUMA
%Заголовок альманаха ГЛОНАСС. В заголовок вынесен номер (литера) частоты- Hn
fprintf(fid,'GLO*Week %i almanac for prn-%i NAUHN-%i***** \n',Yuma(ns).week,
Yuma(ns).id,alm_gln(ns).Hn);
%Номер спутника ГЛОНАСС 1...24. В заголовке номера спутников ГЛОНАСС 38...61.
fprintf(fid,'ID: %i\n',Yuma(ns).id-37);% для Planning
fprintf(fid,'ID: %i\n',Yuma(ns).id);% вариант записи номера спутника
%Здоровье спутников ГЛОНАСС: 000- здоров
Yuma(ns).Health = alm_gln(ns).Health ;
if (Yuma(ns).Health > 100)
    fprintf(fid,'Health: %i\n', Yuma(ns).Health);
else
    if (Yuma(ns).Health > 10)
        fprintf(fid,'Health: 0%i\n', Yuma(ns).Health);
    else
        fprintf(fid,'Health: 00%i\n', Yuma(ns).Health);
    end;
end;
%Ексцентриситет орбиты спутника
strdop = e_norm(alm_gln(ns).ecc, 9);
fprintf(fid,'Eccentricity: %s \n', strdop);
%Время, к которому относятся данные орбита спутника ГЛОНАСС, приведенное к началу недели
GPS.
%При необходимости нужно учесть поправку смещения времени GPS от UTC (в 2006 году- 14 се-
кунд)
strdop = e_norm((alm_gln(ns).TLambdaN + 86400 * (weekday)),9);
fprintf(fid,'Time of Applicability(s): %s \n',strdop);
%Наклонение орбиты спутника ГЛОНАСС
Yuma(ns).inc = alm_gln(ns).deltai;

```



```

fprintf(fid,'Orbital Inclination(rad):  %11.9e \n',Yuma(ns).inc);
%Скорость изменения восходящего узла орбиты спутника ГЛОНАСС
strdop = e_norm(omega1,9);
fprintf(fid,'Rate of Right Ascen(r/s):  %s\n', strdop);
%Корень квадратный из большой полуоси орбиты спутника ГЛОНАСС
Yuma(ns).sqrt_a = sqrt(a_n * 1000);
fprintf(fid,'SQRT(A) (m^1/2):          %11.6f \n',Yuma(ns).sqrt_a);
%Долгота восходящего узла орбиты спутника ГЛОНАСС
strdop = e_norm(omega_i,9);
fprintf(fid,'Right Ascen at Week(rad):  %s\n', strdop);
%Аргумент перигея орбиты спутника ГЛОНАСС
strdop = e_norm(w_i,9);
fprintf(fid,'Argument of Perigee(rad):  %s\n',strdop);
%Средняя аномалия спутника ГЛОНАСС
strdop = e_norm(mm_i,9);
fprintf(fid,'Mean Anom(rad):           %s\n', strdop);
%Коэффициенты полинома для учета поправок времени
strdop = e_norm(alm_gln(ns).tau_n,9);
fprintf(fid,'Af0(s):                   %s\n', strdop);
Yuma(ns).Af1 = 0.0;
fprintf(fid,'Af1(s/s):                  %11.9e\n', Yuma(ns).Af1);
%Номер недели
fprintf(fid,'week:                      %i \n\n',Yuma(ns).week);
%Номер дня от начала недели, к которому относятся данные альманаха
Yuma(ns).day = weekday;

```

Функция week_GLONAS_gps

```

function [wGPS,weekday] = week_GLONAS_gps(Na, timeUTC)
% Имя функции: week_GLONAS_gps
% Функция вычисляет :
% Выходные данные: wGPS, weekday.
%wGPS -номер GPS- недели,
% weekday - день недели (0 - воскресенье, 1 - понедельник, 2 - вторник, 3 - среда,
%4 - четверг, 5 - пятница, 6 - суббота.
% Входные данные:
%Na- номер суток, к которым относятся данные альманаха спутника ГЛОНАСС, отсчитываемые от
%ближайшего високосного года
% timesUTC -время UTC с начала текущей недели (секунда),
%Используемые константы:
% сдвиг времени между UTC и системным временем GPS на начало 2004 года
%diff.UTC_GPS = 13;(до конца 2005 года)
diff.UTC_GPS = 14;%с 2006 года)

```

```

%количество дней в месяцах
DnMon = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31];
% юлианский день начала отсчета недели GPS на ноль часов UTC с 5 на 6
% января 1980 года:
wGPS_0 = 2444244.5;
% Расчет номеров юлианского дня и дня года, от которого отсчитываются дни ГЛОНАСС
[JD, day_year] = JD_data(timeUTC);
%Сдвиг на ноль часов UTC на гринвичском меридиане
JD = JD - 0.5;
%Расчет номера GPS неделе стандартной функцией MatLab "floor"
wGPS = floor((JD + Na - wGPS_0) / 7);
%Расчет дня недели стандартной функцией MatLab "mod"
weekday = mod((JD+Na - wGPS_0) , 7);

```

Функция semi_axis

```

function [a_npp] = semi_axis(A_PZ90_KM, t_dr, i_incl, ecc, omega_n);
%Имя функции:semi_axis
%Функция вычисляет радиус орбиты спутника ГЛОНАСС в соответствии с интерфейсным контрольным
%документом ГЛОНАСС
% A_WGS84 = 6378.1370 ; %km
%A_PZ90_KM = A_WGS84;
% A_PZ90_KM = 6378.136;   %(Km) Equatorial radius of the Earth
J20 = -1082625.7e-9; % Factor at the second zone harmonic

% B_PZ90_KM = 6356.75136174571344; % AP_LAND (Km) Polar radius of the Earth */
%a_npp = 1;
MU = 398600.44;   % (Km^3/сек^2) constant of a gravitational
epsilon = 1.0e-3;
sin_i = sin(i_incl);
sin_i2 = sin_i * sin_i;
v = -omega_n; % omega_n - angle of a perigee
ecc2_1 = 1.0 - ecc * ecc;
b1 = 2.0 - 2.5 * sin_i2;
b2 = sqrt(ecc2_1 * ecc2_1 * ecc2_1);
b3 = 1.0 + ecc * cos(omega_n);
b3 = b3 * b3;
b4 = 1.0 + ecc * cos(v);
b5 = b4 * b4 * b4 / ecc2_1;
b = b1 * b2 / b3 + b5;
t_ock = t_dr;

```

```

tock_2pi = t_ock / (pi * 2);
p1_3 = 1.0 / 3.0;
a_dop = MU * tock_2pi * tock_2pi;
a_n = a_dop^(1/3);
% a_n = pow(a_dop, p1_3);
nn = 0;

dda = epsilon + 1;
while ( (dda > epsilon) & (nn < 50) )
    p = a_n * ecc2_1; % Focal parameter
    ae_p = A_PZ90_KM / p;
    b0 = 1.0 + (1.5 * J20 * ae_p * ae_p) * b;
    t_ock = t_dr / b0;
    tock_2pi = t_ock / (pi * 2);
    a_dop = MU * tock_2pi * tock_2pi;
    a_npp = a_dop^(1/3);
    dda = abs(a_n - a_npp);
    a_n = a_npp;
    nn = nn + 1;
end;

```

Функция JD_epochi

```

function jden = JD_epochi(epocha)
%Имя: JD_epochi
%Функция JD_epochi(epocha) рассчитывает номер юлианского дня
%до начала года (epocha) на 12h, 0 день, январь.
%Входные данные: epocha, размерность-год
%Выходные данные:
% jden- номер юлианского дня на 12h, 0 день, январь ( размерность -дни)
rk = mod(epocha,4);
if ( rk == 0 ) rk = 1.0;
else
    rk = 2.0 - rk * 0.25;
end;
n100 = floor(epocha / 100);
n400 = floor(epocha / 400);
jden = (4712 + epocha) * 365.25 + n400 - n100 + rk;
% fprintf('epocha=%d rk=%f jden=%6.2f \n', epocha, rk, jden);

```

Функция *JD_data*

```
function [JD, day_year] = JD_data(timeUTC)
%Имя:JD_data
% Функция JD_data(timeUTC) вычисляет :
%JD - номер юлианского дня, day_year - номер дня года.
%Входные данные:
%timeUTC.year - год,
% timeUTC.mon - месяц,
% timeUTC.day - день.
%Выходные данные:
%JD - юлианский день, day_year- день от начала года.
%Оригинальные функции: function jd0 = JD_erohi(year),
%(смотри комментарий).
% число дней от начала периода до 12ч. всемирного времени (полдень)
% указанной даты ( по Гринвичу)
%Входные данные для контрольного примера:
%year = 2000; mon = 1; day = 1;
%количество дней в месяцах
DnMon = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31];
%Вычисление номера юлианского дня опорной эпохи
jd0 = JD_erohi(timeUTC.year);
%Учет високосного года
nfebr = 0;
if mod(timeUTC.year,4) == 0
    nfebr = 1;
end;
%Расчетномера дня года
k = 0;
for i = 2 : timeUTC.mon
    k = k + DnMon(i - 1);
    if (i == 2)
        k = k + nfebr;
    end;
end;
day_year = k + timeUTC.day;
%Расчет номера юлианского дня
JD = jd0 + day_year;
%2451545 - номер юлианского дня; 1- первый день года.
```

5.2 Декодирование данных альманаха спутников GPS

5.2.1 Краткие сведения из теории

Спутниковые навигационные приемники обрабатывают данные навигационных спутников и вырабатывают информации в определенных форматах, как правило, в форматах приемников характерных для фирм-производителей. Рассмотрим программу преобразования данных альманаха, получаемых с одного из широко распространенных приемников фирмы Novatel, *ProPak-G2*, выполненного на базе платы OEM 4 [10, 11]..

Информация, вырабатываемая приемником, записывается в виде файла в бинарной форме и в форме кода ASCII. В программе декодирования альманаха используется бинарная форма. Декодируется сырой альманах (RAWALM, Message ID: 74) [10, 11]. Формат данных из оригинала [10, 11]. имеет вид

Таблица

Формат данных альманаха RAWALM

Field	Field type	Data Description	Format	Binary Bytes	Binary Offset
1	header	Log header		H	0
2	ref week	Almanac reference week number	Ulong	4	H
3	ref secs	Almanac reference time (seconds.)	Ulong	4	H+4
4	subframes	Number of subframes to follow	Ulong	4	H+8
5	svid	SV ID (satellite vehicle ID)a	UShort	2	H+12
6	data	Subframe page data.	Hex	30	H+14
7...	Next subframe offset = H + 12 + (subframe x 32)				
variable	xxxx	32-bit CRC (ASCII and Binary only)	Hex	4	H + 12 + (32 x subframes)
variable	[CR][LF]	Sentence terminator (ASCII only)	-	-	-

На CD-диске программа записана в папке ALM_ProPakG2

Цель лабораторной работы изучение процедур и правил декодирования данных навигационного приемника.

5.2.2 Лабораторная работа 5. 2 «Декодирование данных альманаха навигационных приемников на базе плат OEM- 4»

Пакет программ для выполнения лабораторной работы расположен в папке **ALM_ProPakG2** Для выполнения работы в качестве входных данных потребуется альманах спутников GPS в виде бинарного файла, который можно получить с навигационных приемников на базе плат OEM 4. Для тестирования программы такой альманах приводится в приложении.


```

    'sqrtA',0,'omega0',0,'omega',0,'M0',0,'af0',0,'af1',0);
end;
%диск и последовательность папок, из которых считывается файл с записью альманаха в бинарном
%формате. При необходимости можно указать любой другой путь к файлу
name_in = 'E:\MATLAB7\work\RAW_ALM_BIN_DAT\2006_05_09_16.31_G2_RAW).bin';
%открытие данных для чтения и записи
fw = fopen('E:\MATLAB7\work\RAW_ALM_BIN_ALM\2006_05_09_16.31_G2_RAW_OK).alm','wt');
%формирование CRC-кода для проверки считываемых данных
fid_crc = fopen(name_in,'rb');
fseek(fid_crc, 0, 'eof');
size1 = ftell(fid_crc)- 4;
fseek(fid_crc, size1, 'bof');
CRC = fread(fid_crc,1,'uint32');
fprintf('CRC_hex = %X \n',CRC);
fseek(fid_crc, 0, 'bof');
CRC_own = CalculateBlockCRC32(fid_crc, size1);
fprintf('CRC_own_hex=%X \n',CRC_own);
fclose(fid_crc);
%Чтение "сырых" данных альманаха
fid = fopen(name_in,'rb');
% Заголовок данных- Header Structure (Binary Message)
Syn_c = fread(fid,3,'char'); %B OEM-4 'Char'
Hear_d= fread(fid,1,'uchar');%B OEM-4 'Uchar'
Message_ID = fread(fid,1,'uint16');%B OEM-4 'Long'
Message_Type=fread(fid,1,'char'); %B OEM-4 'Char'
Port_Address = fread(fid,1,'char'); %B OEM-4 'Char'
Message_Length = fread(fid,1,'uint16'); %B OEM-4 'Ushort'
Sequenc_e = fread(fid,1,'uint16'); %B OEM-4 'Ushort'
Idle_Time = fread(fid,1,'char'); %B OEM-4 'Char'
Time_Status = fread(fid,1,'char'); %B OEM-4 'Enum'
week_1 = fread(fid,1,'uint16'); %B OEM-4 'Ushort'
Time = fread(fid,1,'ulong'); %B OEM-4 'Double'
N_ul = fread(fid,1,'uint');
d_fcc=fread(fid,1,'uint16');
d_fcc = dec2hex(d_fcc);
e1 = fread(fid,1,'uint16');
week =fread(fid,1,'uint');% неделя GPS
T0a= fread(fid,1,'ulong');% время привязки альманаха
kol_str= fread(fid,1,'ulong'); %количество строк

%Чтение данных по каждому спутнику
for i=1:kol_str

```

```

Nsv= fread(fid,1,'ushort'); % номер спутника согласно табл.20-V ICD GPS-200 C
TLM= fread(fid,3,'char');
TL_M = dec2hex(TLM,2);
HOW= fread(fid,3,'char');
HO_W = dec2hex(HOW,2);
e6= fread(fid,1,'char');

maska = bin2dec('11000000');
ID_dop = bitand(e6, maska);
data_ID = bitshift(ID_dop, -6);
maska = bin2dec('00111111');
SV_ID = bitand(e6, maska);

ecc1 = fread(fid,1,'char');
ecc2 = fread(fid,1,'char');
ecc = (bitshift(ecc1, 8) + ecc2)*2^(-21);%эксцентриситет

t0a= fread(fid,1,'char')*2^12 ;% время привязки альманаха
d1= fread(fid,1,'bit8');
d2=fread(fid,1,'char');
incl_angle= ((bitshift(d1, 8) + d2)*2^(-19) + 0.3) * pi;%наклонение орбиты спутника
d1= fread(fid,1,'bit8');
d2=fread(fid,1,'char');
omega_dot= (bitshift(d1, 8) + d2)*2^(-38) * pi;%скорость изменения восходящего узла
maska = bin2dec('00111111');
%здоровье спутника GPS
health_1 = fread(fid,1,'char');
if health_1 > 255
    health_1 = health_1;
end;
health = bitand(health_1,maska);
if health > 63
    health = health;
end;
maska = bin2dec('11000000');
health_0 = bitand(health_1,maska);
health_0 = bitshift(health_0, -6);
if health_0 > 3
    health_0 = health_0;
end;
d1 = fread(fid,1,'char');
d2 = fread(fid,1,'char');

```



```

d3 = fread(fid,1,'char');
sqrtA=(bitshift(d1, 16) + bitshift(d2, 8) + d3)*2^(-11);%корень квадратный из большей полуоси орбиты

d1 = fread(fid,1,'bit8');
sign2 = sign(d1);
d2 = fread(fid,1,'char');
d3 = fread(fid,1,'char');
omega0=((bitshift(d1, 16) + bitshift(d2, 8) + d3)*2^(-23)) * pi; %долгота восходящего узла орбиты спутника

d1 = fread(fid,1,'bit8');
sign1 = sign(d1);
d2 = fread(fid,1,'char');
d3 = fread(fid,1,'char');
omega=((bitshift(d1, 16) + bitshift(d2, 8) + d3)*2^(-23)) * pi; %аргумент перигея

d1 = fread(fid,1,'bit8');
d2 = fread(fid,1,'char');
d3 = fread(fid,1,'char');
M0=((bitshift(d1, 16) + bitshift(d2, 8) + d3)*2^(-23)) * pi;%средняя аномалия на время привязки t0a

af0_1 = fread(fid,1,'char');
af0_1_bin = dec2bin(af0_1);
d2 = fread(fid,1,'char');
d2bin = dec2bin(d2);
d3 = fread(fid,1,'char');
d3bin = dec2bin(d3);
maska = bin2dec('00011100');
d4 = bitand(d3,maska);

af0_2 = bitshift(d4, -2);

af0_dop = bitshift(af0_1, 3) + af0_2;
koef = 2^(-20);
af0 = var_sign(af0_dop,11,koef);%коэффициент коррекции времени af0

% af1:
maska = bin2dec('11100000');
d4 = bitand(d3,maska);

af1_2 = bitshift(d2, 3);

af1_1 = bitshift(d4, -5);

```

```

af1_dop = bitor(af1_2, af1_1);

koef = 2^(-38);
af1 = var_sign(af1_dop,11,koef);%коэффициент коррекции времени af1
%-----
if SV_ID > 0
    alm(SV_ID)= struct('SV_ID',SV_ID, 'week',week, 'health_0', health_0, 'health',health, 'ecc',ecc,...
        't0a',t0a, 'incl_angle',incl_angle, 'omega_dot',omega_dot,...
        'sqrtA',sqrtA, 'omega0',omega0, 'omega', omega, 'M0',M0,...
        'af0',af0, 'af1',af1);

end;
end;
%Формирование вывода данных альманаха в формате YUMA
for i=1:30 %63
    if alm(i).SV_ID > 0

if alm(i).SV_ID < 10
    fprintf(fw, '**** Week %i almaNAU for PRN-0%i *****\n',alm(i).week, alm(i).SV_ID);

    fprintf(fw,'ID:           0%i\n',alm(i).SV_ID);
else
    fprintf(fw, '**** Week %i almaNAU for PRN-%i *****\n',alm(i).week, alm(i).SV_ID);
    fprintf(fw,'ID:           %i\n',alm(i).SV_ID);
end;
if alm(i).health < 10
    fprintf(fw,'Health:       %i0%i\n', alm(i).health_0, alm(i).health);
else
    fprintf(fw,'Health:       %i%i\n', alm(i).health_0, alm(i).health);
end;

strdop = e_norm(alm(i).ecc, 10);
fprintf(fw,'Eccentricity:     %s\n', strdop);

fprintf(fw,'Time of Applicability(s):  %6.4f\n',alm(i).t0a);

fprintf(fw,'Orbital Incluation(rad):  %0.10f\n',alm(i).incl_angle);

strdop = e_norm(alm(i).omega_dot, 10);
fprintf(fw,'Rate of Right Ascen(r/s):  %s\n', strdop);

```

```

fprintf(fw,'SQRT(A) (m^1/2):          %4.7f\n',alm(i).sqrtA);

strdop = e_norm(alm(i).omega0, 10);
fprintf(fw,'Right Ascen at Week(rad):  %s\n', strdop);
if alm(i).omega < 0
    fprintf(fw,'Argument of Perigee(rad):  %1.10f\n',alm(i).omega);
else
    fprintf(fw,'Argument of Perigee(rad):  %1.10f\n',alm(i).omega);
end;

strdop = e_norm(alm(i).M0, 10);
fprintf(fw,'Mean Anom(rad):          %s\n', strdop);

strdop = e_norm(alm(i).af0, 10);
fprintf(fw,'Af0(s):                  %s\n', strdop);

strdop = e_norm(alm(i).af1, 10);
fprintf(fw,'Af1(s/s):                %s\n', strdop);

fprintf(fw,'week:                    %i\n',alm(i).week);
fprintf(fw,'\n');
end; %if alm(i).SV_ID > 0
end; %i
%CR_C = fread(fid,1,'uint32');
%CR_C = dec2hex(CR_C);
fclose(fid);%закрытия файла для чтения
fclose(fw);%закрытия файла для записи

```

Функция CRC32Value

```

function[ulCRC] = CRC32Value(i)
%Имя функции: CRC32Value
%Функция для контроля декодированных данных
CRC32_POLYNOMIAL = hex2dec('EDB88320');
ulCRC = i;
for (j = 1:8)
    a = bitand(ulCRC, 1);
    if a > 0
        ulCRC = bitxor(bitshift(ulCRC, -1), CRC32_POLYNOMIAL);
    else
        ulCRC = bitshift(ulCRC, -1);
    end;
end;
end;

```

Функция CalculateBlockCRC32

```
function [ulCRC] = CalculateBlockCRC32(fid, size1)
%Имя функции:CalculateBlockCRC32
%Функция для проверки CRC-кода
ulCRC = 0;
int_ulcrc = uint(32);
FF0 = hex2dec('00ffffff');
ff = hex2dec('ff');
ulCount = size1;
while ( ulCount > 0 )
    ulTemp1 = bitand( bitshift(ulCRC, -8 ), FF0);
    ucBuffer = fread(fid,1,'uint8');
    int_ulcrc = ulCRC;
    xor1 = bitxor(int_ulcrc, ucBuffer );
    param1 = bitand(xor1, ff);
    ulTemp2 = CRC32Value(param1 );
    ulCRC = bitxor(ulTemp1, ulTemp2);
    ulCount = (ulCount - 1);
end
```

Функция e_norm

```
function str_norm = e_norm(e_dat, mantissa);
%Имя функции:e_norm
%Функция для форматирования записи данных
format_e = num2str(mantissa);
format_e = strcat('%0.',format_e,'E');
str_norm = num2str(e_dat*10,format_e);
k=strfind(str_norm, '.');
if k>2
    str_norm = strcat(str_norm(1),'0.',str_norm(2:k-1), str_norm(k+1:length(str_norm)));
else
    str_norm = strcat('0.',str_norm(1:k-1), str_norm(k+1:length(str_norm)));
end;
```

Функция var_sign

```
function [result] = var_sign(a, poz, koef)
%Имя функции:var_sign
%Вспомогательная функция для декодирования коэффициентов af0, af1
%af0_bin = dec2bin(a)
sign_a = bitget(a,poz);
```

```

a_plus = bitset(a, poz, 0);
%a_plus_bin = dec2bin(a_plus)
if sign_a==0
    result = a_plus * koef;
else
    a_minus = bitcmp(a_plus,(poz - 1)) + 1;
    %a_bin = dec2bin(a_minus)
    result = - a_minus * koef;
end
end

```

5.3 Модель движения навигационных спутников GPS и ГЛОНАСС

5.3.1 Краткие сведения из теории

Развитие систем спутниковой радионавигации в настоящее время и в ближайшей перспективе ориентировано на применении нескольких систем. При этом в одном навигационном приемнике должны обрабатываться сигналы спутников систем, которые находятся в зоне видимости. В представленном комплексе программ средствами MatLab [7, 8] решена задача моделирования движения спутников GPS и ГЛОНАСС с применением данных альманаха спутниковых систем. Программирование выполнено по алгоритмам полностью соответствующим интерфейсному контрольному документу [9].

Программа рассчитывает углы азимута и места навигационных спутников GPS и ГЛОНАСС по данным совместного для обеих систем альманаха в формате YUMA. На CD-диске программа находится в папке 06_Vsion_GLONASS_GPS. Файл альманаха можно получить на сайте Национального авиационного университета или запросить по электронной почте snsatm@nau.edu.ua.

Цель лабораторной работы изучение орбитального движения навигационных спутников GPS и ГЛОНАСС с последующей визуализацией спутников на любой момент времени.

5.3.2 Лабораторная работа 5.3 «Модель движения и визуализация спутников GPS и ГЛОНАСС»

Рекомендуется следующий порядок выполнения лабораторной работы.

1. Создайте папку **Vsion_GLONASS_GPS_My**.
2. Скопируйте в папку **Vsion_GLONASS_GPS_My** из **Vsion_GLONASS_GPS** функции и файлы.

3. Запишите в папку **Vsion_GLONASS_GPS_My** альманах спутников GPS и ГЛОНАСС в формате YUMA и выполните **задания 1, 2, 3**
4. **Задание 1.** Разберите тексты функций и файлов в папке **Vsion_GLONASS_GPS_My**. Откройте файл **Vision_GLONASS_GPS.m** и внесите в него входные данные: имя альманаха и время, на которое выполняется моделирование (в комментариях к файлу это указано). Выполните файл и результаты визуализации с полярной диаграммы внесите в отчет.
5. **Задание 2.** Измените время наблюдения на 6 часов, выполните файл **Vision_GLONASS_GPS.m**. Занесите результаты наблюдения в отчет и объясните причины изменения конфигурации спутников.
6. **Задание 3.** В файле **Vision_GLONASS_GPS.m** измените дату наблюдения на два дня, сохранив время наблюдения и выполните файл. Занесите результаты выполнения программы в отчет и объясните причины, по которым спутники ГЛОНАСС изменили свое местоположения, а спутники GPS почти остались на том же месте.

5.3.3 Контрольные вопросы и задания для самоподготовки

1. Чему равен период обращения спутников GPS?.
2. Чему равен период обращения спутников ГЛОНАСС?
3. Применяя второй закон Кеплера и данные альманаха определите периоды обращения спутников GPS и спутников ГЛОНАСС.

5.3.4 Функции и файлы из папки **Vsion_GLONASS_GPS**

Файл *Vsion_GLONASS_GPS.m*

```

clear all
%Имя m-файла:Vsion_GLONASS_GPS.m
%Программа рассчитывает углы видимости навигационных спутников GPS и ГЛОНАСС (азимута и
места)
%видимых спутников на заданный момент времени
%Входные данные:
%файл альманаха в формате Yuma,имя файла альманаха присваивается
%переменной "Dat",например,Dat = 'имя файла альманаха';
%данные о начале отсчета "d2",d2='месяц/день/год';h=час;min=минута;s=секунда;
%координаты позиции приемника -lat(широта в радианах),lon (долгота в радианах,
%hg (высота в метрах);
%шаг с каким будут рассчитываться параметры (step,секунды);
%количество точек (L), в которых будут рассчитываться параметры орбит
%L=12*3600/step,L читается так: количество часов (например,12)

```

```

%число секунд в часе (3600) деленное на шаг (step)

%Постоянные:
%скорость вращения Земли
OMEGAeDOT=7.2921151467e-005;
%радиусы земного эллипсоида
A_WGS84=6378137.0;
B_WGS84=6356752.314;
%константы
mu=398600500000000;
F_CONST      = 4.442807633E-10;
kt=1;%установка времени на титульной надписи графика, определяется параметрами d2'; h; min; s и j
или L;
%Задание цветов для графики

color6(1:14) = ['k' 'b' 'g' 'r' 'c' 'm' 'r' ':' 'g' ':' 'b' ':' 'k' 'h'];
%Входные данные
%Dat = 'AlmGLONASS11_12_06.yum';
Dat = 'AlmGG_1.yum';
%Dat = 'YumaGL8_11_06.alm';
d2='12/21/2006';
h=14; min=52.0; s=58.0;

%координаты точки, из которой наблюдаются спутники
lat = 0.88032730015257;%50 градусов 26минут 20.54 секунд
lon = 0.53109641675259;%30 градусов 25 минут 46.4995секунд
hr=187.488;% метров
X_label=['Широта' ':' num2str(lat) ':' 'долгота' ':' num2str(lon) ':' 'высота' ':' num2str(hr)];
step=300;%шаг отсчета времени в секундах (300=5 минутам);шаг можно изменять
step=0;
%L=(24*3600) / step;% убрать % перед L для вывода таблицы улов видимости и азимута
L=1;% установить % перед L для вывода таблицы улов видимости и азимута
%Чтение альманаха

[alm,max_kol] = Yuma_GPS_GLONAS_Alm(Dat);
%max_kol = 32;
nom = 1;
i = 0;
k = 0;
i = 0;
% for i = 1 : max_kol
while ( i <61)

```

```

id = alm(nom).ID;
Health = alm(nom).Health;
%fprintf('1: i=%i k=%i nom=%i id=%i Health=%i \n', i, k, nom, id, Health);
if ( id > 0)
    Health = alm(nom).Health;
    if ( Health == 0)
        k = k + 1;
        nom_ns(k) = id;
        % fprintf('2: i=%i k=%i nom=%i id=%i Health=%i \n', i, k, nom, id, Health);
        nom = nom + 1;
    else
        nom = nom + 1;
    end;
else
    nom = nom + 1;
end;
end; % i
kol = k;
fprintf('kol=%i \n', kol);

nom_ns % - номер навигационного спутника
[Rx,Ry,Rz] = ECEFLH(lon, lat,hr);
%Rx=0;Ry=0;Rz=0;%центр масс Земли
%Начало отсчета текущего времени
[week,modeweek,d,dweek,weeks]=Tim(d2,h,min,s);

for j = 1% 0:L
%   for j = 1:L % 0:L
    t(j)=weeks+step*(j); %-step;
    %t1(j) = t(j)/60; %изменение дискретности текущего времени
    %d_wn = (week - alm(i).Week);
    %d_wn = 0;

    for k = 1 : kol
        i = nom_ns(k) ; % input "i" !!!

% fprintf('i=%i   alm(i).Week= %i \n',i, alm(i).Week)
        d_wn=(modeweek - alm(i).Week);%если в альманахе не учтено 1024
        tk = t(j) + d_wn * 604800 - alm(i).TOA;

        d_wn = abs(modeweek - alm(i).Week);

```



```

dd = 302400.0 + d_wn * 604800;

if ( ( alm(i).A05 > 0 ) & ( alm(i).Health == 0 ) )

while (abs(tk) > dd)
    if tk > dd
        tk = tk - 604800;
    else
        if tk < -dd
            tk = tk + 604800;
        end
    end % if
end % while

%Справочник по альманаху- цифра в скобках обозначает порядковый номер
%параметра альманаха в формате YUMA
%alm(ID).ID(1); alm(ID).Health(2); alm(ID).e(3); alm(ID).TOA(4); alm(ID).deltai(5);
%alm(ID).OMEGADOT(6); alm(ID).A05(7); alm(ID).omega0(8); alm(ID).omega(9);
%alm(ID).M0(10); alm(ID).Af0(11); alm(ID).Af1(12); alm(ID).Week(13);

n0=sqrt((mu) / (alm(i).A05^6));
j2 = 1082.68E-6;

re = (A_WGS84 + B_WGS84) / 2.;
sin55 = sin(55.0 * pi / 180.0);
dn = 1.5 * j2 * re * re / (alm(i).A05^4 ) * (1. - 1.5 * sin55 * sin55);
%dn = 0;
n=n0 * (1 + dn);
Mk = alm(i).M0 + n*tk;
e=alm(i).e;

%решение уравнения Кеплера
eps = 1.0E-15;
y = e * sin(Mk);
x1 = Mk;
x = y;
for k = 0 : 15 % количество итераций
    x2 = x1;
    x1 = x;
    y1 = y;
    y = Mk - (x - e * sin(x));
    if (abs(y - y1) < eps)

```

```

        break
    end
    x = (x2 * y - x * y1) / (y - y1);
end % kepler
Ek = x;
deltr = F_CONST * alm(i).e * alm(i).A05 * sin(Ek);
dt1 = alm(i).Af0 + alm(i).Af1 * tk + deltr;
tk = tk - dt1;
vd = 1. - alm(i).e * cos(Ek);
nuk = atan2(sqrt(1-alm(i).e^2)*sin(Ek) / vd, (cos(Ek)-alm(i).e) / vd);
Ek = acos((alm(i).e+cos(nuk))/(1+alm(i).e*cos(nuk)));

Fk = nuk + alm(i).omega;
uk = Fk;

ik = alm(i).deltai;
% alm(i)
rk = (alm(i).A05^2)*(1.0-alm(i).e*cos(Ek));
xkk = rk*cos(uk);
ykk = rk*sin(uk);

OMEGAk = alm(i).omega0+(alm(i).OMEGADOT-OMEGAeDOT)*tk-OMEGAeDOT*alm(i).TOA;
% fprintf('i=%i j=%i \n ',i, j);
Xk(j,i) = xkk*cos(OMEGAk)-ykk*cos(ik)*sin(OMEGAk);
Yk(j,i) = xkk*sin(OMEGAk)+ykk*cos(ik)*cos(OMEGAk);
Zk(j,i) = ykk*sin(ik);
%Dальности до спутников
PR(j,i) = sqrt((Xk(j,i) - Rx)^2 + (Yk(j,i) - Ry)^2 + (Zk(j,i) - Rz)^2);
%Перевод в географическую систему если требуется
%[lons,lats,hrs] = LLHECEF(Xk,Yk,Zk);
%(Llon(j),Llat(j),Hhr(j)) = [lons,lats,hrs];
%расчет угла видимости спутника
xls = Xk(j,i) - Rx;
yls = Yk(j,i) - Ry;
zls = Zk(j,i) - Rz;
range1 = sqrt(xls*xls+yls*yls+zls*zls);
if j>1
    doppler(j-1) = (range1 - range2) * 5.2514 / step;
end
range2 = range1;
P = sqrt(Rx * Rx + Ry * Ry + Rz * Rz);
tdot = ( Rx*xls+Ry*yls+Rz*zls)/range1/P;

```

```

xll = xls/range1;
yll = yls/range1;
zll = zls/range1;

if tdot >= 1.00
    b = 0.0;
elseif tdot <= -1.00
    b = pi;
else
    b = acos( tdot);
end
satang = pi/2.0 - b;
TT =satang;
TT(j,i) =TT;%угол видимости спутников
%расчет угла азимута спутников
xn =-cos(lon)*sin(lat);
yn =-sin(lon)*sin(lat);
zn = cos(lat);
xe =-sin(lon);
ye = cos(lon);
xaz = xe*xll + ye*yll;
yaz = xn*xll + yn*yll + zn*zll;
if (xaz == 0) or (yaz == 0)
    az(j)= 0;
else
az(j,i) = atan2(xaz,yaz);
end
if az(j,i) < 0
    az(j,i) = az(j,i) + pi*2;
end

AZ(j,i) =az(j,i) *180/pi;%угла азимута спутников в градусах
EL(j,i) = TT(j,i) *180/pi;%угла видимости спутников в градусах

% ПЕРЕСЧЕТ ВРЕМЕНИ
A(j)=mod(t(j),86400);
her(j)=floor(A(j)/3600);
m(j)=floor((A(j)-her(j)*3600)/60);
sek(j)=A(j)-her(j)*3600-m(j)*60;

%Построение полярной системы координат

```

```

if EL(j,i) < 0
    elp = 180;
else
    elp = (EL(j,i) - 90);
end;
azp = (AZ(j,i) + 90.0);
rad = pi / 180;
x0 = 0; y0 = 0;
xt(j,i) = (elp * cos(azp * rad));
yt(j,i) = -(elp * sin(azp * rad));
% fprintf('i=%i j=%i \n' , i, j);
end % i = ns
end; % if ( alm(i).A05 > 0 )

end % j = time

%ВНИМАНИЕ. Для вывода времени визуализации спутников на график установите kt
t_title=[d2 ' ' num2str(her(kt)) ':' num2str(m(kt)) ':' num2str(sek(kt))];
%X_label=['Широта' ':' num2str(lat) ':' 'долгота' ':' num2str(lon) ':' 'высота' ':' num2str(hr)];
%num2ctr(lat)
%num2str(her(kt))
%X_label=['66' ':'];
n = 6;
max_n = max(nom_ns);
n_end = mod(max(nom_ns),n);
n_end = mod(kol, n);
n2 = fix(kol / n) * n - n + 1;
%Формирование таблицы вывода времени UTC (Time), GPS (Tgps в секундах), номера спутника
(nns),
% углов видимости и азимута от времени и номера спутника
for i=1:n:kol
    fprintf(' Time      Tgps ');
    for k=1: n
        nns = nom_ns(i+k-1);
        fprintf(' %2i      ', nns);
    end;
    fprintf(' \n');
    for j=1:L
        fprintf('%2i:%2i:%2i %i ',her(j),m(j),sek(j), t(j));
        for k=1: n
            nns = nom_ns(i+k-1);
            fprintf('%6.1f *%6.1f ', EL(j,nns), AZ(j,nns) );

```

```

    end;
    fprintf('\n');
    end ; % J=1:L
    if (i) == (n2)
        n = n_end;
    end;
end% i
hold on
%Окружности уровней на круговой диаграмме видимости спутников
k1 = 10;
k2 = 30;
k3 = 50;
k4 = 70;
k5 = 85;
k6=90;
n=0;
for k=1:5:365
    n=n+1;
    m1 = pi / 180;
    x(n)=cos(k*m1);
    y(n)=sin(k*m1);
end;

%График круговой диаграммы
plot(k1*x(:),k1*y(:),'k:', k2*x(:),k2*y(:),'k:', k3*x(:),k3*y(:),'k:',k4*x(:),k4*y(:),'k:', k5*x(:),k5*y(:),'r',
k6*x(:),k6*y(:),'r');
text(5, 10,'80','FontSize',12,'FontName','TimesNewRoman');
text(18, 23,'60','FontSize',12,'FontName','TimesNewRoman');
text(32, 37,'40','FontSize',12,'FontName','TimesNewRoman');
text(45, 50,'20','FontSize',12,'FontName','TimesNewRoman');
text(55, 60,'5','FontSize',12,'FontName','TimesNewRoman');
text(62, 67,'0','FontSize',12,'FontName','TimesNewRoman');
grid on;
%Построение изображений видимых спутников на круговой диаграмме
i=1;
for k=1:kol
    i = nom_ns(k) ;
    if (i < 31)
        plot(xt(kt,i),yt(kt,i), 'Marker' , 'o','MarkerSize',20,'MarkerFaceColor','g' ) ;
    else
        plot(xt(kt,i),yt(kt,i), 'Marker' , 'o','MarkerSize',20,'MarkerFaceColor','r' ) ;
    end;
end;

```

```

title(t_title);
xlabel(X_label,'FontSize',12,'FontName','TimesNewRoman')
set(get(gcf,'CurrentAxes'),'FontSize',14,'FontName','TimesNewRoman')
hold on
if i<32
    str1 = num2str( i, 2);
else
    str1 = num2str( (i - 37), 2);
end
text(xt(kt,i), yt(kt,i),str1,'FontSize',14,'FontName','TimesNewRoman','HorizontalAlignment','center' );
axis( [-100 100 -100 100]);
%axis( [-90 90 -90 90]);
end
clear

```

Функция LLHECEF

```

function [lons,lats,hrs] = LLHECEF(Xk,Yk,Zk)
%Имя функции: LLHECEF
%Функция выполняет преобразование координат.
%Входные данные: Xk (Rx), Yk (Ry), Zk (Rz)- координаты в ECEF
%Выходные данные: lon-долгота, lat-широта, h-высота
%a,b-большая и малая полуоси эллипсоида
a=6378137.0;
b=6356752.314;
xy = sqrt(Xk*Xk + Yk*Yk);
thet = atan(Zk*a/(xy*b));
esq = 1.0-b*b/(a*a);
epsq = a*a/(b*b)-1.0;
lats = atan((Zk+epsq*b*(sin(thet)^3))/(xy-esq*a*(cos(thet)^3)));
lons = atan2(Yk,Xk);%!
if lons < 0
    lons = 2*pi + lons;
end ;
n = a*a/sqrt(a*a*cos(lats)*cos(lats) + b*b*sin(lats)*sin(lats));
hrs = xy/cos(lats)-n;
end

```

Функция ECEFLH

```

function [Rx,Ry,Rz] = ECEFLH(lon, lat,hr)
%Имя функции: ECEFLH
%Функция выполняет преобразование координат

```

```

%Входные данные:lon-долгота,lat-широта,h-высота;a,b-большая
%и малая полуоси эллипсоида
%Выходные данные:Rx,Ry,Rz- координаты в ECEF
%Для WGS-84
a=6378137.0;
b=6356752.314;
n=a*a/sqrt(a*a*cos(lat)*cos(lat)+b*b*sin(lat)*sin(lat));
Rx=(n+hr)*cos(lat)*cos(lon);
Ry=(n+hr)*cos(lat)*sin(lon);
Rz=(b*b/(a*a)*n+hr)*sin(lat);

```

Функция Tim

```

function [week,modeweek,d,dweek,weeks]=Tim(d2,h,min,s)
%Имя функции:Tim
%Функция по данным d2='месяц/день/год';h=час;min=минута;s=секунда рассчитывает
% номер недели ([week], при необходимости номер модифицированной недели (modeweek), число
дней,
%прошедших с 6 января 1980 года (d), номер дня недели (dweek), время GPS в секундах от начала не-
дели.
%Неделя начинается в ночь с субботы на воскресенье. Воскресенье нулевой день.
%d2='10/35/2003';h=23.0;min=59.0;s=59.0;
%week = ceil(DAYSDIF('1/6/1980',d2,3)/7);
week = floor(DAYSDIF('1/6/1980',d2,3)/7);% текущая неделя
modeweek=week;
%modeweek=week - 1024;
d = DAYSDIF('1/6/1980',d2,3);
dweek=fix(d-week*7);
weeks=(dweek)*24*60*60+h*60*60+min*60+s;

```

Результат выполнения программы изображен на рис. 5.2.

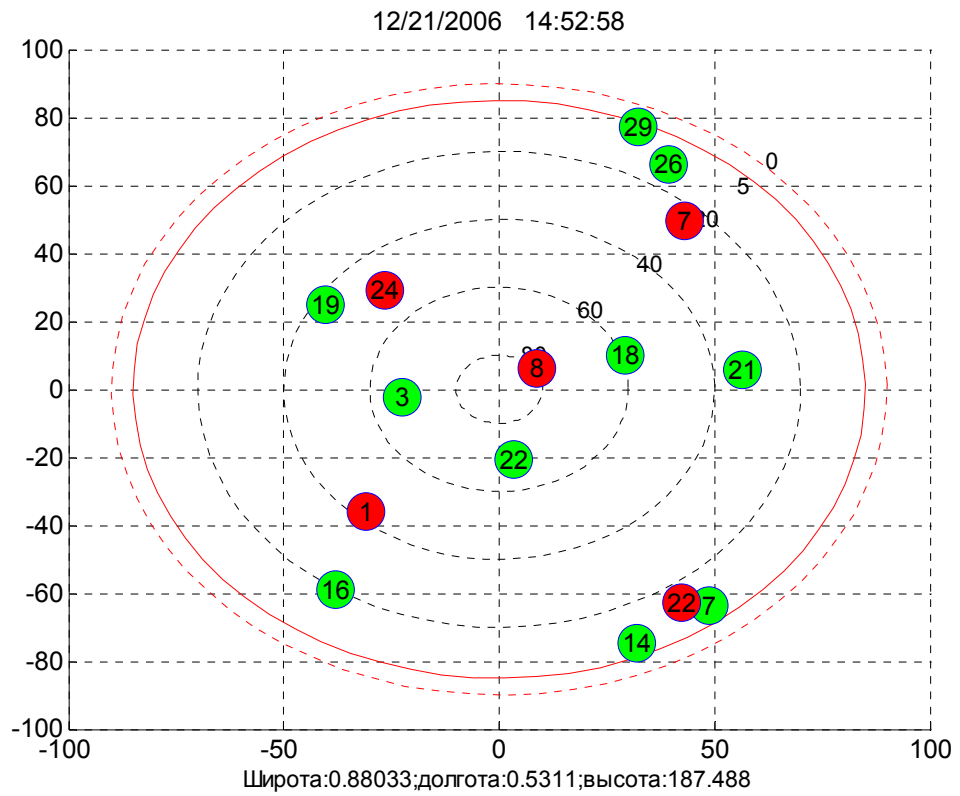


Рис. 5.2. Видимость спутников GPS и ГЛОНАСС

На рис. 5.3 показаны экспериментальные данные видимых спутников, полученные приемником НАВИОР 14.

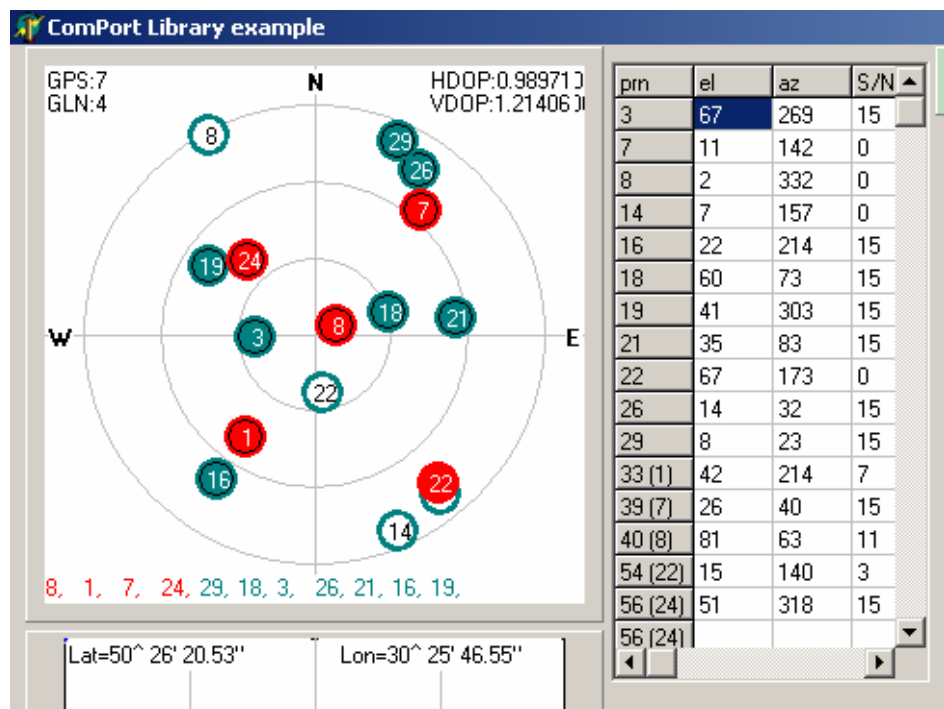


Рис. 5.3. Видимость спутников GPS и ГЛОНАСС (данные измерений)

Приведенная программа позволяет также рассчитать углы азимута и места в течение интервала времени, задаваемого пользователем и прогнозировать сеансы навигационных определений при совместном применении систем GPS и ГЛОНАСС.

5.4 Модель движения спутников GPS, ГЛОНАСС, GALILEO

5.4.1 Краткие сведения из теории

Рассмотрим кратко общие сведения о системе GALILEO, которая в настоящее время находится в стадии создания. Европейская спутниковая система GALILEO будет содержать 27 рабочих и три резервных навигационных спутника, расположенных равномерно на трех орбитах (рис. 5.4).

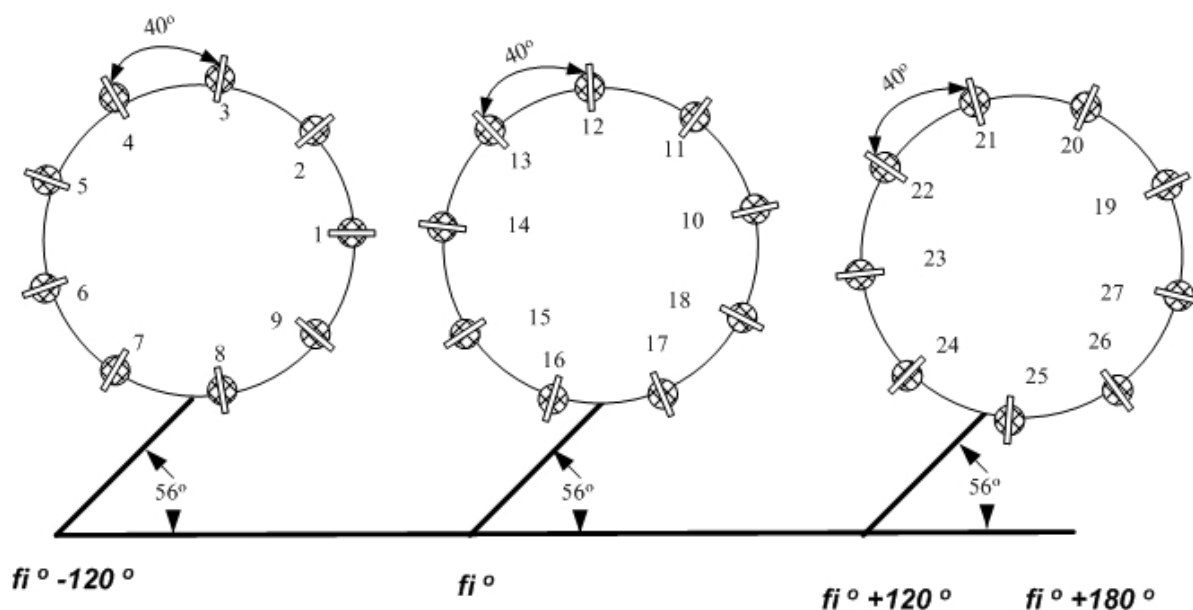


Рис. 5.4. Орбитальное положение спутников GALILEO (модель)

Долготы восходящих узлов орбит отстоят друг от друга на 120° . В плоскости орбит спутники расположены через 40° . Наклонение орбит к плоскости экватора составляет 56° . Средние значения больших полуосей орбит и времени обращения составляют 30049415.54 м и 12 часов 24 мин. соответственно.

Система GALILEO будет предоставлять следующие виды обслуживания:

открытое обслуживание, OS (Open Service)- сигналы доступные всем пользователям;
 служба поиска и спасения, SAR (Search and Rescue)- сигналы, доступные всем пользователям;

служба безопасности движения, SLS (Safety of Life Service) – сигналы доступны авиационным и морским потребителям на договорной основе;

коммерческая служба, CS (Commercial Service) – сигналы доступны пользователям на платной основе;

государственное регулируемое обслуживание, PRC (Public Regulated Service)- государственным службам предоставляются помехозащищенные и зашифрованные навигационные сигналы.

Данные об орбитальном движении спутников GALILEO будут отличаться от данных GPS только количественными значениями. Поскольку эта система еще не развернута, то в рассматриваемой ниже модели в интегральном альманахе GPS, ГЛОНАСС, GALILEO данные по GALILEO получены искусственно на основании сведений, имеющих в периодических публикациях.

Цель лабораторной работы изучение орбитального движения навигационных спутников GPS, ГЛОНАСС, GALILEO с последующей визуализацией спутников на любой момент времени.

5.4.2 Лабораторная работа 5. 4 «Модель движения и визуализация спутников GPS, ГЛОНАСС, GALILEO »

Комплекс программ для выполнения настоящей работы основан на пакете **Vsion_GLONASS_GPS**, в котором файл **Vsion_GLONASS_GPS.m** заменяется файлом **Vision_GL_GPS_GAL.m**, а входными данными является альманах спутников трех систем- GPS, ГЛОНАСС, GALILEO приведенный в приложении.

Рекомендуется следующий порядок выполнения лабораторной работы.

1. Создайте папку **Vision_GLONASS_GPS_GALILEO_My**.
2. Скопируйте в папку **Vision_GLONASS_GPS_GALILEO_My** из **Vsion_GLONASS_GPS** функции и файлы, из папки **Vision_GLONASS_GPS_GALILEO** файл **Vision_GL_GPS_GAL.m**, из приложения файл альманаха **AlmGGG.yum**. Комплекс готов к работе. Выполните **задания 1, 2**.
3. **Задание 1** .Откройте файл **Vision_GL_GPS_GAL.m**, убедитесь в правильности введенных входных данных (альманах , дата и время) исполните файл и результаты визуализации по каждой спутниковой системе внесите в отчет.
4. **Задание 2**. Измените в **Vision_GL_GPS_GAL.m** дату на 1 сутки, выполните файл, внесите результаты в отчет. Объясните причины, изменения местоположения спутников по каждой из систем.

5.4.3 Задание для самоподготовки

Подготовьте сравнительную характеристику спутниковых радионавигационных систем GPS, ГЛОНАСС, GALILEO

5.4.4 Файл из папки Vision_GLONASS_GPS_GALILEO

Файла *Vision_GL_GPS_GAL.m*

```
clear all
%Имя m-файла:Vision_GL_GPS_GAL.m
%Программа рассчитывает углы видимости навигационных спутников GPS, ГЛОНАСС, GALLILEO
%(азимута и места) спутников на заданный момент времени
%Входные данные:
%файл альманаха в формате Yuma, имя файла альманаха присваивается
%переменной "Dat", например, Dat = 'имя файла альманаха';
%данные о начале отсчета "d2", d2='месяц/день/год'; h=час; min=минута; s=секунда;
%координаты позиции приемника -lat(широта в радианах), lon (долгота в радианах),
%hr (высота в метрах);
%шаг с каким будут рассчитываться параметры (step, секунды);
%количество точек (L), в которых будут рассчитываться параметры орбит
%L=12*3600/step, L читается так: количество часов (например, 12)
%число секунд в часе (3600) деленное на шаг (step)

%Постоянные:
%скорость вращения Земли
OMEGAeDOT=7.2921151467e-005;
%радиусы земного эллипсоида
A_WGS84=6378137.0;
B_WGS84=6356752.314;
%константы
mu=3986005000000000;
F_CONST      = 4.442807633E-10;
kt=1;%установка времени на титульной надписи графика, определяется параметрами d2'; h; min; s и j
или L;
%Задание цветов для графики

color6(1:14) = ['k' 'b' 'g' 'r' 'c' 'm' 'r' ':' 'g' ':' 'b' ':' 'k' 'h'];
%Входные данные
Dat = 'AlmGGG.yum'; %альманах спутников GPS, ГЛОНАСС, GALLILEO
d2='12/21/2006';
h=14; min=52.0; s=58.0;

%координаты точки, из которой наблюдаются спутники
lat = 0.88032730015257;%50 градусов 26 минут 20.54 секунд
lon = 0.53109641675259;%30 градусов 25 минут 46.4995 секунд
hr=187.488;% метров
X_label=['Широта' ':' num2str(lat) ':' 'долгота' ':' num2str(lon) ':' 'высота' ':' num2str(hr)];
```

```

step=300;%шаг отсчета времени в секундах (300=5 минутам);шаг можно изменять
step=0;
%L=(24*3600) / step;% убрать % перед L для вывода таблицы улов видимости и азимута
L=1;% установить % перед L для вывода таблицы улов видимости и азимута
%Чтение альманаха

[alm,max_kol] = Yuma_GPS_GLONAS_Alm(Dat);
%max_kol = 32;
nom = 1;
i = 0;
k = 0;
i = 0;
% for i = 1 : max_kol
while ( i <91)
    id = alm(nom).ID;
    Health = alm(nom).Health;
%fprintf('1: i=%i k=%i nom=%i id=%i Health=%i \n', i, k, nom, id, Health);
    if ( id > 0)
        Health = alm(nom).Health;
        if ( Health == 0)
            k = k + 1;
            nom_ns(k) = id;
            % fprintf('2: i=%i k=%i nom=%i id=%i Health=%i \n', i, k, nom, id, Health);
            nom = nom + 1;
        else
            nom = nom + 1;
        end;
    else
        nom = nom + 1;
    end;
    i = i + 1;
end; % i
kol = k;
fprintf('kol=%i \n', kol);

nom_ns % - номер навигационного спутника
[Rx,Ry,Rz] = ECEFLH(lon, lat,hr);
%Rx=0;Ry=0;Rz=0;%центр масс Земли
%Начало отсчета текущего времени
[week,modeweek,d,dweek,weeks]=Tim(d2,h,min,s);

for j = 1% 0:L

```

```

% for j = 1:L % 0:L
    t(j)=weeks+step*(j); %-step;
    %t1(j) = t(j)/60; %изменение дискретности текущего времени
    %d_wn = (week - alm(i).Week);
    %d_wn = 0;

    for k = 1 : kol
        i = nom_ns(k) ; % input "i" !!!

% fprintf('i=%i   alm(i).Week= %i \n',i, alm(i).Week)
        d_wn=(modeweek - alm(i).Week);%если в альманахе не учтено 1024
        tk = t(j) + d_wn * 604800 - alm(i).TOA;

        d_wn = abs(modeweek - alm(i).Week);
        dd = 302400.0 + d_wn * 604800;

if ( ( alm(i).A05 > 0 ) & ( alm(i).Health == 0 ) )

while (abs(tk) > dd)
    if tk > dd
        tk = tk - 604800;
    else
        if tk < -dd
            tk = tk + 604800;
        end
    end % if
end % while

%Справочник по альманаху- цифра в скобках обозначает порядковый номер
%параметра альманаха в формате YUMA
%alm(ID).ID(1); alm(ID).Health(2); alm(ID).e(3); alm(ID).TOA(4); alm(ID).deltai(5);
%alm(ID).OMEGADOT(6); alm(ID).A05(7); alm(ID).omega0(8); alm(ID).omega(9);
%alm(ID).M0(10); alm(ID).Af0(11); alm(ID).Af1(12); alm(ID).Week(13);

    n0=sqrt((mu) / (alm(i).A05^6));
    j2 = 1082.68E-6;

    re = (A_WGS84 + B_WGS84) / 2.;
    sin55 = sin(55.0 * pi / 180.0);
    dn = 1.5 * j2 * re * re / (alm(i).A05^4 ) * (1. - 1.5 * sin55 * sin55);
    %dn = 0;
    n=n0 * (1 + dn);

```

```

Mk = alm(i).M0 + n*tk;
e=alm(i).e;

%решение уравнения Кеплера
eps = 1.0E-15;
y = e * sin(Mk);
x1 = Mk;
x = y;
for k = 0 : 15 % количество итераций
    x2 = x1;
    x1 = x;
    y1 = y;
    y = Mk - (x - e * sin(x));
    if (abs(y - y1) < eps)
        break
    end
    x = (x2 * y - x * y1) / (y - y1);
end % kepler
Ek = x;
deltr = F_CONST * alm(i).e * alm(i).A05 * sin(Ek);
dt1 = alm(i).Af0 + alm(i).Af1 * tk + deltr;
tk = tk - dt1;
vd = 1. - alm(i).e * cos(Ek);
nuk = atan2(sqrt(1-alm(i).e^2)*sin(Ek) / vd,(cos(Ek)-alm(i).e) / vd);
Ek = acos((alm(i).e+cos(nuk))/(1+alm(i).e*cos(nuk)));

Fk =nuk + alm(i).omega;
uk =Fk;

ik=alm(i).deltai;
% alm(i)
rk =(alm(i).A05^2)*(1.0-alm(i).e*cos(Ek));
xkk =rk*cos(uk);
ykk =rk*sin(uk);

OMEGAk =alm(i).omega0+(alm(i).OMEGADOT-OMEGAeDOT)*tk-OMEGAeDOT*alm(i).TOA;
% fprintf('i=%i j=%i \n ',i, j);
Xk(j,i) = xkk*cos(OMEGAk)-ykk*cos(ik)*sin(OMEGAk);
Yk(j,i) = xkk*sin(OMEGAk)+ykk*cos(ik)*cos(OMEGAk);
Zk(j,i) = ykk*sin(ik);
%Dальности до спутников
PR(j,i) = sqrt((Xk(j,i) - Rx)^2 + (Yk(j,i) - Ry)^2 + (Zk(j,i) - Rz)^2);

```

```

%Перевод в географическую систему если требуется
%[lons,lats,hrs] = LLHECEF(Xk,Yk,Zk);
%(Llon(j),Llat(j),Hhr(j)) = [lons,lats,hrs];
  %расчет угла видимости спутника
xls = Xk(j,i) - Rx;
  yls = Yk(j,i) - Ry;
  zls = Zk(j,i) - Rz;
range1 = sqrt(xls*xls+yls*yls+zls*zls);
if j>1
  doppler(j-1) = (range1 - range2) * 5.2514 / step;
end
range2 = range1;
P = sqrt(Rx * Rx + Ry * Ry + Rz * Rz);
tdot = ( Rx*xls+Ry*yls+Rz*zls)/range1/P;

xll = xls/range1;
  yll = yls/range1;
  zll = zls/range1;

if tdot >= 1.00
  b = 0.0;
elseif tdot <= -1.00
  b = pi;
else
  b = acos( tdot);
end
satang = pi/2.0 - b;
TT =satang;
TT(j,i) =TT;%угол видимости спутников
  %расчет угла азимута спутников
  xn =-cos(lon)*sin(lat);
  yn =-sin(lon)*sin(lat);
  zn = cos(lat);
  xe =-sin(lon);
  ye = cos(lon);
  xaz = xe*xll + ye*yll;
  yaz = xn*xll + yn*yll + zn*zll;
if (xaz == 0) or (yaz == 0)
  az(j)= 0;
else
az(j,i) = atan2(xaz,yaz);
end

```

```

if az(j,i) < 0
    az(j,i) = az(j,i) + pi*2;
end

AZ(j,i) = az(j,i) * 180/pi; %угла азимута спутников в градусах
EL(j,i) = TT(j,i) * 180/pi; %угла видимости спутников в градусах

% ПЕРЕСЧЕТ ВРЕМЕНИ
A(j) = mod(t(j), 86400);
her(j) = floor(A(j)/3600);
m(j) = floor((A(j) - her(j)*3600)/60);
sek(j) = A(j) - her(j)*3600 - m(j)*60;

% Построение полярной системы координат
if EL(j,i) < 0
    elp = 180;
else
    elp = (EL(j,i) - 90);
end;
azp = (AZ(j,i) + 90.0);
rad = pi / 180;
x0 = 0; y0 = 0;
xt(j,i) = (elp * cos(azp * rad));
yt(j,i) = -(elp * sin(azp * rad));
% fprintf('i=%i j=%i \n', i, j);
end % i = ns
end; % if ( alm(i).A05 > 0 )

end % j = time

%ВНИМАНИЕ. Для вывода времени визуализации спутников на график установите kt
t_title = [d2 ' ' num2str(her(kt)) ':' num2str(m(kt)) ':' num2str(sek(kt))];
%X_label = ['Широта' ':' num2str(lat) ':' 'долгота' ':' num2str(lon) ':' 'высота' ':' num2str(hr)];
%num2ctr(lat)
%num2str(her(kt))
%X_label = ['66' ':'];
n = 6;
max_n = max(nom_ns);
n_end = mod(max(nom_ns), n);
n_end = mod(kol, n);
n2 = fix(kol / n) * n - n + 1;

```



```

%Формирование таблицы вывода времени UTC (Time), GPS (Tgps в секундах), номера спутника
(nns),
% углов видимости и азимута от времени и номера спутника
for i=1:n:kol
    fprintf(' Time      Tgps ');
    for k=1: n
        nns = nom_ns(i+k-1);
        fprintf('      %2i      ', nns);
        end;
        fprintf('\n');
        for j=1:L
            fprintf('%2i:%2i:%2i  %i  ',her(j),m(j),sek(j), t(j));
            for k=1: n
                nns = nom_ns(i+k-1);
                fprintf('%6.1f*%6.1f  ', EL(j,nns), AZ(j,nns) );
            end;
            fprintf('\n');
        end ; % J=1:L
    if (i) == (n2)
        n = n_end;
    end;
end% i
hold on
%Окружности уровней на круговой диаграмме видимости спутников
k1 = 10;
k2 = 30;
k3 = 50;
k4 = 70;
k5 = 85;
k6=90;
n=0;
for k=1:5:365
    n=n+1;
    m1 = pi / 180;
    x(n)=cos(k*m1);
    y(n)=sin(k*m1);
end;

%График круговой диаграммы
plot(k1*x(:),k1*y(:),'k:', k2*x(:),k2*y(:),'k:', k3*x(:),k3*y(:),'k:',k4*x(:),k4*y(:),'k:', k5*x(:),k5*y(:),'r',
k6*x(:),k6*y(:),'r:');
text(5, 10,'80','FontSize',12,'FontName','TimesNewRoman');

```

```

text(18, 23,'60','FontSize',12,'FontName','TimesNewRoman');
text(32, 37,'40','FontSize',12,'FontName','TimesNewRoman');
text(45, 50,'20','FontSize',12,'FontName','TimesNewRoman');
text(55, 60,'5','FontSize',12,'FontName','TimesNewRoman');
text(62, 67,'0','FontSize',12,'FontName','TimesNewRoman');
grid on;
%Построение изображений видимых спутников на круговой диаграмме
i=1;
for k=1:kol
i = nom_ns(k) ;
if (i < 31)
plot(xt(kt,i),yt(kt,i), 'Marker' , 'o','MarkerSize',20,'MarkerFaceColor','g' ) ;
end;
if ((i > 31) & ( i < 62))
plot(xt(kt,i),yt(kt,i), 'Marker' , 'o','MarkerSize',20,'MarkerFaceColor','r' ) ;
end;
if (i > 61)
plot(xt(kt,i),yt(kt,i), 'Marker' , 'o','MarkerSize',20,'MarkerFaceColor','y' ) ;
end;
title(t_title);
xlabel(X_label,'FontSize',12,'FontName','TimesNewRoman')
set(get(gcf,'CurrentAxes'),'FontSize',14,'FontName','TimesNewRoman')
hold on
str1 = num2str( i, 2);
%{
if i<32
str1 = num2str( i, 2);
else
str1 = num2str( (i), 2);
end
%}
text(xt(kt,i), yt(kt,i),str1,'FontSize',14,'FontName','TimesNewRoman','HorizontalAlignment','center' );
axis( [-100 100 -100 100]);
%axis( [-90 90 -90 90]);
end
clear

```

Результат выполнения программы изображен на рис. 5.5.

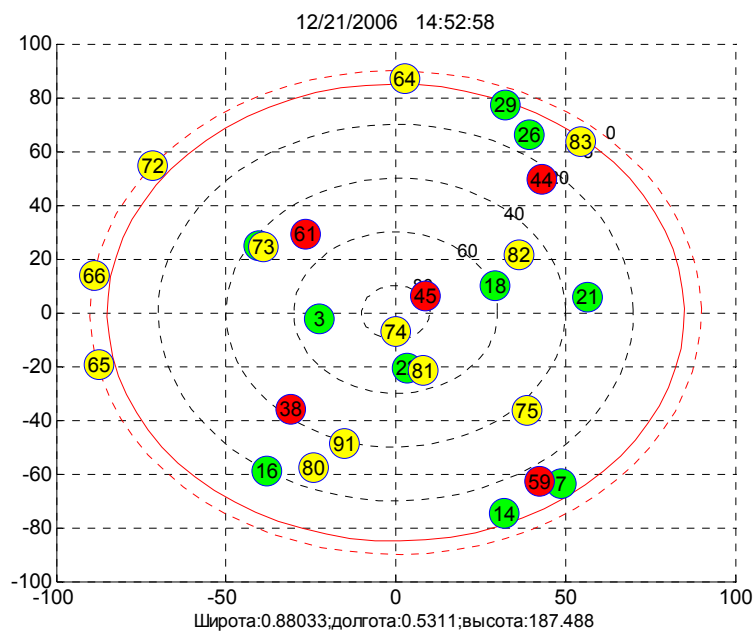


Рис. 5.5. Видимость спутников GPS, ГЛОНАСС, GALILEO

5.4.5 Лабораторная работа 5. 5 «Орбиты спутников GPS, ГЛОНАСС, GALILEO »

Цель лабораторной работы моделирование орбит спутников GPS, ГЛОНАСС, GALILEO

Для одновременного расчета орбит спутников трех систем GPS, ГЛОНАСС, GALILEO пакет программ, сформированный в лабораторной работе 5. 4 может быть дополнен файлами ORBITA_GGG.m и tar. m из папки **ORBIT_GGG**.

Рекомендуется следующий порядок выполнения лабораторной работы.

1. Создайте папку **ORBIT_GGG_My**.
2. Из папки **Vision_GLONASS_GPS_GALILEO** скопируйте в папку **ORBIT_GGG_My** функции ECEFLH, LLHECEF, Tim; файлы Yuma_GPS_GLONAS_Alm.m, AlmGGG.yum; из папки **ORBIT_GGG** файл ORBITA_GGG.m.
3. Запустите MatLab [7, 8].
4. Обратитесь к папке **ORBIT_GGG_My** , откройте ее, изучите функции, файлы и программные процедуры по комментариям и выполните задания **1, 2**.
5. **Задание 1.** Откройте файл ORBITA_GGG.m. Сформируйте желаемую конфигурацию спутников, орбиты которых требуется рассчитать (см. комментарий к файлу). Установите скорость вращения Земли, равную 0 ($\text{OMEGAeDOT}=0$). Выполните файл. Дайте описание, полученного графического изображения орбит. Результаты внесите в отчет.
6. **Задание 2.** В файле ORBITA_GGG.m сформируйте конфигурацию спутников: 1 спутник GPS, 1 спутник ГЛОНАСС, 1 спутник GALILEO. Установите

OMEGAeDOT=7.2921151467e-005. Выполните файл. Дайте описание, полученного графического изображения орбит. Результаты внесите в отчет.

5.4.6 Задания и вопросы для самоподготовки

1. Чем различаются орбиты спутников GPS, ГЛОНАСС, GALILEO?
2. По каким параметрам можно определить принадлежность спутника той или иной орбите?
3. Сформулируйте алгоритм определения дальности до спутника из точки наблюдения.
4. В какой системе координат рассчитываются углы видимости спутника?

5.4.7 Листинг файла **ORBITA_GGG.m**

Файл ORBITA_GGG.m

```
clear all
%Имя m-файла:ORBITA_GGG.m
%Программа рассчитывает орбиты спутников GPS, ГЛОНАСС, GALILEO
%Входные данные:
%файл альманаха в формате Yuma, имя файла альманаха присваивается
%переменной "Dat", например, Dat = 'имя файла альманаха';
%данные о начале отсчета "d2", d2='месяц/день/год'; h=час; min=минута; s=секунда;
%координаты позиции приемника -lat(широта в радианах), lon (долгота в радианах),
%hg (высота в метрах);
%шаг с каким будут рассчитываться параметры (step, секунды);
%количество точек (L), в которых будут рассчитываться параметры орбит
%L=12*3600/step, L читается так: количество часов (например, 12)
%число секунд в часе (3600) деленное на шаг (step)
%Выходные данные:
%Xk(j,i) - координата X спутника с номером i на момент времени j;
%Yk(j,i) - координата Y спутника с номером i на момент времени j;
%Zk(j,i) - координата Z спутника с номером i на момент времени j;
%Примечание: входные и выходные данные могут быть дополнены / изменены закомментированными
ми
%параметрами, помеченными строкой %%%%%%%%%%;
%Постоянные:
%скорость вращения Земли
%OMEGAeDOT=7.2921151467e-005;
OMEGAeDOT=0;%%%%%%%%%для орбитальных плоскостей
%радиусы земного эллипсоида
A_WGS84=6378137.0;
B_WGS84=6356752.314;
%константы
mu=3986005000000000;
```

```

F_CONST      = 4.442807633E-10;
%Задание цветов для графики
j_color = 0;
color6(1:11) = ['B' 'B' 'B' 'B' 'B' 'B' 'o' 'o' '+' '+' '+'];%%%%%%%%%
%Входные данные
Dat = 'AlmGGG.yum';%альманах спутников GPS, ГЛОНАСС, GALLILEO
N=6378136;% радиус Земли (используется, как нормирующий коэффициент
map(N);%функция выводит на графики Землю
d2='12/21/2006'; %%%%%%%%%%
h=14; min=52.0; s=58.0;%%%%%%%%%
%координаты точки, из которой наблюдаются спутники
lat = 0.88032730015257;%50 градусов 26минут 20.54 секунд
%%%%%%%%%
lon = 0.53109641675259;%30 градусов 25 минут 46.4995секунд
%%%%%%%%%
hr=187.488;% метров %%%%%%%%%%
step=600;%шаг отсчета времени в секундах (300=5 минутам);шаг можно изменять
%step=0;
L=(12*3600) / step;% убрать перед L для вывода таблицы улов видимости и азимута
%%%%%%%%%
%L=1;% установить % перед L для вывода таблицы улов видимости и азимута
%Чтение альманаха

[alm,max_kol] = Yuma_GPS_GLONAS_Alm(Dat);%%%%%%%%%
nom = 1;
i = 0;
k = 0;
i = 0;
while ( i <91)
    id = alm(nom).ID;
    Health = alm(nom).Health;
    if ( id > 0)
        Health = alm(nom).Health;
        if ( Health == 0)
            k = k + 1;
            nom_ns(k) = id;
            nom = nom + 1;
        else
            nom = nom + 1;
        end;
    else
end;
else

```

```

        nom = nom + 1;
    end;
    i = i + 1;
end; % i
kol = k;
nom_ns ; % - номер навигационного спутника
[Rx,Ry,Rz] = ECEFLH(lon, lat,hr);%%%%%%%%%%
[week,modeweek,d,dweek,weeks]=Tim(d2,h,min,s);
for j = 1:L
%   for j = 1:L % 0:L
        t(j)=weeks+step*(j); %-step;%%%%%%%%%
        %t1(j) = t(j)/60; %изменение дискретности текущего времени
        %d_wn = (week - alm(i).Week);
        %d_wn = 0;
kol=11;%%%%%%%%%
nom_ns(1:kol)=[1 3 4 5 9 10 38 58 62 72 82 ];%%%%%%%%%
    for k = 1 : kol
        i = nom_ns(k) ; % input "i" !!!
        d_wn=(modeweek - alm(i).Week);%если в альманахе не учтено 1024
        tk = t(j) + d_wn * 604800 - alm(i).TOA;
        d_wn = abs(modeweek - alm(i).Week);
        dd = 302400.0 + d_wn * 604800;
if ( ( alm(i).A05 > 0 ) & ( alm(i).Health == 0 ) )
        while (abs(tk) > dd)
            if tk > dd
                tk = tk - 604800;
            else
                if tk < -dd
                    tk = tk + 604800;
                end
            end % if
        end % while
%Справочник по альманаху- цифра в скобках обозначает порядковый номер
%параметра альманаха в формате YUMA
%alm(ID).ID(1); alm(ID).Health(2); alm(ID).e(3); alm(ID).TOA(4); alm(ID).deltai(5);
%alm(ID).OMEGADOT(6); alm(ID).A05(7); alm(ID).omega0(8); alm(ID).omega(9);
%alm(ID).M0(10); alm(ID).Af0(11); alm(ID).Af1(12); alm(ID).Week(13);
n0=sqrt((mu) / (alm(i).A05^6));
    j2 = 1082.68E-6;
    re = (A_WGS84 + B_WGS84) / 2.;
    sin55 = sin(55.0 * pi / 180.0);
    dn = 1.5 * j2 * re * re / (alm(i).A05^4 ) * (1. - 1.5 * sin55 * sin55);

```

```

%dn = 0;
n=n0 * (1 + dn);
Mk = alm(i).M0 + n*tk;
e=alm(i).e;
%решение уравнения Кеплера
eps = 1.0E-15;
y = e * sin(Mk);
x1 = Mk;
x = y;
for k = 0 : 15 % количество итераций
    x2 = x1;
    x1 = x;
    y1 = y;
    y = Mk - (x - e * sin(x));
    if (abs(y - y1) < eps)
        break
    end
    x = (x2 * y - x * y1) / (y - y1);
end % kepler
Ek = x;
deltr = F_CONST * alm(i).e * alm(i).A05 * sin(Ek);
dt1 = alm(i).Af0 + alm(i).Af1 * tk + deltr;
tk = tk - dt1;
vd = 1. - alm(i).e * cos(Ek);
nuk =atan2(sqrt(1-alm(i).e^2)*sin(Ek) / vd,(cos(Ek)-alm(i).e) / vd);
Ek = acos((alm(i).e+cos(nuk))/(1+alm(i).e*cos(nuk)));
Fk =nuk + alm(i).omega;
uk =Fk;
ik=alm(i).deltai;
rk =(alm(i).A05^2)*(1.0-alm(i).e*cos(Ek));
xkk =rk*cos(uk);
ykk =rk*sin(uk);
OMEGAk =alm(i).omega0+(alm(i).OMEGADOT-OMEGAeDOT)*tk-OMEGAeDOT*alm(i).TOA;
Xk(j,i) = xkk*cos(OMEGAk)-ykk*cos(ik)*sin(OMEGAk);
Yk(j,i) = xkk*sin(OMEGAk)+ykk*cos(ik)*cos(OMEGAk);
Zk(j,i) = ykk*sin(ik);
%%%%%%%%%%
%{
%Dальности до спутников
PR(j,i) = sqrt((Xk(j,i) - Rx)^2 + (Yk(j,i) - Ry)^2 + (Zk(j,i) - Rz)^2);
%Перевод в географическую систему если требуется
%[lons,lats,hrs] = LLHECEF(Xk,Yk,Zk);

```

```

%(Llon(j),Llat(j),Hhr(j)) = [lons,lats,hrs];
    %расчет угла видимости спутника
xls = Xk(j,i) - Rx;
    yls = Yk(j,i) - Ry;
    zls = Zk(j,i) - Rz;
range1 = sqrt(xls*xls+yls*yls+zls*zls);
if j>1
    doppler(j-1) = (range1 - range2) * 5.2514 / step;
end
range2 = range1;
P = sqrt(Rx * Rx + Ry * Ry + Rz * Rz);
tdot = ( Rx*xls+Ry*yls+Rz*zls)/range1/P;

xll = xls/range1;
    yll = yls/range1;
    zll = zls/range1;

if tdot >= 1.00
    b = 0.0;
elseif tdot <= -1.00
    b = pi;
else
    b = acos( tdot);
end
satang = pi/2.0 - b;
TT =satang;
TT(j,i) =TT;%угол видимости спутников
    %расчет угла азимута спутников
xn =-cos(lon)*sin(lat);
yn =-sin(lon)*sin(lat);
zn = cos(lat);
xe =-sin(lon);
ye = cos(lon);
xaz = xe*xll + ye*yll;
yaz = xn*xll + yn*yll + zn*zll;
if (xaz == 0) or (yaz == 0)
    az(j)= 0;
else
az(j,i) = atan2(xaz,yaz);
end
if az(j,i) < 0
    az(j,i) = az(j,i) + pi*2;

```



```

end

AZ(j,i) =az(j,i) *180/pi;%угла азимута спутников в градусах
EL(j,i) = TT(j,i) *180/pi;%угла видимости спутников в градусах

% ПЕРЕСЧЕТ ВРЕМЕНИ
A(j)=mod(t(j),86400);
her(j)=floor(A(j)/3600);
m(j)=floor((A(j)-her(j)*3600)/60);
sek(j)=A(j)-her(j)*3600-m(j)*60;

%Построение полярной системы координат
if EL(j,i) < 0
    elp = 180;
else
    elp = (EL(j,i) - 90);
end;
azp = (AZ(j,i) + 90.0);
rad = pi / 180;
x0 = 0; y0 = 0;
xt(j,i) = (elp * cos(azp * rad));
yt(j,i) = -(elp * sin(azp * rad));
%}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end % i = ns
end; % if ( alm(i).A05 > 0 )
end % j = time
for (i=1:kol)
j_color = j_color + 1;
if (j_color > 11 )
    j_color = 1;
end
S = color6(j_color);
prn = nom_ns(i);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
hold on
h_F1 = gca;
plot3(Xk(:,prn),Yk(:,prn),Zk(:,prn),S,'LineWidth',0.5);
axis([-2.552*10^(7) 2.552*10^(7) -2.552*10^(7) 2.552*10^(7) -2.552*10^(7) 2.552*10^(7)]);
set(get(gcf,'CurrentAxes'),'FontSize',14,'FontName','TimesNewRoman');
set(h_F1,'Position',[0.1 0.1 0.85 0.9]) ;
xlabel('Координата X')

```

```

ylabel('Координата Y'),
zlabel('Координата Z'),grid on
str1 = num2str( prn);
text(Xk(j,prn),
Yk(j,prn),Zk(j,prn),str1,'FontSize',14,'FontName','TimesNewRoman','HorizontalAlignment','center' );
hold on
%%%%%%%%%%%%%%
end
clear

```

5.4.8 Пример выполнения файла:ORBITA_GGG.m

Результат выполнения программы изображен на рис. 5.6.

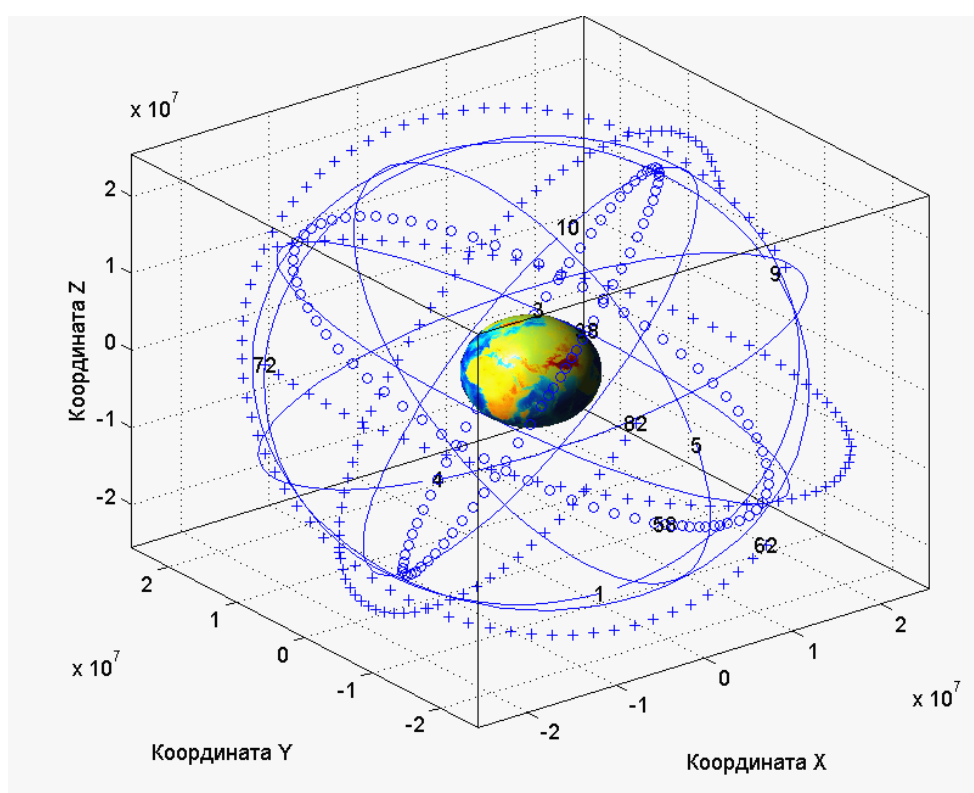


Рис. 5.6. Орбиты спутников GPS, ГЛОНАСС, GALILEO

На рис. 5.6 сплошными линиями показаны 6 орбит спутников GPS, маркером «o» 2 орбиты ГЛОНАСС, маркером «+» 3 орбиты GALILEO; цифрами обозначены номера спутников соответствующих систем.

5.5 Декодирование и расшифровка данных спутников ГЛОНАСС

5.5.1 Краткие сведения из теории

В данном подразделе приводится программа декодирования данных спутников ГЛОНАСС. Данные, которые декодируются получены экспериментально с реального на-

вигационного приемника после корреляционной обработки в виде последовательности символов «0» и «1». Поскольку объем данных относительно большой около 60000 символов, то приводятся два вспомогательных m-файла. Эти вспомогательные программы позволяют сформировать данные на бумажном носителе в виде последовательности шестнадцатеричных чисел, из которых можно восстановить m-файл для декодирования. Экспериментальные данные рассматриваемые далее представляют собой символы длительностью 10 миллисекунд и являются составляющими информационных символов сложенных по модулю 2 с меандром имеющим частоту 100 Гц.

Примененный алгоритм декодирования основан на строгом соответствии данных, передаваемых с навигационного спутника ГЛОНАСС, которые соответствуют формату сообщений. Более подробно формат сообщений и данные, передаваемые в кадрах сообщений приводятся в интерфейсном контрольном документе ГЛОНАСС [] и в книге [1] (подраздел 2. 2, стр. 77- 87).

Комплекс программ декодирования и расшифровки находится в папке **Decod_GLONASS**.

Цель лабораторной работы освоение и применение методики декодирования и расшифровки данных спутниковой системы ГЛОНАСС.

5.5.2 Лабораторная работа 5. 6 «Декодирование данных спутников ГЛОНАСС в навигационном приемнике»

Рекомендуется следующий порядок выполнения лабораторной работы.

1. Создайте папку **Decod_GLONASS_My** и скопируете в ее файлы и функции из папки **Decod_GLONASS**.
2. Запустите MatLab.
3. Обратитесь к папке **Decod_GLONASS_My** , откройте ее, изучите функции, файлы и программные процедуры по комментариям и выполните задания **1, 2**.
4. **Задание 1.** Откройте файл Hex_GLONASS.m. Изучите программные процедуры файла, используя его текст и расширенный комментарий.

Файл преобразования экспериментальных данных в шестнадцатеричные символы Hex_GLONASS.m

```
%Имя m- файла: Hex_GLONASS.m
%Файл Hex_GLONASS.m предназначен для преобразования экспериментальных данных, записанных
%в файле с именем raw_bits_SV18J.dat в файл SV1.dat, записанный в шестнадцатеричных символах
name_in = 'raw_bits_SV18J.dat'; % имя экспериментального файла
fw = fopen('SV1.dat','wt');%имя файла в шестнадцатеричных символах
M=dlmread(name_in);%чтение экспериментального файла в матрицу M
%чтение двоичных символов в 1875 строк по 16 символов в строке
```


го декодирования. Для этого скопируем приведенное шестнадцатеричное представление файла и запишем его в виде файла с именем FilGL1.txt с помощью текстового редактора. В форму для декодирования (эта форма в точности соответствует файлу raw_bits_SV18J.dat файл FilGL1.txt преобразуется с помощью программы HexBin_GLONASS.m.

Файл HexBin_GLONASS.m

```
%Имя m- файла: HexBin_GLONASS.m
%Файл HexBin_GLONASS предназначен для преобразования файла FilGL1.txt
%в шестнадцатеричных символах в файл FilGL1_bin.txt в символах "0" и "1"
name_in1='FilGL1.txt';%имя файла в шестнадцатеричном представлении
fid = fopen(name_in1,'rt'); %открытие файла
fw_bin=fopen('FilGL1_bin.txt','wb');%запись преобразованного файла
%чтение и переформатирование файла
MAS = fscanf(fid,'%s');
for i=1:1875
    str_MAS(1:4,i)=sprintf('%s',MAS(i+3*(i-1):(3+i+3*(i-1))));
end
MATR=reshape(MAS,4,1875);
hex_dec= base2dec(MATR,16);%преобразование шестнадцатеричных чисел в десятичные
dec_bin= dec2bin(hex_dec);%преобразование десятичных чисел в двоичные
MATR_Char = reshape(dec_bin,1,1875*16);%преобразование в строку
%fprintf(fw_bin,'%c ',MATR_Char);%запись файла в символах "0" и "1"(вариант, пробел после c имеет значение)
fprintf(fw_bin,'%c\n',MATR_Char);%запись файла в символах "0" и "1" (вариант)
fclose(fw_bin);% закрытие файла
```

Файлы FilGL1.txt (восстановленный из шестнадцатеричного) и raw_bits_SV18J.dat (экспериментальный) одинаковы и применяя любой из них можно проводить декодирование данных спутников ГЛОНАСС.

Прежде чем приступить к обработке данных файла FilGL1.txt рассмотрим процесс формирования и декодирования в навигационном приемнике информационного навигационного сигнала. Этот процесс может быть представлен в виде следующей блок – схемы (рис. 5.7).

Информация, передаваемая с каждого навигационного спутника представляет собой непрерывный поток информационных символов. Начала отсчетов передачи данных нуль часов Московского времени (начало передачи первого суперкадра каждые сутки).

Структурно поток цифровой информации (ЦИ) формируется в виде непрерывно повторяющихся суперкадров (рис. 5.8).

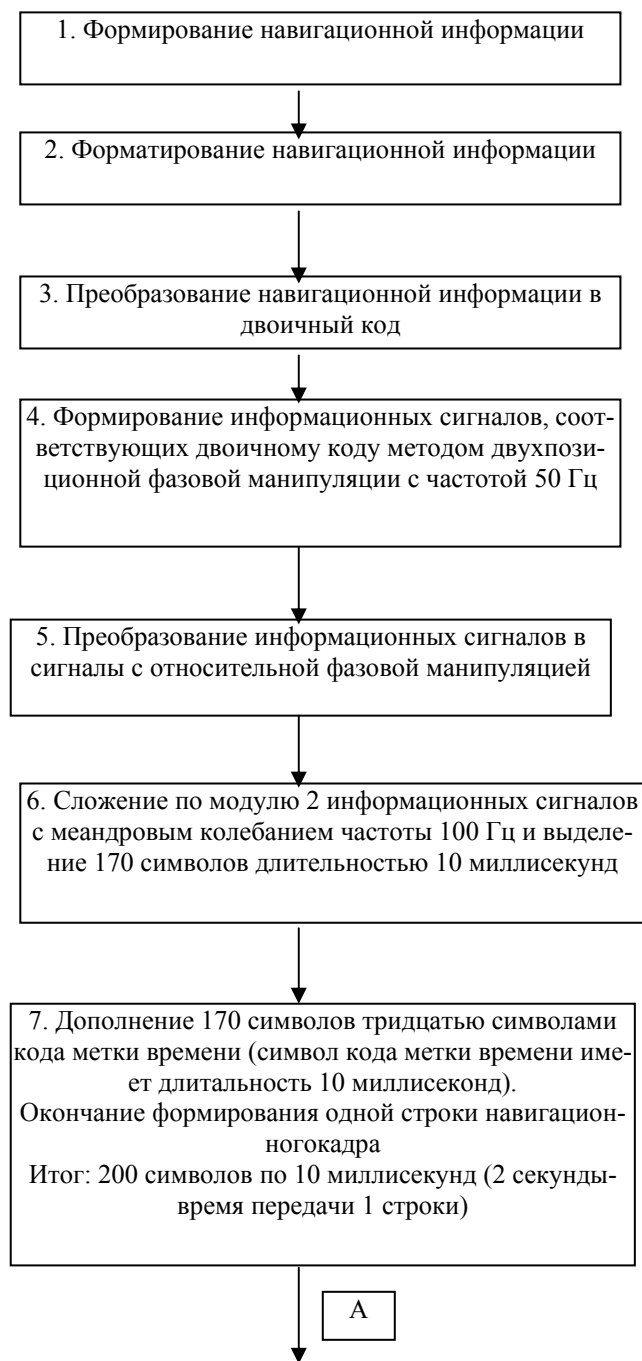


Рис. 5.7. Формирование и декодирование радионавигационного сигнала

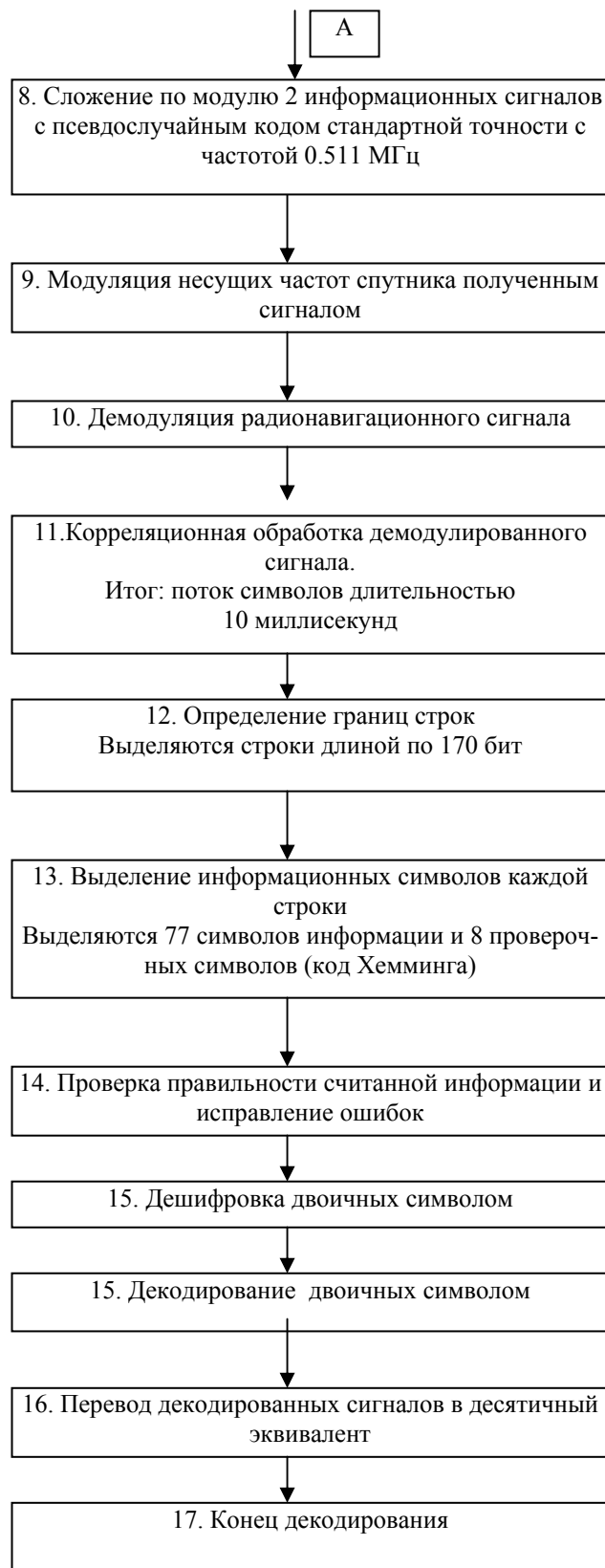


Рис. 5.7. Формирование и декодирование радионавигационного сигнала (продолжение)

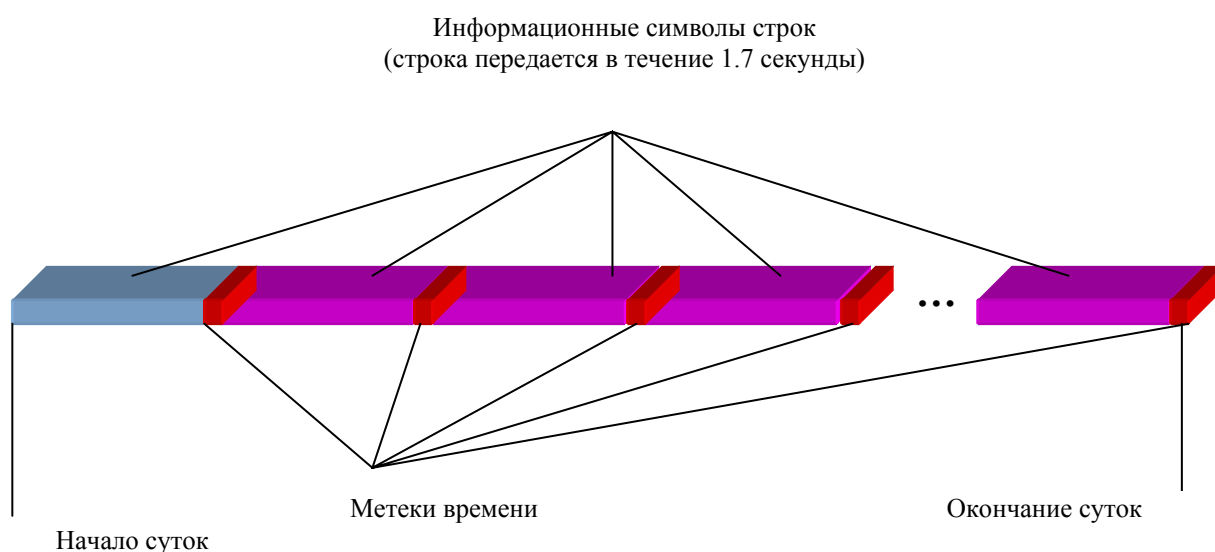


Рис. 5.8. Изображение потока данных навигационного спутника

В приведенной далее программе декодирования навигационных данных спутников ГЛОНАСС выполняются процедуры п. п. 12... 17 (рис. 5.7) по данным п. 11 в соответствии с ИКД ГЛОНАСС.

5. **Задание 2.** Откройте файл *Decod_GL3.m*. Изучите программные процедуры файла, используя его текст и расширенный комментарий. Выполните файл и в командном окне MatLab прочитайте результат декодирования и занесите его в отчет.

5.5.3 Задания и вопросы для самоподготовки

1. Изучите форматы сообщений спутников ГЛОНАСС?
2. Что понимается под суперкадром, кадром, строкой и словом в сообщениях спутников ГЛОНАСС?
3. Что понимается под оперативной и неоперативной информацией, передаваемой со спутников ГЛОНАСС?
4. Как контролируется достоверность информации, передаваемой со спутников ГЛОНАСС?
5. Как выполняется преобразование между системами счисления: переход из двоичной системы счисления в десятичную и наоборот; переход из шестнадцатеричной системы счисления в десятичную и наоборот; переход из двоичной системы счисления в шестнадцатеричную и наоборот?
6. Что такой бит, байт, младший разряд, старший разряд?

5.5.4 Листинг файла:Decod_GL3.m

```
%Имя m- файла:Decod_GL3.m
%Программа декодирования данных спутников ГЛОНАСС
%Символы метки времени
Met=[1 1 1 1 1 0 0 0 1 1 0 1 1 1 0 1 0 1 0 0 0 0 1 0 0 1 0 1 1 0];
%Открытие файла для считывания данных
%fid = fopen('raw_bits_SV18J.dat','rb')
%fid = fopen('SV18fw_bin.txt','rb')
fid = fopen('FilGL1_bin.txt','rb')
if fid~-=-1
kol = 0;%Порядковый номер считываемых данных
while (kol < 24)
%Поиск и обнаружение меток времени в массиве данных
for i = 1:30
MAS(i) = fscanf(fid,'%d',1);
end
C = xor(MAS, Met);
D = any(C);
kk = 0;

while ( (feof(fid)==0) & (D == 1) )
MAS(1:29) = MAS(2:30);
ch = fscanf(fid,'%d',1);
MAS(30) = ch;
C = xor(MAS, Met);
D = any(C); %0 - all 0
kk = kk + 1;
end % while
%Считывание данных строк
kol = kol + 1

for i = 1:170
danDDD(i) = (fscanf(fid,'%d',1));
end;
str_danDDD=sprintf('%d',danDDD(1:170));

%Сложение данных по модулю 2 с меандром
for i = 1:170
meandr = rem((i-1), 2); %остаток от деления на 2 ((i-1)/2)
danDD(i) = xor(danDDD(i), meandr);

end; % for i = 1:170
```

```

% DD = danDD
%Выделение информационных символов
dd1 = danDD(1:2:169);
dd2 = danDD(2:2:170);
ddxor = xor(dd1, dd2);
Dany = any(ddxor);
str1(1) = dd1(1);
%Перевод данных из относительного кода
for i = 2:85
    str1(i) = xor(dd1(i-1), dd1(i));
end;
% Контроля четности (правильности) принятых данных

Skod(1,1:85)=[0 0 0 0 0 0 0 0 1 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1 0 1 ...
0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 ...
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 1 1 0 1 0 1 0 1 ...
0 1 0 1 0 1 0 1 0 1 0 1 0];

Skod(2,1:85)=[0 0 0 0 0 0 0 0 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 0 1 1 ...
0 0 1 1 0 0 1 1 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 ...
0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 1 1 0 0 1 1 ...
0 0 1 1 0 0 1 1 0 0 1 1 0];

Skod(3,1:85)=[0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 ...
0 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 ...
0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 1 1 1 1 ...
0 0 0 0 1 1 1 1 0 0 0 0 1];

Skod(4,1:85)=[0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 ...
1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 ...
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 ...
1 1 1 1 1 1 1 1 0 0 0 0 0];

Skod(5,1:85)=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 ...
1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 ...
0 0 0 0 0 0 0 0 1 1 1 1 0];

Skod(6,1:85)=[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ...
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 ...
0 0 0 0 0 0 0 0 0 0 0 0 0];

```



```

% в кадре [например, num(kol) == 1- строка номер 1]
if num(kol) == 1
    zx=1; zxdot=1; zx2dot=1;
    Nstr=sprintf('%d',str1(2:5));
    R1= sprintf('%d',str1(6:7));
    P1=sprintf('%d',str1(8:9));
    tk5=sprintf('%d',str1(10:14));
    tk6=sprintf('%d',str1(15:20));
    tk1=sprintf('%d',str1(21));
    if sprintf('%d',str1(22))=='1'
        zxdot = -1;
    end;
    Xdot=sprintf('%d',str1(23:45));
    if sprintf('%d',str1(46))=='1'
        zx2dot = -1;
    end;
    X2dot=sprintf('%d',str1(47:50));
    if sprintf('%d',str1(51))=='1'
        zx = -1;
    end;
    X=sprintf('%d',str1(52:77));

    St1=bin2dec({Nstr tk5 tk6 tk1 X Xdot X2dot});

    Mng=[1 1 1 30 zx*2^(-11) zxdot*2^(-20) zx2dot*2^(-30)];
    Str1=(St1'.*Mng)'%вывод декодированных данных
end;

if num(kol) == 2
    zy=1; zydote=1; zy2dote=1;
    Nstr=sprintf('%d',str1(2:5));
    Bn= sprintf('%d',str1(6:8));
    P2=sprintf('%d',str1(9));
    tb=sprintf('%d',str1(10:16));
    R2=sprintf('%d',str1(17:21));
    if sprintf('%d',str1(22))=='1'
        zydote = -1;
    end;
    Ydote=sprintf('%d',str1(23:45));
    if sprintf('%d',str1(46))=='1'
        zy2dote = -1;
    end;
end;

```

```

Y2dot=sprintf('%d',str1(47:50));
if sprintf('%d',str1(51))=='1'
    zy = -1;
end;
Y=sprintf('%d',str1(52:77));

St2=bin2dec({Nstr Bn P2 tb Y Ydot Y2dot});

Mng=[1 1 1 15*60 zy*2^(-11) zydots*2^(-20) zy2dots*2^(-30)];
Str2=(St2.*Mng)%вывод декодированных данных
end;

if num(kol) == 3
    zz=1; zzdots=1; zz2dots=1; zgamma=1;
    Nstr=sprintf('%d',str1(2:5));
    P3= sprintf('%d',str1(6));
    if sprintf('%d',str1(7))=='1'
        zgamma = -1;
    end;
    Gamma=sprintf('%d',str1(8:17));
    R3=sprintf('%d',str1(18:19));
    P=sprintf('%d',str1(20));
    ln=sprintf('%d',str1(21));
    if sprintf('%d',str1(22))=='1'
        zzdots = -1;
    end;
    Zdots=sprintf('%d',str1(23:45));
    if sprintf('%d',str1(46))=='1'
        zz2dots = -1;
    end;
    Z2dots=sprintf('%d',str1(47:50));
    if sprintf('%d',str1(51))=='1'
        zz = -1;
    end;
    Z=sprintf('%d',str1(52:77));

St3=bin2dec({Nstr P3 ln Gamma Z Zdots Z2dots});

Mng=[1 1 1 zgamma*2^(-40) zz*2^(-11) zzdots*2^(-20) zz2dots*2^(-30)];
Str3=(St3.*Mng)%вывод декодированных данных
end;

```

```

if num(kol) == 4
    ztaun=1; zdtaun=1;
    Nstr=sprintf('%d',str1(2:5));
    if sprintf('%d',str1(6))=='1'
        ztaun = -1;
    end;
    taun= sprintf('%d',str1(7:27));
    if sprintf('%d',str1(28))=='1'
        zdtaun = -1;
    end;
    deltatau=sprintf('%d',str1(29:32));
    En=sprintf('%d',str1(33:37));
    R4=sprintf('%d',str1(38:51));
    P4=sprintf('%d',str1(52));
    FT=sprintf('%d',str1(53:56));
    R4i4=sprintf('%d',str1(57:59));
    NT=sprintf('%d',str1(60:70));
    nSV=sprintf('%d',str1(71:75));
    modifSV=sprintf('%d',str1(76:77));
    St4=bin2dec({Nstr taun deltatau En FT NT nSV modifSV});
    Mng=[1 ztaun*2^(-30) zdtaun*2^(-30) 1 1 1 1 1];
    Str4=(St4.*Mng)%вывод декодированных данных
end;

```

```

if num(kol) == 5
    ztauc=1; ztauGPS=1;
    Nstr=sprintf('%d',str1(2:5));
    NA= sprintf('%d',str1(6:16));
    if sprintf('%d',str1(17))=='1'
        ztauc = -1;
    end;
    tauc=sprintf('%d',str1(18:48));
    R5=sprintf('%d',str1(49));
    N4=sprintf('%d',str1(50:54));
    if sprintf('%d',str1(55))=='1'
        ztauGPS = -1;
    end;

    tauGPS=sprintf('%d',str1(56:76));
    In=sprintf('%d',str1(77));
    St5=bin2dec({Nstr NA tauc N4 tauGPS In Nstr Nstr});
    Mng=[1 1 ztauc*2^(-27) 1 ztauGPS*2^(-30) 1 0 0];

```

```

Str5=(St5'.*Mng)%вывод декодированных данных
end;

if num(kol) == 6
    ztaun=1; zlambda=1;zdeltai=1;
    Nstr=sprintf('%d',str1(2:5));
    CNA= sprintf('%d',str1(6));
    Mn=sprintf('%d',str1(7:8));
    nSV=sprintf('%d',str1(9:13));
    if sprintf('%d',str1(14))=='1'
        ztaun = -1;
    end;
    taun=sprintf('%d',str1(15:23));
    if sprintf('%d',str1(24))=='1'
        zlambda = -1;
    end;
    lambda=sprintf('%d',str1(25:44));
    if sprintf('%d',str1(45))=='1'
        zdeltai= -1;
    end;
    deltai=sprintf('%d',str1(46:62));
    ecs=sprintf('%d',str1(63:77));
    St6=bin2dec({Nstr Mn nSV taun lambda deltai ecs});
    Mng=[1 1 1 ztaun*2^(-18) zlambda*2^(-20) zdeltai*2^(-20) 2^(-20)];
    Str6=(St6'.*Mng)%вывод декодированных данных
end;

if num(kol) == 7
    zomegan=1;zdeltaT=1; zdeltaTdot=1;
    Nstr=sprintf('%d',str1(2:5));
    if sprintf('%d',str1(6))=='1'
        zomegan = -1;
    end;
    omegan= sprintf('%d',str1(7:21));
    tlambd=sprintf('%d',str1(22:44));
    if sprintf('%d',str1(43))=='1'
        zdeltaT = -1;
    end;
    deltaT=sprintf('%d',str1(44:64));
    if sprintf('%d',str1(65))=='1'
        zdeltaTdot= -1;
    end;
end;

```



```

deltaTdot=sprintf('%d',str1(66:71));
Hn=sprintf('%d',str1(72:76));
ln=sprintf('%d',str1(77));
St7=bin2dec({Nstr omegan tlambda deltaT deltaTdot Hn ln});
Mng=[1 zomegan*2^(-15) 2^(-5) zdeltaT*2^(-9) zdeltaTdot*2^(-14) 1 1];
Str7=(St7.*Mng)%вывод декодированных данных
end;

if num(kol) == 8
    ztaun=1; zlambda=1;zdeltai=1;
    Nstr=sprintf('%d',str1(2:5));
    CNA= sprintf('%d',str1(6));
    Mn=sprintf('%d',str1(7:8));
    nSV=sprintf('%d',str1(9:13));
    if sprintf('%d',str1(14))=='1'
        ztaun = -1;
    end;
    taun=sprintf('%d',str1(15:23));
    if sprintf('%d',str1(24))=='1'
        zlambda = -1;
    end;
    lambda=sprintf('%d',str1(25:44));
    if sprintf('%d',str1(45))=='1'
        zdeltai= -1;
    end;
    deltai=sprintf('%d',str1(46:62));
    ecs=sprintf('%d',str1(63:77));
    St8=bin2dec({Nstr Mn nSV taun lambda deltai ecs});
    Mng=[1 1 1 ztaun*2^(-18) zlambda*2^(-20) zdeltai*2^(-20) 2^(-20)];
    Str8=(St8.*Mng)' %вывод декодированных данных
end;

if num(kol) == 9
    zomegan=1;zdeltaT=1; zdeltaTdot=1;
    Nstr=sprintf('%d',str1(2:5));
    if sprintf('%d',str1(6))=='1'
        zomegan = -1;
    end;
    omegan= sprintf('%d',str1(7:21));
    tlambda=sprintf('%d',str1(22:44));
    if sprintf('%d',str1(43))=='1'
        zdeltaT = -1;

```

```

end;
deltaT=sprintf('%d',str1(44:64));
if sprintf('%d',str1(65))=='1'
    zdeltaTdot= -1;
end;
deltaTdot=sprintf('%d',str1(66:71));
Hn=sprintf('%d',str1(72:76));
ln=sprintf('%d',str1(77));
St9=bin2dec({Nstr omegan tlambda deltaT deltaTdot Hn ln});
Mng=[1 zomegan*2^(-15) 2^(-5) zdeltaT*2^(-9) zdeltaTdot*2^(-14) 1 1];
Str9=(St9.*Mng)%вывод декодированных данных
end;

if num(kol) == 10
    ztaun=1; zlambda=1;zdeltai=1;
    Nstr=sprintf('%d',str1(2:5));
    CNA= sprintf('%d',str1(6));
    Mn=sprintf('%d',str1(7:8));
    nSV=sprintf('%d',str1(9:13));
    if sprintf('%d',str1(14))=='1'
        ztaun = -1;
    end;
    taun=sprintf('%d',str1(15:23));
    if sprintf('%d',str1(24))=='1'
        zlambda = -1;
    end;
    lambda=sprintf('%d',str1(25:44));
    if sprintf('%d',str1(45))=='1'
        zdeltai= -1;
    end;
    deltai=sprintf('%d',str1(46:62));
    ecs=sprintf('%d',str1(63:77));
    St10=bin2dec({Nstr Mn nSV taun lambda deltai ecs});
    Mng=[1 1 1 ztaun*2^(-18) zlambda*2^(-20) zdeltai*2^(-20) 2^(-20)];
    Str10=(St10.*Mng)%вывод декодированных данных
end;

if num(kol) == 11
    zomegan=1;zdeltaT=1; zdeltaTdot=1;
    Nstr=sprintf('%d',str1(2:5));
    if sprintf('%d',str1(6))=='1'
        zomegan = -1;

```

```

end;
omegan= sprintf('%d',str1(7:21));
tlambda=sprintf('%d',str1(22:44));
if sprintf('%d',str1(43))=='1'
    zdeltaT = -1;
end;
deltaT=sprintf('%d',str1(44:64));
if sprintf('%d',str1(65))=='1'
    zdeltaTdot= -1;
end;
deltaTdot=sprintf('%d',str1(66:71));
Hn=sprintf('%d',str1(72:76));
ln=sprintf('%d',str1(77));
St11=bin2dec({Nstr omegan tlambda deltaT deltaTdot Hn ln});
Mng=[1 zomegan*2^(-15) 2^(-5) zdeltaT*2^(-9) zdeltaTdot*2^(-14) 1 1];
Str11=(St11'.*Mng)%вывод декодированных данных
end;

if num(kol) == 12
    ztaun=1; zlambda=1;zdeltai=1;
    Nstr=sprintf('%d',str1(2:5));
    CNA= sprintf('%d',str1(6));
    Mn=sprintf('%d',str1(7:8));
    nSV=sprintf('%d',str1(9:13));
    if sprintf('%d',str1(14))=='1'
        ztaun = -1;
    end;
    taun=sprintf('%d',str1(15:23));
    if sprintf('%d',str1(24))=='1'
        zlambda = -1;
    end;
    lambda=sprintf('%d',str1(25:44));
    if sprintf('%d',str1(45))=='1'
        zdeltai= -1;
    end;
    deltai=sprintf('%d',str1(46:62));
    ecs=sprintf('%d',str1(63:77));
    St12=bin2dec({Nstr Mn nSV taun lambda deltai ecs});
    Mng=[1 1 1 ztaun*2^(-18) zlambda*2^(-20) zdeltai*2^(-20) 2^(-20)];
    Str12=(St12'.*Mng)%вывод декодированных данных
end;

```

```

if num(kol) == 13
    zomegan=1;zdeltaT=1; zdeltaTdot=1;
    Nstr=sprintf('%d',str1(2:5));
    if sprintf('%d',str1(6))=='1'
        zomegan = -1;
    end;
    omegan= sprintf('%d',str1(7:21));
    tlambd=sprintf('%d',str1(22:44));
    if sprintf('%d',str1(43))=='1'
        zdeltaT = -1;
    end;
    deltaT=sprintf('%d',str1(44:64));
    if sprintf('%d',str1(65))=='1'
        zdeltaTdot= -1;
    end;
    deltaTdot=sprintf('%d',str1(66:71));
    Hn=sprintf('%d',str1(72:76));
    ln=sprintf('%d',str1(77));
    St13=bin2dec({Nstr omegan tlambd deltaT deltaTdot Hn ln});
    Mng=[1 zomegan*2^(-15) 2^(-5) zdeltaT*2^(-9) zdeltaTdot*2^(-14) 1 1];
    Str13=(St13'.*Mng)%вывод декодированных данных
end;

if num(kol) == 14
    ztaun=1; zlambda=1;zdeltai=1;
    Nstr=sprintf('%d',str1(2:5));
    CNA= sprintf('%d',str1(6));
    Mn=sprintf('%d',str1(7:8));
    nSV=sprintf('%d',str1(9:13));
    if sprintf('%d',str1(14))=='1'
        ztaun = -1;
    end;
    taun=sprintf('%d',str1(15:23));
    if sprintf('%d',str1(24))=='1'
        zlambda = -1;
    end;
    lambda=sprintf('%d',str1(25:44));
    if sprintf('%d',str1(45))=='1'
        zdeltai= -1;
    end;
    deltai=sprintf('%d',str1(46:62));
    ecs=sprintf('%d',str1(63:77));

```

```

St14=bin2dec({Nstr Mn nSV taun lambda delta i ecs});
Mng=[1 1 1 ztaun*2^(-18) zlambda*2^(-20) zdeltai*2^(-20) 2^(-20)];
Str14=(St14'.*Mng)%вывод декодированных данных
end;

%Строка 14 для 5 кадра
if num(kol) == 14
    zB1=1; zB2=1;
    Nstr=sprintf('%d',str1(2:5));

    if sprintf('%d',str1(6))=='1'
        zB1 = -1;
    end;
    B1=sprintf('%d',str1(7:16));
    if sprintf('%d',str1(17))=='1'
        zB2 = -1;
    end;
    B2=sprintf('%d',str1(18:26));
    Kp=sprintf('%d',str1(27:28));
    St14_5=bin2dec({Nstr B1 B2 Kp});
    Mng=[1 zB1*2^(-10) zB2*2^(-16) 1];
    Str14_5=(St14_5'.*Mng)%вывод декодированных данных
end;

if num(kol) == 15
    zomegan=1;zdeltaT=1; zdeltaTdot=1;
    Nstr=sprintf('%d',str1(2:5));
    if sprintf('%d',str1(6))=='1'
        zomegan = -1;
    end;
    omegan= sprintf('%d',str1(7:21));
    tlambd=sprintf('%d',str1(22:44));
    if sprintf('%d',str1(43))=='1'
        zdeltaT = -1;
    end;
    deltaT=sprintf('%d',str1(44:64));
    if sprintf('%d',str1(65))=='1'
        zdeltaTdot= -1;
    end;
    deltaTdot=sprintf('%d',str1(66:71));
    Hn=sprintf('%d',str1(72:76));
    ln=sprintf('%d',str1(77));

```

```

Str15=bin2dec({Nstr omegan tlambda deltaT deltaTdot Hn ln});
Mng=[1 zomegan*2^(-15) 2^(-5) zdeltaT*2^(-9) zdeltaTdot*2^(-14) 1 1];
Str15=(Str15'.*Mng)%вывод декодированных данных
end;

str1;

end;%while (feof(fid)==0) | % while (kol < 14)
end % if fid~-1
fclose(fid);
%Технологические данные
Qout=[num' cc_all] %Номер строки и проверка на четность
%Qoutnum1=Qout(1,2:9);
%SS1=sprintf('%d',str1);
%Qout123=[Str1 Str2 Str3]
%Qout45=[Str4 Str5]
%Qout6789=[Str6 Str7 Str8 Str9]
%Qout10_11_12=[Str10 Str11 Str12]
%Qout13_14_15=[Str13 Str14 Str15]
clear;

```

5.5.5 Пример выполнения файла:Decod_GL3.m

В результате выполнения программы в командном окне Command Window MatLab отображаются декодированные данные по каждой из строк в последовательности и соответствии ИКД ГЛОНАСС. Так, например, структура строки 1согласно ИКД ГЛОНАСС имеет вид

1	2 - 5	6 - 7	8 - 9	10 - 21	22 - 45
85	84 - 81	80 - 79	78 - 77	76 - 65	64 - 41
0	<i>m</i>		<i>PI</i>	<i>t_k</i>	$\dot{X}_n(t_b)$
Номер строки в навигационном кадре	Резерв		Признак смены оперативной информации	Время начала кадра внутри текущих суток, исчисляемое в шкале бортового времени. Начало суток по бортовому времени спутника совпадает с началом очередного суперкадра.	Составляющие вектора скорости n-го спутника в системе координат ПЗ-90 на момент времени t_b .

46 - 50	51 - 77	1 - 8	1 - 30
40 - 36	35 - 9	8 - 1	1 - 30
$\ddot{X}_n(t_b)$	$X_n(t_b)$	Код Хемминга	Метка времени
Составляющие ускорения n-го спутника на момент времени t_b .	Координаты n-го спутника в системе координат ПЗ-90 на момент времени t_b .		

Рис. 5.9. - Структура 1 строки для (1- 4) кадров суперкадра

Декодированная информация из командного окна MatLab приведена на рис. 5.10.

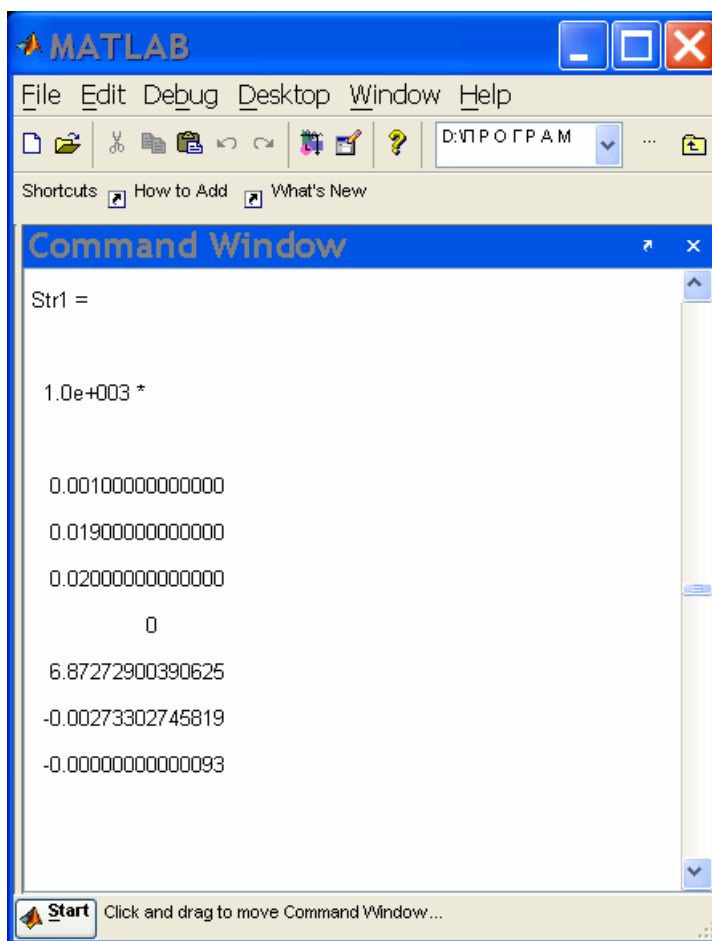


Рис. 5.10 Декодированные данные спутника ГЛОНАСС

Данные, изображенные на рис. 5.10 соответствуют рис. 5.9, но записаны в столбец.

РАЗДЕЛ 6 Решение навигационной задачи

6.1 Краткие сведения из теории

Главной задачей спутникового навигационного приемника является определения координат. Для решения этой задачи требуется обнаружить сигналы спутников, перейти в режим слежения за сигналами, принять и декодировать данные, поступающие в сообщениях спутников, измерить расстояния до спутников, обработать всю информацию и решить навигационную задачу.

Один из алгоритмов определения координат приемника потребителя изложен в книге [1].

В данном подразделе приводится m- файл positionV0.m, решающий навигационную задачу. На CD- диске файл расположен в папке «Координаты приемника». Комментарий к программным процедурам дается в тексте программы.

Цель лабораторной работы: изучение метода расчета координат потребителя по данным, полученным с навигационных спутников GPS, ГЛОНАСС и измеренным псевдодальностям

6.2 Лабораторная работа 4. 1 «Решение навигационной задачи»

Рекомендуется следующий порядок выполнения лабораторной работы.

1. Создайте папку **Координаты приемника_My** запишите в ее файл из папки **Координаты приемника**.
2. Откройте файл positionV0.m, изучите программные процедуры и комментарии и выполните задания 1- 2.
3. **Задание 1.** Последовательно выполните файл positionV0.m для расчета координат при участии в расчетах 15, 14, 13, 9, 8, 7, 6 спутников из разных созвездий. Проанализируйте полученные результаты. Расчеты позиции и результаты анализа запишите в отчет.
4. **Задание 2.** Добавьте к псевдодальностям до спутников, которые используются для решения навигационной задачи одинаковое приращение. Исполните файл. Проанализируйте и объясните полученный результат. Данные занесите в отчет.

6.3 Вопросы и задания для самоподготовки

1. Какие входные данные требуются для решения навигационной задачи?
2. Какое минимальное количество навигационных спутников одного созвездия требуется для решения навигационной задачи?.

3. Какое минимальное количество навигационных спутников двух созвездий требуется для решения навигационной задачи?
4. Каким образом определяется псевдодальность до навигационного спутника?
5. Почему при увеличении псевдодальностей до спутников, принимающих участие в навигационной задаче, на одну и ту же величину позиция координат не изменяется?
6. Как определяется пространственный геометрический фактор (*PDOP*)?
7. Какие начальные условия могут быть при решении навигационной задачи?

6.4 Файл из папки «Координаты приемника»

Файл *positionV0.m*

```

%Имя m-файла:positionV0.m
%Входные данные:
%Координаты спутников GPS; задаются в виде коэффициентов матрицы A(i,j);i=1,...,11- условный номер
%спутника; j= 1,...,3- координата x, y, z соответственно; размерность метр.
%Координаты спутников ГЛОНАСС; задаются в виде коэффициентов матрицы A(i,j);i=12,...,15- условный
%номер спутника; j= 1,...,3- координата x, y, z соответственно; размерность метр.
%Измеренные псевдодальности до спутников GPS; задаются в виде вектора P(i),i=1,...,11, размерность
%метр.
%Измеренные псевдодальности до спутников ГЛОНАСС; задаются в виде вектора P(i),i=12,...,15,
% записанного в виде произведения скорости света умноженной на время распространения сигнала
% от спутника до фазового центра антенны приемника, размерность метр.
%N- количество спутников участвующих в расчетах
%Выходные данные:
%Позиция приемника COOR(1:3): COOR(1), COOR(2), COOR(3)- координаты x, y, z соответственно.
%delta- разность между рассчитанной и измеренной позициями приемника
%PDOP- пространственный геометрический фактор
% Координаты позиции приемника
Rx=3504451.023;
Ry=2061316.876;
Rz=4897990.975;
c=299792458;% скорость света, м/сек
%Координаты спутников GPS
A(1,1)=7439695.07931521 ;
A(1,2)=18359432.4518111 ;
A(1,3)= -18242050.2245942;

A(2,1)=13202399.4654164;
A(2,2)=12654731.9283261;

```

A(2,3)=-19136436.3144384;

A(3,1)=-3504533.72350397;

A(3,2)=-22776742.1951248;

A(3,3)=13405348.1750206;

%5

A(4,1)=-16282945.450944;

A(4,2)=-1270691.73493271;

A(4,3)=21052945.4792735;

A(5,1)=6910739.42467448;

A(5,2)=-15581694.9377712;

A(5,3)=-20210215.8550669;

A(6,1)=-19914539.0913112 ;

A(6,2)=-11309423.4985691;

A(6,3)=13445462.9750841;

A(7,1)=23697731.7992212 ;

A(7,2)=-7760416.42074437;

A(7,3)=-9233928.25341097;

A(8,1)=-9966566.45967017;

A(8,2)=14363132.6112417;

A(8,3)=19937635.2625642;

A(9,1)=-16928888.1195712;

A(9,2)= 1667886.70427429;

A(9,3)=-20937559.8370906;

A(10,1)=12332560.1018299 ;

A(10,2)=-23677009.6410659;

A(10,3)=-712026.572990682;

A(11,1)=-16509966.8126889 ;

A(11,2)=-10852956.1028937;

A(11,3)=-18034660.0011514;

%Координаты спутников ГЛОНАСС

A(15,1)=12279069.90;

A(15,2)=10026127.03;

A(15,3)=19974462.48;

A(14,1)=24375988.15 ;

```

A(14,2)=4063057.68;
A(14,3)=6272838.71;
A(13,1)=-12411749.31;
A(13,2)=397576.93;
A(13,3)=22271317.88;
A(12,1)=-681900.76;
A(12,2)=-16134355.65;
A(12,3)= 19720999.24;

```

```

%Начало отсчета для решения задачи расчета координат позиции приемника (центр земли)
X=0;Y=0;Z=0;T1=0.0;T2=0;

```

```

% Задание ТОЧНОСТИ вычислений

```

```

eps1=10;
dX=eps1*10;
dY=eps1*10;dZ=eps1*10;dT1=eps1*10;dT2=eps1*10;
apred=[X Y Z T1 T2];
da=[eps1*10 eps1*10 eps1*10 eps1*10 eps1*10];%abs(apred)
while da > [eps1 eps1 eps1 eps1 eps1]
a=[X Y Z T1 T2];
R=[a;a;a;a;a;a;a;a;a;a;a];

```

```

W=0.0;%коэффициент при искусственном введении ошибки в псевдодальности; может выбираться
%произвольно
%Измеренные псевдодальности до спутников GPS

```

```

P(1) = 2.857579786854914e+007+W*c;
P(2)= 2.79986489244377e+007+W*c ;
P(3)= 2.71740716883972e+007+W*c ;
P(4)= 2.57609373982301e+007+W*c ;
P(5)= 3.08755682823437e+007+W*c ;
P(6)= 2.82893102378011e+007+W*c ;
P(7)= 2.65319839917412e+007+W*c;
P(8)= 2.36430514675815e+007+W*c ;
P(9)= 3.29416426666836e+007+W*c ;
P(10)=2.77825351380247e+007+W*c ;
P(11)=3.30645104102405e+007+W*c;

```

```

%Измеренные псевдодальности до спутников ГЛОНАСС

```

```

P(15)=c*0.0635691153+W*c;
P(14)=c*0.0696933803+W*c;
P(13)=c*0.0783930561+W*c;
P(12)=c*0.0791243185+W*c;

```

```

%i=1:15;j=1:3;
n=15;
for ( i = 1: n )
%Расчетные псевдодальности
PR(i)=sqrt((A(i,1)-R(1,1))^2+(A(i,2)-R(1,2))^2....
+(A(i,3)-R(1,3))^2);
%Формирование матрицы для решения навигационной задачи
for ( j = 1: 3)
    D(i,j) = (A(i,j) - R(i)) / PR(i);
end;
D(1:11,4)=1;
D(12:15,4)=0;
D(1:11,5)=0;
D(12:15,5)=1;
DD=[D];

%Расчет приращений
L(i)=PR(i)-P(i);
end; % for ( i =1: n )
%Выбор спутников
%ДЛЯ ДАННОЙ ЗАДАЧИ НЕ МЕНЕЕ 5 ОБЯЗАТЕЛЬНО ИЗ 1-11 И 12-15
N=15;%ЧИСЛО СПУТНИКОВ
W1=eye(N);% единичная матрица (может выступать как матрица весовых коэффициентов
VSv=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15];% спутники, участвующие в решении навигационной задачи
%(размерность вектора VSv равна N
G=DD(VSv,:);
DL=L(VSv);
Dr=inv(G'*W1*G)*G'*W1*DL';
%Конечный этап решения навигационной задачи
a1=a+(Dr(1:5))';
X=a1(1);
Y=a1(2);
Z=a1(3);
Dr(4)=a1(4);
Dr(5)=a1(5);
dX=abs(a1(1)-apred(1));
dY=abs(a1(2)-apred(2));
dZ=abs(a1(3)-apred(3));
dT1=abs(a1(4)-apred(4));
dT2=abs(a1(5)-apred(5));
da=[dX dY dZ dT1 dT2];
apred=a1;

```

```
Dr(4);
RM=(inv(G'*G));
COOR=a1';

end % while da > [eps1 eps1 eps1 eps1 eps1]
%Результат решения (отображается в командном окне)
COOR(1:3)
delta=COOR(1:3)- [ Rx; Ry ;Rz]
PDOP=sqrt(RM(1,1)+RM(2,2)+RM(3,3))
```

ПРИЛОЖЕНИЯ

Приложение 1 Отчеты по лабораторной работе

Национальный авиационный университет			
Кафедра аэронавигационных систем			
Дисциплина	Авиационные геоинформационные технологии и системы		
Название работы	Преобразование координат	Модуль	4
Группа	309	Студент	Дата
Задание 1			
Входные данные	Результат		
Задание 2			
Входные данные	Результат		
Задание 3			
Входные данные	Результат		
Задание 4			
Входные данные	Результат		
Задание 5			
Входные данные	Результат		
Задание 6			
Входные данные	Результат		
Вопросы и предложения по совершенствованию пакета программ, возникшие при выполнении лабораторной работы (не снижают оценки)			
Работу выполнил	Работа защищена	Оценка	Баллы

Приложение 2 Альманах almanac_yuma_week0371_589824.txt

***** Week 371 almanac for PRN-01 *****

ID: 01
Health: 000
Eccentricity: 0.6433486938E-002
Time of Applicability(s): 589824.0000
Orbital Inclination(rad): 0.9891683363
Rate of Right Ascen(r/s): -0.7428880871E-008
SQRT(A) (m 1/2): 5153.682617
Right Ascen at Week(rad): 0.1840750730E+000
Argument of Perigee(rad): -1.785071272
Mean Anom(rad): 0.7153421297E+000
Af0(s): 0.8106231689E-004
Af1(s/s): 0.3637978807E-011
week: 371

***** Week 371 almanac for PRN-02 *****

ID: 02
Health: 000
Eccentricity: 0.8791923523E-002
Time of Applicability(s): 589824.0000
Orbital Inclination(rad): 0.9486137192
Rate of Right Ascen(r/s): -0.8343204671E-008
SQRT(A) (m 1/2): 5153.676270
Right Ascen at Week(rad): -0.1956580136E+001
Argument of Perigee(rad): 2.154924301
Mean Anom(rad): 0.1956755406E+001
Af0(s): 0.4291534424E-004
Af1(s/s): 0.3637978807E-011
week: 371

***** Week 371 almanac for PRN-03 *****

ID: 03
Health: 000
Eccentricity: 0.8668422699E-002
Time of Applicability(s): 589824.0000
Orbital Inclination(rad): 0.9265986981
Rate of Right Ascen(r/s): -0.8000333246E-008
SQRT(A) (m 1/2): 5153.649902
Right Ascen at Week(rad): -0.3094183060E+001
Argument of Perigee(rad): 0.693907594
Mean Anom(rad): 0.1250441517E+001

Af0(s): 0.0000000000E+000
Af1(s/s): 0.0000000000E+000
week: 371

***** Week 371 almanac for PRN-04 *****

ID: 04
Health: 000
Eccentricity: 0.7489204407E-002
Time of Applicability(s): 589824.0000
Orbital Inclination(rad): 0.9480983976
Rate of Right Ascen(r/s): -0.8377491813E-008
SQRT(A) (m 1/2): 5153.614746
Right Ascen at Week(rad): -0.1937034989E+001
Argument of Perigee(rad): 0.175379020
Mean Anom(rad): -0.1777541434E+001
Af0(s): 0.3204345703E-003
Af1(s/s): 0.3637978807E-011
week: 371

***** Week 371 almanac for PRN-05 *****

ID: 05
Health: 000
Eccentricity: 0.7562637329E-002
Time of Applicability(s): 589824.0000
Orbital Inclination(rad): 0.9383791912
Rate of Right Ascen(r/s): -0.8377491813E-008
SQRT(A) (m 1/2): 5153.708496
Right Ascen at Week(rad): 0.2152298633E+001
Argument of Perigee(rad): 1.119532585
Mean Anom(rad): -0.2106830109E+001
Af0(s): 0.3604888916E-003
Af1(s/s): 0.7275957614E-011
week: 371

***** Week 371 almanac for PRN-06 *****

ID: 06
Health: 000
Eccentricity: 0.5908012390E-002
Time of Applicability(s): 589824.0000
Orbital Inclination(rad): 0.9341367755
Rate of Right Ascen(r/s): -0.7908900866E-008
SQRT(A) (m 1/2): 5153.619141

Right Ascen at Week(rad): -0.3034609103E+001
Argument of Perigee(rad): -1.780730361
Mean Anom(rad): -0.9042577060E+000
Af0(s): 0.5846023560E-003
Af1(s/s): -0.7275957614E-011
week: 371

***** Week 371 almanac for PRN-07 *****

ID: 07
Health: 000
Eccentricity: 0.1040410995E-001
Time of Applicability(s): 589824.0000
Orbital Inclination(rad): 0.9356827406
Rate of Right Ascen(r/s): -0.7886042771E-008
SQRT(A) (m 1/2): 5153.685059
Right Ascen at Week(rad): -0.3060504766E+001
Argument of Perigee(rad): -1.749639911
Mean Anom(rad): -0.1922240838E+001
Af0(s): 0.5168914795E-003
Af1(s/s): -0.3637978807E-011
week: 371

***** Week 371 almanac for PRN-08 *****

ID: 08
Health: 000
Eccentricity: 0.9914398193E-002
Time of Applicability(s): 589824.0000
Orbital Inclination(rad): 0.9745475819
Rate of Right Ascen(r/s): -0.7966046104E-008
SQRT(A) (m 1/2): 5153.539062
Right Ascen at Week(rad): 0.1231207959E+001
Argument of Perigee(rad): 2.686667357
Mean Anom(rad): -0.7501956263E+000
Af0(s): -0.8201599121E-004
Af1(s/s): 0.0000000000E+000
week: 371

***** Week 371 almanac for PRN-09 *****

ID: 09
Health: 000
Eccentricity: 0.1845741272E-001
Time of Applicability(s): 589824.0000

Orbital Inclination(rad): 0.9622577593
Rate of Right Ascen(r/s): -0.8068907531E-008
SQRT(A) (m 1/2): 5153.671387
Right Ascen at Week(rad): 0.1149889750E+001
Argument of Perigee(rad): 1.287954388
Mean Anom(rad): -0.1098117899E+001
Af0(s): 0.3910064697E-004
Af1(s/s): 0.0000000000E+000
week: 371

***** Week 371 almanac for PRN-10 *****

ID: 10
Health: 000
Eccentricity: 0.7097244263E-002
Time of Applicability(s): 589824.0000
Orbital Inclination(rad): 0.9720308947
Rate of Right Ascen(r/s): -0.8148910863E-008
SQRT(A) (m 1/2): 5153.645020
Right Ascen at Week(rad): -0.8825482826E+000
Argument of Perigee(rad): 0.433749921
Mean Anom(rad): 0.2462789677E+001
Af0(s): 0.9059906006E-004
Af1(s/s): 0.0000000000E+000
week: 371

***** Week 371 almanac for PRN-11 *****

ID: 11
Health: 000
Eccentricity: 0.5926132202E-002
Time of Applicability(s): 589824.0000
Orbital Inclination(rad): 0.8959370587
Rate of Right Ascen(r/s): -0.8766079428E-008
SQRT(A) (m 1/2): 5153.632812
Right Ascen at Week(rad): -0.2108013177E+001
Argument of Perigee(rad): 0.390399234
Mean Anom(rad): -0.2848103480E+000
Af0(s): 0.3538131714E-003
Af1(s/s): 0.3637978807E-011
week: 371

***** Week 371 almanac for PRN-13 *****

ID: 13

Health: 000
Eccentricity: 0.3000736237E-002
Time of Applicability(s): 589824.0000
Orbital Inclination(rad): 0.9933448387
Rate of Right Ascen(r/s): -0.7371735634E-008
SQRT(A) (m 1/2): 5153.679199
Right Ascen at Week(rad): 0.1701995876E+000
Argument of Perigee(rad): 1.288820997
Mean Anom(rad): 0.2232474970E+001
Af0(s): 0.1010894775E-003
Af1(s/s): 0.3637978807E-011
week: 371

***** Week 371 almanac for PRN-14 *****

ID: 14
Health: 000
Eccentricity: 0.2767562866E-002
Time of Applicability(s): 589824.0000
Orbital Inclination(rad): 0.9874366158
Rate of Right Ascen(r/s): -0.7440309919E-008
SQRT(A) (m 1/2): 5153.655762
Right Ascen at Week(rad): 0.1563154885E+000
Argument of Perigee(rad): -2.149011959
Mean Anom(rad): 0.1778858575E+001
Af0(s): -0.3814697266E-005
Af1(s/s): 0.0000000000E+000
week: 371

***** Week 371 almanac for PRN-15 *****

ID: 15
Health: 063
Eccentricity: 0.9871482849E-002
Time of Applicability(s): 589824.0000
Orbital Inclination(rad): 0.9550732165
Rate of Right Ascen(r/s): -0.8331775623E-008
SQRT(A) (m 1/2): 5153.590332
Right Ascen at Week(rad): -0.1878719376E+001
Argument of Perigee(rad): 2.707203450
Mean Anom(rad): -0.8356221774E+000
Af0(s): -0.1239776611E-004
Af1(s/s): -0.7275957614E-011
week: 371

***** Week 371 almanac for PRN-16 *****

ID: 16
Health: 000
Eccentricity: 0.3452777863E-002
Time of Applicability(s): 589824.0000
Orbital Inclination(rad): 0.9625393886
Rate of Right Ascen(r/s): -0.8046049436E-008
SQRT(A) (m 1/2): 5153.742676
Right Ascen at Week(rad): 0.2246890869E+001
Argument of Perigee(rad): -0.752799948
Mean Anom(rad): -0.2465972986E+001
Af0(s): 0.8010864258E-004
Af1(s/s): 0.3637978807E-011
week: 371

***** Week 371 almanac for PRN-17 *****

ID: 17
Health: 000
Eccentricity: 0.2206325531E-002
Time of Applicability(s): 589824.0000
Orbital Inclination(rad): 0.9605200467
Rate of Right Ascen(r/s): -0.7646032774E-008
SQRT(A) (m 1/2): 5153.700684
Right Ascen at Week(rad): -0.2997563617E+001
Argument of Perigee(rad): 3.072893084
Mean Anom(rad): 0.2863357900E+001
Af0(s): 0.9346008301E-004
Af1(s/s): 0.0000000000E+000
week: 371

***** Week 371 almanac for PRN-18 *****

ID: 18
Health: 000
Eccentricity: 0.7766723633E-002
Time of Applicability(s): 589824.0000
Orbital Inclination(rad): 0.9561158440
Rate of Right Ascen(r/s): -0.8320346576E-008
SQRT(A) (m 1/2): 5153.662598
Right Ascen at Week(rad): -0.8627342392E+000
Argument of Perigee(rad): -2.623552062
Mean Anom(rad): -0.2391461443E+001

Af0(s): -0.2508163452E-003
Af1(s/s): 0.0000000000E+000
week: 371

***** Week 371 almanac for PRN-19 *****

ID: 19
Health: 000
Eccentricity: 0.3551006317E-002
Time of Applicability(s): 589824.0000
Orbital Inclination(rad): 0.9584467758
Rate of Right Ascen(r/s): -0.7691748964E-008
SQRT(A) (m 1/2): 5153.620605
Right Ascen at Week(rad): -0.2938595987E+001
Argument of Perigee(rad): -1.107215049
Mean Anom(rad): 0.2514810201E+001
Af0(s): 0.2193450928E-004
Af1(s/s): 0.3637978807E-011
week: 371

***** Week 371 almanac for PRN-20 *****

ID: 20
Health: 000
Eccentricity: 0.2744674683E-002
Time of Applicability(s): 589824.0000
Orbital Inclination(rad): 0.9556065145
Rate of Right Ascen(r/s): -0.8366062766E-008
SQRT(A) (m 1/2): 5153.614258
Right Ascen at Week(rad): -0.9157782912E+000
Argument of Perigee(rad): 1.354951823
Mean Anom(rad): -0.2545847096E+001
Af0(s): -0.2193450928E-004
Af1(s/s): 0.0000000000E+000
week: 371

***** Week 371 almanac for PRN-21 *****

ID: 21
Health: 000
Eccentricity: 0.1168870926E-001
Time of Applicability(s): 589824.0000
Orbital Inclination(rad): 0.9416448924
Rate of Right Ascen(r/s): -0.8468924193E-008
SQRT(A) (m 1/2): 5153.627441

Right Ascen at Week(rad): -0.1917635525E+001
Argument of Perigee(rad): -2.914216703
Mean Anom(rad): -0.1107067493E+001
Af0(s): 0.5626678467E-004
Af1(s/s): 0.0000000000E+000
week: 371

***** Week 371 almanac for PRN-22 *****

ID: 22
Health: 000
Eccentricity: 0.5007266998E-002
Time of Applicability(s): 589824.0000
Orbital Inclination(rad): 0.9536770543
Rate of Right Ascen(r/s): -0.8343204671E-008
SQRT(A) (m 1/2): 5153.625000
Right Ascen at Week(rad): -0.8549179031E+000
Argument of Perigee(rad): -1.675495384
Mean Anom(rad): 0.2358868096E+001
Af0(s): 0.1401901245E-003
Af1(s/s): 0.3637978807E-011
week: 371

***** Week 371 almanac for PRN-23 *****

ID: 23
Health: 000
Eccentricity: 0.4817008972E-002
Time of Applicability(s): 589824.0000
Orbital Inclination(rad): 0.9696759945
Rate of Right Ascen(r/s): -0.7600316584E-008
SQRT(A) (m 1/2): 5153.688965
Right Ascen at Week(rad): 0.1404933156E+000
Argument of Perigee(rad): 2.539610056
Mean Anom(rad): 0.1512449255E+001
Af0(s): 0.1382827759E-003
Af1(s/s): 0.0000000000E+000
week: 371

***** Week 371 almanac for PRN-24 *****

ID: 24
Health: 000
Eccentricity: 0.8838176727E-002
Time of Applicability(s): 589824.0000

Orbital Inclination(rad): 0.9582849887
Rate of Right Ascen(r/s): -0.8263201338E-008
SQRT(A) (m 1/2): 5155.323242
Right Ascen at Week(rad): -0.1901905855E+001
Argument of Perigee(rad): -0.829185525
Mean Anom(rad): -0.7264556257E+000
Af0(s): 0.4196166992E-004
Af1(s/s): 0.3637978807E-011
week: 371

***** Week 371 almanac for PRN-25 *****

ID: 25
Health: 000
Eccentricity: 0.1263189316E-001
Time of Applicability(s): 589824.0000
Orbital Inclination(rad): 0.9556604435
Rate of Right Ascen(r/s): -0.8057478483E-008
SQRT(A) (m 1/2): 5153.598633
Right Ascen at Week(rad): 0.1094536862E+001
Argument of Perigee(rad): -1.322550598
Mean Anom(rad): -0.5200636789E+000
Af0(s): 0.3700256348E-003
Af1(s/s): 0.1818989404E-010
week: 371

***** Week 371 almanac for PRN-26 *****

ID: 26
Health: 000
Eccentricity: 0.1748418808E-001
Time of Applicability(s): 589824.0000
Orbital Inclination(rad): 0.9914573233
Rate of Right Ascen(r/s): -0.7406022776E-008
SQRT(A) (m 1/2): 5153.694336
Right Ascen at Week(rad): 0.1706179119E+000
Argument of Perigee(rad): 0.818631543
Mean Anom(rad): 0.6507314281E+000
Af0(s): -0.2841949463E-003
Af1(s/s): -0.1455191523E-010
week: 371

***** Week 371 almanac for PRN-27 *****

ID: 27

Health: 000
Eccentricity: 0.1995992661E-001
Time of Applicability(s): 589824.0000
Orbital Inclination(rad): 0.9601964726
Rate of Right Ascen(r/s): -0.8080336578E-008
SQRT(A) (m 1/2): 5153.531250
Right Ascen at Week(rad): 0.1126961306E+001
Argument of Perigee(rad): -1.892119987
Mean Anom(rad): -0.1971266429E+001
Af0(s): 0.7152557373E-004
Af1(s/s): 0.3637978807E-011
week: 371

***** Week 371 almanac for PRN-28 *****

ID: 28
Health: 000
Eccentricity: 0.1171159744E-001
Time of Applicability(s): 589824.0000
Orbital Inclination(rad): 0.9606279047
Rate of Right Ascen(r/s): -0.8091765626E-008
SQRT(A) (m 1/2): 5153.618652
Right Ascen at Week(rad): 0.2256309346E+001
Argument of Perigee(rad): -2.241381122
Mean Anom(rad): 0.3110969588E+001
Af0(s): 0.1716613770E-004
Af1(s/s): 0.0000000000E+000
week: 371

***** Week 371 almanac for PRN-29 *****

ID: 29
Health: 000
Eccentricity: 0.9555339813E-002
Time of Applicability(s): 589824.0000
Orbital Inclination(rad): 0.9880538034
Rate of Right Ascen(r/s): -0.7440309919E-008
SQRT(A) (m 1/2): 5153.555664
Right Ascen at Week(rad): 0.1356359595E+000
Argument of Perigee(rad): -0.841310565
Mean Anom(rad): 0.2484434685E+001
Af0(s): 0.4196166992E-004
Af1(s/s): 0.5093170330E-010
week: 371

***** Week 371 almanac for PRN-30 *****

ID: 30
Health: 000
Eccentricity: 0.9448528290E-002
Time of Applicability(s): 589824.0000
Orbital Inclination(rad): 0.9449225780
Rate of Right Ascen(r/s): -0.8308917528E-008
SQRT(A) (m 1/2): 5153.618164
Right Ascen at Week(rad): 0.2199777136E+001
Argument of Perigee(rad): 1.316367861
Mean Anom(rad): -0.2894497036E+001
Af0(s): 0.1239776611E-004
Af1(s/s): 0.0000000000E+000
week: 371

***** Week 371 almanac for PRN-31 *****

ID: 31
Health: 063
Eccentricity: 0.4019737244E-002
Time of Applicability(s): 589824.0000
Orbital Inclination(rad): 0.9588722157
Rate of Right Ascen(r/s): -0.7931758961E-008
SQRT(A) (m 1/2): 5171.043457
Right Ascen at Week(rad): 0.1184157143E+001
Argument of Perigee(rad): -2.911533361
Mean Anom(rad): 0.9753739691E+000
Af0(s): -0.8773803711E-004
Af1(s/s): 0.2546585165E-010
week: 371

Приложение 3 Альманах 002.txt

GPS: id=1

health=0
eccentricity=0.00668621063232421875
corr. to inclination=0.9891563653945922852
rate of right ascension=-8.011762260562438342e-12
square root of semi-major axis=5153.6787109375
right ascension parameter=-1.190934181213378906
argument of perigee=-1.770861387252807617
mean anomaly=1.673161864280700684
Af0m=0.09632110595703125

Af1=3.637978807091712952e-12

Af0l=0

week almanac=382

time of week almanac=405504000

GPS: id=2

health=0

eccentricity=0.00882053375244140625

corr. to inclination=0.948260188102722168

rate of right ascension=-7.714606731212825252e-12

square root of semi-major axis=5153.7060546875

right ascension parameter=2.949104547500610352

argument of perigee=2.182616949081420898

mean anomaly=2.895400762557983398

Af0m=0.06580352783203125

Af1=3.637978807091712952e-12

Af0l=0

week almanac=382

time of week almanac=405504000

GPS: id=3

health=0

eccentricity=0.0090236663818359375

corr. to inclination=0.9254841804504394531

rate of right ascension=-8.263201754787896647e-12

square root of semi-major axis=5153.63916015625

right ascension parameter=1.81049799919128418

argument of perigee=0.726136922836303711

mean anomaly=2.211142301559448242

Af0m=0.47397613525390625

Af1=1.818989403545856476e-11

Af0l=0

week almanac=382

time of week almanac=405504000

GPS: id=4

health=0

eccentricity=0.00775909423828125

corr. to inclination=0.9477388858795166016

rate of right ascension=-7.748894408077244834e-12

square root of semi-major axis=5153.6650390625

right ascension parameter=2.96854710578918457

argument of perigee=0.2202932685613632202

mean anomaly=-0.8276365399360656738

Af0m=0.34809112548828125

Af1=3.637978807091712952e-12

Af0l=0

week almanac=382

time of week almanac=405504000

GPS: id=5

health=0

eccentricity=0.007638454437255859375

corr. to inclination=0.9389963746070861816

rate of right ascension=-7.806039668822872812e-12

square root of semi-major axis=5153.82275390625

right ascension parameter=0.7738637328147888184

argument of perigee=1.15076756477355957

mean anomaly=-1.178214788436889648

Af0m=0.438690185546875

Af1=1.818989403545856476e-11

Af0l=0

week almanac=382

time of week almanac=405504000

GPS: id=6

health=0

eccentricity=0.00571727752685546875

corr. to inclination=0.9330402016639709473

rate of right ascension=-8.18319804279932228e-12

square root of semi-major axis=5153.53857421875

right ascension parameter=1.870720624923706055

argument of perigee=-1.767207980155944824

mean anomaly=0.1111147403717041016

Af0m=0.49877166748046875

Af1=-1.818989403545856476e-11

Af0l=0

week almanac=382

time of week almanac=405504000

GPS: id=7

health=0

eccentricity=0.010822296142578125

corr. to inclination=0.9345921874046325684

rate of right ascension=-8.18319804279932228e-12

square root of semi-major axis=5152.01513671875

right ascension parameter=1.84484410285949707

argument of perigee=-1.705631256103515625

mean anomaly=-0.2914162874221801758

Af0m=0.46825408935546875

Af1=-1.091393642127513886e-11

Af0l=0

week almanac=382

time of week almanac=405504000

GPS: id=8

health=255

eccentricity=0.009907245635986328125

corr. to inclination=0.9763332605361938477

rate of right ascension=-7.611745435343042488e-12

square root of semi-major axis=5153.5458984375

right ascension parameter=-0.1443144083023071289

argument of perigee=2.730259895324707031

mean anomaly=0.2543621659278869629

Af0m=-0.09059906005859375

Af1=0

Af0l=0

week almanac=382

time of week almanac=405504000

GPS: id=9

health=0

eccentricity=0.01850032806396484375

corr. to inclination=0.9640074372291564941

rate of right ascension=-7.783181217579926426e-12

square root of semi-major axis=5153.68310546875

right ascension parameter=-0.2266040891408920288

argument of perigee=1.299516916275024414

mean anomaly=-0.1355632990598678589

Af0m=0.049591064453125

Af1=0

Af0l=0

week almanac=382

time of week almanac=405504000

GPS: id=10

health=0

eccentricity=0.00734424591064453125

corr. to inclination=0.9713417887687683105

rate of right ascension=-7.771751991958453232e-12

square root of semi-major axis=5153.69580078125

right ascension parameter=-2.259819507598876953

argument of perigee=0.4371939003467559814

mean anomaly=-2.844457387924194336

Af0m=0.09441375732421875

Af1=0
Af0l=0
week almanac=382
time of week almanac=405504000

GPS: id=11

health=0
eccentricity=0.006408214569091796875
corr. to inclination=0.8953858017921447754
rate of right ascension=-8.274630113047631852e-12
square root of semi-major axis=5153.63232421875
right ascension parameter=2.79404592514038086
argument of perigee=0.440089195966720581
mean anomaly=0.6644263863563537598
Af0m=0.00095367431640625
Af1=0
Af0l=0
week almanac=382
time of week almanac=405504000

GPS: id=12

health=0
eccentricity=0.003298282623291015625
corr. to inclination=0.9608616232872009277
rate of right ascension=-7.600316209721569294e-12
square root of semi-major axis=5153.68505859375
right ascension parameter=0.8550205230712890625
argument of perigee=-1.343068361282348633
mean anomaly=1.30158078670501709
Af0m=-0.08106231689453125
Af1=1.091393642127513886e-11
Af0l=0
week almanac=382
time of week almanac=405504000

GPS: id=13

health=0
eccentricity=0.002939701080322265625
corr. to inclination=0.9933208823204040527
rate of right ascension=-7.97747545105975675e-12
square root of semi-major axis=5153.63232421875
right ascension parameter=-1.204549789428710938
argument of perigee=1.263086795806884766
mean anomaly=-3.038789987564086914
Af0m=0.12493133544921875

Af1=3.637978807091712952e-12

Af0l=0

week almanac=382

time of week almanac=405504000

GPS: id=14

health=0

eccentricity=0.002999782562255859375

corr. to inclination=0.9873707294464111328

rate of right ascension=-8.03462071180538473e-12

square root of semi-major axis=5153.67041015625

right ascension parameter=-1.218866467475891113

argument of perigee=-2.083027839660644531

mean anomaly=2.695198297500610352

Af0m=0.0019073486328125

Af1=0

Af0l=0

week almanac=382

time of week almanac=405504000

GPS: id=16

health=0

eccentricity=0.003540515899658203125

corr. to inclination=0.9628809094429016113

rate of right ascension=-7.634603886585988874e-12

square root of semi-major axis=5153.51513671875

right ascension parameter=0.8703708052635192871

argument of perigee=-0.7588751912117004394

mean anomaly=-1.416219711303710938

Af0m=0.10395050048828125

Af1=3.637978807091712952e-12

Af0l=0

week almanac=382

time of week almanac=405504000

GPS: id=17

health=0

eccentricity=0.002378940582275390625

corr. to inclination=0.9593995213508605957

rate of right ascension=-7.88604251344970919e-12

square root of semi-major axis=5153.6953125

right ascension parameter=1.909842729568481445

argument of perigee=3.054111957550048828

mean anomaly=-2.438158750534057617

Af0m=0.0858306884765625

Af1=0
Af0l=0
week almanac=382
time of week almanac=405504000
GPS: id=18
health=0
eccentricity=0.007780551910400390625
corr. to inclination=0.9554147720336914062
rate of right ascension=-7.908900964692655578e-12
square root of semi-major axis=5153.69287109375
right ascension parameter=-2.241117000579833984
argument of perigee=-2.594098567962646484
mean anomaly=-1.445593714714050293
Af0m=-0.2536773681640625
Af1=0
Af0l=0
week almanac=382
time of week almanac=405504000
GPS: id=19
health=0
eccentricity=0.00347232818603515625
corr. to inclination=0.9573382139205932617
rate of right ascension=-7.908900964692655578e-12
square root of semi-major axis=5153.6171875
right ascension parameter=1.968618631362915039
argument of perigee=-1.0020391941070556641
mean anomaly=-2.867439746856689453
Af0m=0.0476837158203125
Af1=3.637978807091712952e-12
Af0l=0
week almanac=382
time of week almanac=405504000
GPS: id=20
health=0
eccentricity=0.002883434295654296875
corr. to inclination=0.9549533724784851074
rate of right ascension=-7.897471739071182384e-12
square root of semi-major axis=5153.5830078125
right ascension parameter=-2.294228076934814453
argument of perigee=1.280813694000244141
mean anomaly=-1.452993631362915039
Af0m=-0.02193450927734375

Af1=0
Af0l=0
week almanac=382
time of week almanac=405504000

GPS: id=21

health=0
eccentricity=0.01182270050048828125
corr. to inclination=0.9413033127784729004
rate of right ascension=-7.817468894444346006e-12
square root of semi-major axis=5153.6162109375
right ascension parameter=2.987459897994995117
argument of perigee=-2.899925470352172852
mean anomaly=-0.1171304434537887573
Af0m=0.06580352783203125
Af1=0
Af0l=0
week almanac=382
time of week almanac=405504000

GPS: id=22

health=0
eccentricity=0.00497913360595703125
corr. to inclination=0.9529759883880615234
rate of right ascension=-7.94318777419533717e-12
square root of semi-major axis=5153.6376953125
right ascension parameter=-2.233473777770996094
argument of perigee=-1.647951126098632812
mean anomaly=-2.951039314270019531
Af0m=0.15735626220703125
Af1=3.637978807091712952e-12
Af0l=0
week almanac=382
time of week almanac=405504000

GPS: id=23

health=0
eccentricity=0.00469684600830078125
corr. to inclination=0.9696160554885864258
rate of right ascension=-8.206056494042268668e-12
square root of semi-major axis=5153.6533203125
right ascension parameter=-1.235992670059204102
argument of perigee=2.586992979049682617
mean anomaly=2.443128108978271484
Af0m=0.1430511474609375

Af1=0
Af0l=0
week almanac=382
time of week almanac=405504000

GPS: id=24

health=0
eccentricity=0.008758544921875
corr. to inclination=0.9579614400863647461
rate of right ascension=-7.566029400218887702e-12
square root of semi-major axis=5155.3740234375
right ascension parameter=3.00449061393737793
argument of perigee=-0.772795259952545166
mean anomaly=-0.7267537117004394531
Af0m=0.06198883056640625
Af1=3.637978807091712952e-12
Af0l=0
week almanac=382
time of week almanac=405504000

GPS: id=25

health=0
eccentricity=0.01288080215454101562
corr. to inclination=0.9573562145233154297
rate of right ascension=-7.874614155189973986e-12
square root of semi-major axis=5153.671875
right ascension parameter=-0.2823831737041473389
argument of perigee=-1.308804750442504883
mean anomaly=0.4633183777332305908
Af0m=0.400543212890625
Af1=-7.275957614183425904e-12
Af0l=0
week almanac=382
time of week almanac=405504000

GPS: id=26

health=0
eccentricity=0.017627716064453125
corr. to inclination=0.9914153814315795898
rate of right ascension=-8.000333034940965149e-12
square root of semi-major axis=5153.564453125
right ascension parameter=-1.204336762428283691
argument of perigee=0.8254280686378479004
mean anomaly=1.649268627166748047
Af0m=-0.057220458984375

Af1=-1.818989403545856476e-11

Af0l=0

week almanac=382

time of week almanac=405504000

GPS: id=27

health=0

eccentricity=0.02038526535034179688

corr. to inclination=0.9619401693344116211

rate of right ascension=-7.794610443201399619e-12

square root of semi-major axis=5153.75048828125

right ascension parameter=-0.249639272689819336

argument of perigee=-1.867654919624328613

mean anomaly=-0.9617273211479187012

Af0m=0.08392333984375

Af1=3.637978807091712952e-12

Af0l=0

week almanac=382

time of week almanac=405504000

GPS: id=28

health=0

eccentricity=0.0119800567626953125

corr. to inclination=0.9609574675559997559

rate of right ascension=-7.588887851461834089e-12

square root of semi-major axis=5153.6337890625

right ascension parameter=0.8796859383583068848

argument of perigee=-2.232036352157592774

mean anomaly=-2.177708625793457031

Af0m=0.011444091796875

Af1=0

Af0l=0

week almanac=382

time of week almanac=405504000

GPS: id=29

health=0

eccentricity=0.00990962982177734375

corr. to inclination=0.9879399538040161133

rate of right ascension=-8.011762260562438342e-12

square root of semi-major axis=5153.56396484375

right ascension parameter=-1.239597320556640625

argument of perigee=-0.8064963817596435547

mean anomaly=-2.79503321647644043

Af0m=0.27370452880859375

Af1=2.182787284255027771e-11

Af0l=0

week almanac=382

time of week almanac=405504000

GPS: id=30

health=0

eccentricity=0.009554386138916015625

corr. to inclination=0.9454199075698852539

rate of right ascension=-7.748894408077244834e-12

square root of semi-major axis=5153.60595703125

right ascension parameter=0.821842193603515625

argument of perigee=1.328736305236816406

mean anomaly=-1.895180821418762207

Af0m=0.02002716064453125

Af1=0

Af0l=0

week almanac=382

time of week almanac=405504000

GPS: id=31

health=0

eccentricity=0.00612545013427734375

corr. to inclination=0.9606039524078369141

rate of right ascension=-7.794610443201399619e-12

square root of semi-major axis=5153.56884765625

right ascension parameter=-0.1925415545701980591

argument of perigee=-1.421732783317565918

mean anomaly=0.668113410472869873

Af0m=-0.0171661376953125

Af1=7.275957614183425904e-12

Af0l=0

week almanac=382

time of week almanac=405504000

GLONASS: id=1

здоровье=0

номер литера=7

уход шкалы времени спутника=-0.057220458984375

долгота первого внутри суток восходящего узла=3.13790440559387207

наклонение орбиты=1.125113844871520996

эксцентриситет=0.00095844268798828125

аргумент перигея орбиты спутника=2.598275899887084961

время прохождения восходящего узла орбиты=21426938

драконический период=40543929.6875

скорость изменения драконического периода=-0.18310546875
номер дня=1086
GLONASS: id=2
здоровье=0
номер литера=1
уход шкалы времени спутника=0.030517578125
долгота первого внутри суток восходящего узла=2.796126365661621094
наклонение орбиты=1.127121090888977051
эксцентриситет=0.00434970855712890625
аргумент перигея орбиты спутника=1.660821795463562012
время прохождения восходящего узла орбиты=26365062
драконический период=40543832.03125
скорость изменения драконического периода=-0.18310546875
номер дня=1086
GLONASS: id=3
здоровье=255
номер литера=12
уход шкалы времени спутника=-0.1220703125
долгота первого внутри суток восходящего узла=2.377643346786499024
наклонение орбиты=1.12163841724395752
эксцентриситет=0.002330780029296875
аргумент перигея орбиты спутника=-2.618601083755493164
время прохождения восходящего узла орбиты=31583032
драконический период=40543917.96875
скорость изменения драконического периода=-0.18310546875
номер дня=1086
GLONASS: id=4
здоровье=0
номер литера=6
уход шкалы времени спутника=-0.209808349609375
долгота первого внутри суток восходящего узла=2.064510345458984375
наклонение орбиты=1.12721097469329834
эксцентриситет=0.00396633148193359375
аргумент перигея орбиты спутника=1.437052369117736816
время прохождения восходящего узла орбиты=36400220
драконический период=40543822.265625
скорость изменения драконического периода=-0.18310546875
номер дня=1086
GLONASS: id=5
здоровье=255
номер литера=7
уход шкалы времени спутника=0

долгота первого внутри суток восходящего узла=0
наклонение орбиты=1.099557399749755859
эксцентриситет=0
аргумент перигея орбиты спутника=0
время прохождения восходящего узла орбиты=0
драконический период=43200000
скорость изменения драконического периода=0
номер дня=1086

GLONASS: id=6

здоровье=0
номер литера=1
уход шкалы времени спутника=0.034332275390625
долгота первого внутри суток восходящего узла=3.117007017135620117
наклонение орбиты=1.126872420310974121
эксцентриситет=0.004566192626953125
аргумент перигея орбиты спутника=1.749025702476501465
время прохождения восходящего узла орбиты=22698750
драконический период=40543734.375
скорость изменения драконического периода=-0.244140625
номер дня=1083

GLONASS: id=7

здоровье=0
номер литера=4
уход шкалы времени спутника=-0.20599365234375
долгота первого внутри суток восходящего узла=2.752881288528442383
наклонение орбиты=1.124948978424072266
эксцентриситет=0.00047016143798828125
аргумент перигея орбиты спутника=2.485048770904541016
время прохождения восходящего узла орбиты=27434032
драконический период=40544041.015625
скорость изменения драконического периода=-0.244140625
номер дня=1083

GLONASS: id=8

здоровье=0
номер литера=6
уход шкалы времени спутника=0.0457763671875
долгота первого внутри суток восходящего узла=2.38481283187866211
наклонение орбиты=1.12516176700592041
эксцентриситет=0.0001850128173828125
аргумент перигея орбиты спутника=2.661936044692993164
время прохождения восходящего узла орбиты=32490438
драконический период=40544046.875

скорость изменения драконического периода=-0.30517578125
номер дня=1083
GLONASS: id=9
здоровье=255
номер литера=0
уход шкалы времени спутника=0
долгота первого внутри суток восходящего узла=0
наклонение орбиты=1.099557399749755859
эксцентриситет=0
аргумент перигея орбиты спутника=0
время прохождения восходящего узла орбиты=0
драконический период=43200000
скорость изменения драконического периода=0
номер дня=1083
GLONASS: id=10
здоровье=255
номер литера=0
уход шкалы времени спутника=0
долгота первого внутри суток восходящего узла=0
наклонение орбиты=1.099557399749755859
эксцентриситет=0
аргумент перигея орбиты спутника=0
время прохождения восходящего узла орбиты=0
драконический период=43200000
скорость изменения драконического периода=0
номер дня=1083
GLONASS: id=11
здоровье=255
номер литера=0
уход шкалы времени спутника=0
долгота первого внутри суток восходящего узла=0
наклонение орбиты=1.099557399749755859
эксцентриситет=0
аргумент перигея орбиты спутника=0
время прохождения восходящего узла орбиты=0
драконический период=43200000
скорость изменения драконического периода=0
номер дня=1083
GLONASS: id=12
здоровье=255
номер литера=0
уход шкалы времени спутника=0

долгота первого внутри суток восходящего узла=0
наклонение орбиты=1.099557399749755859
эксцентриситет=0
аргумент перигея орбиты спутника=0
время прохождения восходящего узла орбиты=0
драконический период=43200000
скорость изменения драконического периода=0
номер дня=1083
GLONASS: id=13
здоровье=255
номер литера=0
уход шкалы времени спутника=0
долгота первого внутри суток восходящего узла=0
наклонение орбиты=1.099557399749755859
эксцентриситет=0
аргумент перигея орбиты спутника=0
время прохождения восходящего узла орбиты=0
драконический период=43200000
скорость изменения драконического периода=0
номер дня=1083
GLONASS: id=14
здоровье=255
номер литера=0
уход шкалы времени спутника=0
долгота первого внутри суток восходящего узла=0
наклонение орбиты=1.099557399749755859
эксцентриситет=0
аргумент перигея орбиты спутника=0
время прохождения восходящего узла орбиты=0
драконический период=43200000
скорость изменения драконического периода=0
номер дня=1083
GLONASS: id=15
здоровье=255
номер литера=0
уход шкалы времени спутника=0
долгота первого внутри суток восходящего узла=0
наклонение орбиты=1.099557399749755859
эксцентриситет=0
аргумент перигея орбиты спутника=0
время прохождения восходящего узла орбиты=0
драконический период=43200000

скорость изменения драконического периода=0
номер дня=1083
GLONASS: id=16
здоровье=255
номер литера=0
уход шкалы времени спутника=0
долгота первого внутри суток восходящего узла=0
наклонение орбиты=1.099557399749755859
эксцентриситет=0
аргумент перигея орбиты спутника=0
время прохождения восходящего узла орбиты=0
драконический период=43200000
скорость изменения драконического периода=0
номер дня=1083
GLONASS: id=17
здоровье=255
номер литера=5
уход шкалы времени спутника=0
долгота первого внутри суток восходящего узла=0
наклонение орбиты=1.099557399749755859
эксцентриситет=0
аргумент перигея орбиты спутника=0
время прохождения восходящего узла орбиты=0
драконический период=43200000
скорость изменения драконического периода=0
номер дня=1083
GLONASS: id=18
здоровье=0
номер литера=10
уход шкалы времени спутника=-0.049591064453125
долгота первого внутри суток восходящего узла=2.599642038345336914
наклонение орбиты=1.125769972801208496
эксцентриситет=0.0044803619384765625
аргумент перигея орбиты спутника=-2.544298887252807617
время прохождения восходящего узла орбиты=990531.25
драконический период=40544230.46875
скорость изменения драконического периода=2.25830078125
номер дня=1083
GLONASS: id=19
здоровье=0
номер литера=3
уход шкалы времени спутника=-0.133514404296875

долгота первого внутри суток восходящего узла=2.391326189041137695
наклонение орбиты=1.13172602653503418
эксцентриситет=0.0021305084228515625
аргумент перигея орбиты спутника=2.57890939712524414
время прохождения восходящего узла орбиты=3720968.75
драконический период=40543798.828125
скорость изменения драконического периода=1.15966796875
номер дня=1083

GLONASS: id=20

здоровье=255
номер литера=11
уход шкалы времени спутника=0
долгота первого внутри суток восходящего узла=0
наклонение орбиты=1.099557399749755859
эксцентриситет=0
аргумент перигея орбиты спутника=0
время прохождения восходящего узла орбиты=0
драконический период=43200000
скорость изменения драконического периода=0
номер дня=1083

GLONASS: id=21

здоровье=0
номер литера=5
уход шкалы времени спутника=0.049591064453125
долгота первого внутри суток восходящего узла=-0.1811745166778564453
наклонение орбиты=1.129386186599731445
эксцентриситет=0.00034046173095703125
аргумент перигея орбиты спутника=-0.3073714077472686768
время прохождения восходящего узла орбиты=38448936
драконический период=40543839.84375
скорость изменения драконического периода=0.18310546875
номер дня=1086

GLONASS: id=22

здоровье=0
номер литера=10
уход шкалы времени спутника=-0.0152587890625
долгота первого внутри суток восходящего узла=2.414731502532958984
наклонение орбиты=1.129173398017883301
эксцентриситет=0.0003719329833984375
аргумент перигея орбиты спутника=1.688817024230957031
время прохождения восходящего узла орбиты=2847375
драконический период=40543816.40625

скорость изменения драконического периода=0.6103515625
 номер дня=1086
 GLONASS: id=23
 здоровье=255
 номер литера=3
 уход шкалы времени спутника=0.06866455078125
 долгота первого внутри суток восходящего узла=2.016429662704467774
 наклонение орбиты=1.131525397300720215
 эксцентриситет=0.0008907318115234375
 аргумент перигея орбиты спутника=0.1979793906211853027
 время прохождения восходящего узла орбиты=8127531
 драконический период=40543927.734375
 скорость изменения драконического периода=0.30517578125
 номер дня=1086
 GLONASS: id=24
 здоровье=0
 номер литера=2
 уход шкалы времени спутника=0.09918212890625
 долгота первого внутри суток восходящего узла=1.651369214057922363
 наклонение орбиты=1.131845951080322266
 эксцентриситет=0.00195980072021484375
 аргумент перигея орбиты спутника=-0.151192978024482727
 время прохождения восходящего узла орбиты=13136219
 драконический период=40543720.703125
 скорость изменения драконического периода=0.06103515625
 номер дня=1086

Приложение 4 Альманах AlmGGG.yum

*** Week 1406 almaNAU for PRN-01 *****

ID: 1
 Health: 0
 Eccentricity: 0.6686210632E-002
 Time of Applicability(s): 405504.0000
 Orbital Incluation(rad): 0.9891563654
 Rate of Right Ascen(r/s): -0.8011762261E-008
 SQRT(A) (m^{1/2}): 5153.6787109
 Right Ascen at Week(rad): -0.1190934181E+001
 Argument of Perigee(rad): -0.1770861387E+001
 Mean Anom(rad): 0.1673161864E+001
 Af0(s): 0.9632110596E-004
 Af1(s/s): 0.3637978807E-011
 week: 1406

**** Week 1406 almaNAU for PRN-02 ****

ID: 2
Health: 0
Eccentricity: 0.8820533752E-002
Time of Applicability(s): 405504.0000
Orbital Inclination(rad): 0.9482601881
Rate of Right Ascen(r/s): -0.7714606731E-008
SQRT(A) (m^{1/2}): 5153.7060547
Right Ascen at Week(rad): 0.2949104548E+001
Argument of Perigee(rad): 0.2182616949E+001
Mean Anom(rad): 0.2895400763E+001
Af0(s): 0.6580352783E-004
Af1(s/s): 0.3637978807E-011
week: 1406

**** Week 1406 almaNAU for PRN-03 ****

ID: 3
Health: 0
Eccentricity: 0.9023666382E-002
Time of Applicability(s): 405504.0000
Orbital Inclination(rad): 0.9254841805
Rate of Right Ascen(r/s): -0.8263201755E-008
SQRT(A) (m^{1/2}): 5153.6391602
Right Ascen at Week(rad): 0.1810497999E+001
Argument of Perigee(rad): 0.7261369228E+000
Mean Anom(rad): 0.2211142302E+001
Af0(s): 0.4739761353E-003
Af1(s/s): 0.1818989404E-010
week: 1406

**** Week 1406 almaNAU for PRN-04 ****

ID: 4
Health: 0
Eccentricity: 0.7759094238E-002
Time of Applicability(s): 405504.0000
Orbital Inclination(rad): 0.9477388859
Rate of Right Ascen(r/s): -0.7748894408E-008
SQRT(A) (m^{1/2}): 5153.6650391
Right Ascen at Week(rad): 0.2968547106E+001
Argument of Perigee(rad): 0.2202932686E+000
Mean Anom(rad): -0.8276365399E+000

Af0(s): 0.3480911255E-003
Af1(s/s): 0.3637978807E-011
week: 1406

**** Week 1406 almaNAU for PRN-05 ****

ID: 5
Health: 0
Eccentricity: 0.7638454437E-002
Time of Applicability(s): 405504.0000
Orbital Incluation(rad): 0.9389963746
Rate of Right Ascen(r/s): -0.7806039669E-008
SQRT(A) (m^{1/2}): 5153.8227539
Right Ascen at Week(rad): 0.7738637328E+000
Argument of Perigee(rad): 0.1150767565E+001
Mean Anom(rad): -0.1178214788E+001
Af0(s): 0.4386901855E-003
Af1(s/s): 0.1818989404E-010
week: 1406

**** Week 1406 almaNAU for PRN-06 ****

ID: 6
Health: 0
Eccentricity: 0.5717277527E-002
Time of Applicability(s): 405504.0000
Orbital Incluation(rad): 0.9330402017
Rate of Right Ascen(r/s): -0.8183198043E-008
SQRT(A) (m^{1/2}): 5153.5385742
Right Ascen at Week(rad): 0.1870720625E+001
Argument of Perigee(rad): -0.1767207980E+001
Mean Anom(rad): 0.1111147404E+000
Af0(s): 0.4987716675E-003
Af1(s/s): -0.1818989404E-010
week: 1406

**** Week 1406 almaNAU for PRN-07 ****

ID: 7
Health: 0
Eccentricity: 0.1082229614E-001
Time of Applicability(s): 405504.0000
Orbital Incluation(rad): 0.9345921874
Rate of Right Ascen(r/s): -0.8183198043E-008
SQRT(A) (m^{1/2}): 5152.0151367

Right Ascen at Week(rad): 0.1844844103E+001
Argument of Perigee(rad): -0.1705631256E+001
Mean Anom(rad): -0.2914162874E+000
Af0(s): 0.4682540894E-003
Af1(s/s): -0.1091393642E-010
week: 1406

**** Week 1406 almaNAU for PRN-08 ****

ID: 8
Health: 255
Eccentricity: 0.9907245636E-002
Time of Applicability(s): 405504.0000
Orbital Incluation(rad): 0.9763332605
Rate of Right Ascen(r/s): -0.7611745435E-008
SQRT(A) (m^{1/2}): 5153.5458984
Right Ascen at Week(rad): -0.1443144083E+000
Argument of Perigee(rad): 0.2730259895E+001
Mean Anom(rad): 0.2543621659E+000
Af0(s): -0.9059906006E-004
Af1(s/s): 0.0000000000E+000
week: 1406

**** Week 1406 almaNAU for PRN-09 ****

ID: 9
Health: 0
Eccentricity: 0.1850032806E-001
Time of Applicability(s): 405504.0000
Orbital Incluation(rad): 0.9640074372
Rate of Right Ascen(r/s): -0.7783181218E-008
SQRT(A) (m^{1/2}): 5153.6831055
Right Ascen at Week(rad): -0.2266040891E+000
Argument of Perigee(rad): 0.1299516916E+001
Mean Anom(rad): -0.1355632991E+000
Af0(s): 0.4959106445E-004
Af1(s/s): 0.0000000000E+000
week: 1406

**** Week 1406 almaNAU for PRN-010 ****

ID: 10
Health: 0
Eccentricity: 0.7344245911E-002
Time of Applicability(s): 405504.0000

Orbital Incluation(rad): 0.9713417888
Rate of Right Ascen(r/s): -0.7771751992E-008
SQRT(A) (m^{1/2}): 5153.6958008
Right Ascen at Week(rad): -0.2259819508E+001
Argument of Perigee(rad): 0.4371939003E+000
Mean Anom(rad): -0.2844457388E+001
Af0(s): 0.9441375732E-004
Af1(s/s): 0.0000000000E+000
week: 1406

**** Week 1406 almaNAU for PRN-011 ****

ID: 11
Health: 0
Eccentricity: 0.6408214569E-002
Time of Applicability(s): 405504.0000
Orbital Incluation(rad): 0.8953858018
Rate of Right Ascen(r/s): -0.8274630113E-008
SQRT(A) (m^{1/2}): 5153.6323242
Right Ascen at Week(rad): 0.2794045925E+001
Argument of Perigee(rad): 0.4400891960E+000
Mean Anom(rad): 0.6644263864E+000
Af0(s): 0.9536743164E-006
Af1(s/s): 0.0000000000E+000
week: 1406

**** Week 1406 almaNAU for PRN-012 ****

ID: 12
Health: 0
Eccentricity: 0.3298282623E-002
Time of Applicability(s): 405504.0000
Orbital Incluation(rad): 0.9608616233
Rate of Right Ascen(r/s): -0.7600316210E-008
SQRT(A) (m^{1/2}): 5153.6850586
Right Ascen at Week(rad): 0.8550205231E+000
Argument of Perigee(rad): -0.1343068361E+001
Mean Anom(rad): 0.1301580787E+001
Af0(s): -0.8106231689E-004
Af1(s/s): 0.1091393642E-010
week: 1406

**** Week 1406 almaNAU for PRN-013 ****

ID: 13

Health: 0
Eccentricity: 0.2939701080E-002
Time of Applicability(s): 405504.0000
Orbital Inclination(rad): 0.9933208823
Rate of Right Ascen(r/s): -0.7977475451E-008
SQRT(A) (m^{1/2}): 5153.6323242
Right Ascen at Week(rad): -0.1204549789E+001
Argument of Perigee(rad): 0.1263086796E+001
Mean Anom(rad): -0.3038789988E+001
Af0(s): 0.1249313354E-003
Af1(s/s): 0.3637978807E-011
week: 1406

**** Week 1406 almaNAU for PRN-014 ****

ID: 14
Health: 0
Eccentricity: 0.2999782562E-002
Time of Applicability(s): 405504.0000
Orbital Inclination(rad): 0.9873707294
Rate of Right Ascen(r/s): -0.8034620712E-008
SQRT(A) (m^{1/2}): 5153.6704102
Right Ascen at Week(rad): -0.1218866467E+001
Argument of Perigee(rad): -0.2083027840E+001
Mean Anom(rad): 0.2695198298E+001
Af0(s): 0.1907348633E-005
Af1(s/s): 0.0000000000E+000
week: 1406

**** Week 1406 almaNAU for PRN-016 ****

ID: 16
Health: 0
Eccentricity: 0.3540515900E-002
Time of Applicability(s): 405504.0000
Orbital Inclination(rad): 0.9628809094
Rate of Right Ascen(r/s): -0.7634603887E-008
SQRT(A) (m^{1/2}): 5153.5151367
Right Ascen at Week(rad): 0.8703708053E+000
Argument of Perigee(rad): -0.7588751912E+000
Mean Anom(rad): -0.1416219711E+001
Af0(s): 0.1039505005E-003
Af1(s/s): 0.3637978807E-011
week: 1406

**** Week 1406 almaNAU for PRN-017 ****

ID: 17
Health: 0
Eccentricity: 0.2378940582E-002
Time of Applicability(s): 405504.0000
Orbital Incluation(rad): 0.9593995214
Rate of Right Ascen(r/s): -0.7886042513E-008
SQRT(A) (m^{1/2}): 5153.6953125
Right Ascen at Week(rad): 0.1909842730E+001
Argument of Perigee(rad): 0.3054111958E+001
Mean Anom(rad): -0.2438158751E+001
Af0(s): 0.8583068848E-004
Af1(s/s): 0.0000000000E+000
week: 1406

**** Week 1406 almaNAU for PRN-018 ****

ID: 18
Health: 0
Eccentricity: 0.7780551910E-002
Time of Applicability(s): 405504.0000
Orbital Incluation(rad): 0.9554147720
Rate of Right Ascen(r/s): -0.7908900965E-008
SQRT(A) (m^{1/2}): 5153.6928711
Right Ascen at Week(rad): -0.2241117001E+001
Argument of Perigee(rad): -0.2594098568E+001
Mean Anom(rad): -0.1445593715E+001
Af0(s): -0.2536773682E-003
Af1(s/s): 0.0000000000E+000
week: 1406

**** Week 1406 almaNAU for PRN-019 ****

ID: 19
Health: 0
Eccentricity: 0.3472328186E-002
Time of Applicability(s): 405504.0000
Orbital Incluation(rad): 0.9573382139
Rate of Right Ascen(r/s): -0.7908900965E-008
SQRT(A) (m^{1/2}): 5153.6171875
Right Ascen at Week(rad): 0.1968618631E+001
Argument of Perigee(rad): -0.1002039194E+001
Mean Anom(rad): -0.2867439747E+001

Af0(s): 0.4768371582E-004
Af1(s/s): 0.3637978807E-011
week: 1406

**** Week 1406 almaNAU for PRN-020 ****

ID: 20
Health: 0
Eccentricity: 0.2883434296E-002
Time of Applicability(s): 405504.0000
Orbital Incluation(rad): 0.9549533725
Rate of Right Ascen(r/s): -0.7897471739E-008
SQRT(A) (m^{1/2}): 5153.5830078
Right Ascen at Week(rad): -0.2294228077E+001
Argument of Perigee(rad): 0.1280813694E+001
Mean Anom(rad): -0.1452993631E+001
Af0(s): -0.2193450928E-004
Af1(s/s): 0.0000000000E+000
week: 1406

**** Week 1406 almaNAU for PRN-021 ****

ID: 21
Health: 0
Eccentricity: 0.1182270050E-001
Time of Applicability(s): 405504.0000
Orbital Incluation(rad): 0.9413033128
Rate of Right Ascen(r/s): -0.7817468894E-008
SQRT(A) (m^{1/2}): 5153.6162109
Right Ascen at Week(rad): 0.2987459898E+001
Argument of Perigee(rad): -0.2899925470E+001
Mean Anom(rad): -0.1171304435E+000
Af0(s): 0.6580352783E-004
Af1(s/s): 0.0000000000E+000
week: 1406

**** Week 1406 almaNAU for PRN-022 ****

ID: 22
Health: 0
Eccentricity: 0.4979133606E-002
Time of Applicability(s): 405504.0000
Orbital Incluation(rad): 0.9529759884
Rate of Right Ascen(r/s): -0.7943187774E-008
SQRT(A) (m^{1/2}): 5153.6376953

Right Ascen at Week(rad): -0.2233473778E+001
Argument of Perigee(rad): -0.1647951126E+001
Mean Anom(rad): -0.2951039314E+001
Af0(s): 0.1573562622E-003
Af1(s/s): 0.3637978807E-011
week: 1406

**** Week 1406 almaNAU for PRN-023 ****

ID: 23
Health: 0
Eccentricity: 0.4696846008E-002
Time of Applicability(s): 405504.0000
Orbital Incluation(rad): 0.9696160555
Rate of Right Ascen(r/s): -0.8206056494E-008
SQRT(A) (m^{1/2}): 5153.6533203
Right Ascen at Week(rad): -0.1235992670E+001
Argument of Perigee(rad): 0.2586992979E+001
Mean Anom(rad): 0.2443128109E+001
Af0(s): 0.1430511475E-003
Af1(s/s): 0.0000000000E+000
week: 1406

**** Week 1406 almaNAU for PRN-024 ****

ID: 24
Health: 0
Eccentricity: 0.8758544922E-002
Time of Applicability(s): 405504.0000
Orbital Incluation(rad): 0.9579614401
Rate of Right Ascen(r/s): -0.7566029400E-008
SQRT(A) (m^{1/2}): 5155.3740234
Right Ascen at Week(rad): 0.3004490614E+001
Argument of Perigee(rad): -0.7727952600E+000
Mean Anom(rad): -0.7267537117E+000
Af0(s): 0.6198883057E-004
Af1(s/s): 0.3637978807E-011
week: 1406

**** Week 1406 almaNAU for PRN-025 ****

ID: 25
Health: 0
Eccentricity: 0.1288080215E-001
Time of Applicability(s): 405504.0000

Orbital Incluation(rad): 0.9573562145
Rate of Right Ascen(r/s): -0.7874614155E-008
SQRT(A) (m^{1/2}): 5153.6718750
Right Ascen at Week(rad): -0.2823831737E+000
Argument of Perigee(rad): -0.1308804750E+001
Mean Anom(rad): 0.4633183777E+000
Af0(s): 0.4005432129E-003
Af1(s/s): -0.7275957614E-011
week: 1406

**** Week 1406 almaNAU for PRN-026 ****

ID: 26
Health: 0
Eccentricity: 0.1762771606E-001
Time of Applicability(s): 405504.0000
Orbital Incluation(rad): 0.9914153814
Rate of Right Ascen(r/s): -0.8000333035E-008
SQRT(A) (m^{1/2}): 5153.5644531
Right Ascen at Week(rad): -0.1204336762E+001
Argument of Perigee(rad): 0.8254280686E+000
Mean Anom(rad): 0.1649268627E+001
Af0(s): -0.5722045898E-004
Af1(s/s): -0.1818989404E-010
week: 1406

**** Week 1406 almaNAU for PRN-027 ****

ID: 27
Health: 0
Eccentricity: 0.2038526535E-001
Time of Applicability(s): 405504.0000
Orbital Incluation(rad): 0.9619401693
Rate of Right Ascen(r/s): -0.7794610443E-008
SQRT(A) (m^{1/2}): 5153.7504883
Right Ascen at Week(rad): -0.2496392727E+000
Argument of Perigee(rad): -0.1867654920E+001
Mean Anom(rad): -0.9617273211E+000
Af0(s): 0.8392333984E-004
Af1(s/s): 0.3637978807E-011
week: 1406

**** Week 1406 almaNAU for PRN-028 ****

ID: 28

Health: 0
Eccentricity: 0.1198005676E-001
Time of Applicability(s): 405504.0000
Orbital Inclination(rad): 0.9609574676
Rate of Right Ascen(r/s): -0.7588887851E-008
SQRT(A) (m^{1/2}): 5153.6337891
Right Ascen at Week(rad): 0.8796859384E+000
Argument of Perigee(rad): -0.2232036352E+001
Mean Anom(rad): -0.2177708626E+001
Af0(s): 0.1144409180E-004
Af1(s/s): 0.0000000000E+000
week: 1406

**** Week 1406 almaNAU for PRN-029 ****

ID: 29
Health: 0
Eccentricity: 0.9909629822E-002
Time of Applicability(s): 405504.0000
Orbital Inclination(rad): 0.9879399538
Rate of Right Ascen(r/s): -0.8011762261E-008
SQRT(A) (m^{1/2}): 5153.5639648
Right Ascen at Week(rad): -0.1239597321E+001
Argument of Perigee(rad): -0.8064963818E+000
Mean Anom(rad): -0.2795033216E+001
Af0(s): 0.2737045288E-003
Af1(s/s): 0.2182787284E-010
week: 1406

**** Week 1406 almaNAU for PRN-030 ****

ID: 30
Health: 0
Eccentricity: 0.9554386139E-002
Time of Applicability(s): 405504.0000
Orbital Inclination(rad): 0.9454199076
Rate of Right Ascen(r/s): -0.7748894408E-008
SQRT(A) (m^{1/2}): 5153.6059570
Right Ascen at Week(rad): 0.8218421936E+000
Argument of Perigee(rad): 0.1328736305E+001
Mean Anom(rad): -0.1895180821E+001
Af0(s): 0.2002716064E-004
Af1(s/s): 0.0000000000E+000
week: 1406

**** Week 1406 almaNAU for PRN-031 *****

ID: 31
Health: 0
Eccentricity: 0.6125450134E-002
Time of Applicability(s): 405504.0000
Orbital Incluation(rad): 0.9606039524
Rate of Right Ascen(r/s): -0.7794610443E-008
SQRT(A) (m^{1/2}): 5153.5688477
Right Ascen at Week(rad): -0.1925415546E+000
Argument of Perigee(rad): -0.1421732783E+001
Mean Anom(rad): 0.6681134105E+000
Af0(s): -0.1716613770E-004
Af1(s/s): 0.7275957614E-011
week: 1406

GLO*Week 1406 almanac for prn-38 NAUHn-7*****

ID: 38
Health: 000
Eccentricity: 0.9584426880E-003
Time of Applicability(s): 0.3562269380E+006
Orbital Incluation(rad): 1.125113845e+000
Rate of Right Ascen(r/s): 0.9569060294E-008
SQRT(A) (m^{1/2}): 5050.788727
Right Ascen at Week(rad): 0.3912832946E+001
Argument of Perigee(rad): 0.2598275900E+001
Mean Anom(rad): -0.2597284302E+001
Af0(s): -0.5722045898E-004
Af1(s/s): 0.000000000e+000
week: 1406

GLO*Week 1406 almanac for prn-39 NAUHn-1*****

ID: 39
Health: 000
Eccentricity: 0.4349708557E-002
Time of Applicability(s): 0.3611650620E+006
Orbital Incluation(rad): 1.127121091e+000
Rate of Right Ascen(r/s): 0.9594496759E-008
SQRT(A) (m^{1/2}): 5050.785669
Right Ascen at Week(rad): 0.3931148587E+001
Argument of Perigee(rad): 0.1660821795E+001
Mean Anom(rad): -0.1652155092E+001

Af0(s): 0.3051757813E-004
Af1(s/s): 0.000000000e+000
week: 1406

GLO*Week 1406 almanac for prn-40 NAUHn-12*****

ID: 40
Health: 255
Eccentricity: 0.2330780029E-002
Time of Applicability(s): 0.3663830320E+006
Orbital Incluation(rad): 1.121638417e+000
Rate of Right Ascen(r/s): 0.9525712332E-008
SQRT(A) (m^{1/2}): 5050.786367
Right Ascen at Week(rad): 0.3893165941E+001
Argument of Perigee(rad): -0.2618601084E+001
Mean Anom(rad): 0.2616269225E+001
Af0(s): -0.1220703125E-003
Af1(s/s): 0.000000000e+000
week: 1406

GLO*Week 1406 almanac for prn-41 NAUHn-6*****

ID: 41
Health: 000
Eccentricity: 0.3966331482E-002
Time of Applicability(s): 0.3712002200E+006
Orbital Incluation(rad): 1.127210975e+000
Rate of Right Ascen(r/s): 0.9595558774E-008
SQRT(A) (m^{1/2}): 5050.785635
Right Ascen at Week(rad): 0.3931307828E+001
Argument of Perigee(rad): 0.1437052369E+001
Mean Anom(rad): -0.1429193686E+001
Af0(s): -0.2098083496E-003
Af1(s/s): 0.000000000e+000
week: 1406

GLO*Week 1406 almanac for prn-42 NAUHn-7*****

ID: 42
Health: 255
Eccentricity: 0.000000000E+000
Time of Applicability(s): 0.3348000000E+006
Orbital Incluation(rad): 1.099557400e+000
Rate of Right Ascen(r/s): 0.7974461797E-008
SQRT(A) (m^{1/2}): 5158.728496

Right Ascen at Week(rad): -0.7875484200E+000
Argument of Perigee(rad): 0.0000000000E+000
Mean Anom(rad): 0.0000000000E+000
Af0(s): 0.0000000000E+000
Af1(s/s): 0.0000000000E+000
week: 1406

GLO*Week 1406 almanac for prn-43 NAUHN-1*****

ID: 43
Health: 000
Eccentricity: 0.4566192627E-002
Time of Applicability(s): 0.9829875000E+005
Orbital Inclination(rad): 1.126872420E+000
Rate of Right Ascen(r/s): 0.9591488890E-008
SQRT(A) (m^{1/2}): 5050.781360
Right Ascen at Week(rad): 0.3984677551E+001
Argument of Perigee(rad): 0.1749025702E+001
Mean Anom(rad): -0.1740032552E+001
Af0(s): 0.3433227539E-004
Af1(s/s): 0.0000000000E+000
week: 1406

GLO*Week 1406 almanac for prn-44 NAUHN-4*****

ID: 44
Health: 000
Eccentricity: 0.4701614380E-003
Time of Applicability(s): 0.1030340320E+006
Orbital Inclination(rad): 1.124948978E+000
Rate of Right Ascen(r/s): 0.9566926156E-008
SQRT(A) (m^{1/2}): 5050.793445
Right Ascen at Week(rad): 0.3965854031E+001
Argument of Perigee(rad): 0.2485048771E+001
Mean Anom(rad): -0.2484474654E+001
Af0(s): -0.2059936523E-003
Af1(s/s): 0.0000000000E+000
week: 1406

GLO*Week 1406 almanac for prn-45 NAUHN-6*****

ID: 45
Health: 000
Eccentricity: 0.1850128174E-003
Time of Applicability(s): 0.1080904380E+006

Orbital Incluation(rad): 1.125161767e+000
Rate of Right Ascen(r/s): 0.9569575246E-008
SQRT(A) (m^{1/2}): 5050.793852
Right Ascen at Week(rad): 0.3966504515E+001
Argument of Perigee(rad): 0.2661936045E+001
Mean Anom(rad): -0.2661765266E+001
Af0(s): 0.4577636719E-004
Af1(s/s): 0.000000000e+000
week: 1406

GLO*Week 1406 almanac for prn-54 NAUHN-5*****

ID: 54
Health: 255
Eccentricity: 0.000000000E+000
Time of Applicability(s): 0.756000000E+005
Orbital Incluation(rad): 1.099557400e+000
Rate of Right Ascen(r/s): 0.7974461797E-008
SQRT(A) (m^{1/2}): 5158.728496
Right Ascen at Week(rad): -0.7875484200E+000
Argument of Perigee(rad): 0.000000000E+000
Mean Anom(rad): 0.000000000E+000
Af0(s): 0.000000000E+000
Af1(s/s): 0.000000000e+000
week: 1406

GLO*Week 1406 almanac for prn-55 NAUHN-10*****

ID: 55
Health: 000
Eccentricity: 0.4480361938E-002
Time of Applicability(s): 0.7659053125E+005
Orbital Incluation(rad): 1.125769973e+000
Rate of Right Ascen(r/s): 0.9577458107E-008
SQRT(A) (m^{1/2}): 5050.800447
Right Ascen at Week(rad): 0.1884324296E+001
Argument of Perigee(rad): -0.2544298887E+001
Mean Anom(rad): 0.2539245283E+001
Af0(s): -0.4959106445E-004
Af1(s/s): 0.000000000e+000
week: 1406

GLO*Week 1406 almanac for prn-56 NAUHN-3*****

ID: 56

Health: 000
Eccentricity: 0.2130508423E-002
Time of Applicability(s): 0.7932096875E+005
Orbital Inclination(rad): 1.131726027e+000
Rate of Right Ascen(r/s): 0.9651539532E-008
SQRT(A) (m^{1/2}): 5050.785680
Right Ascen at Week(rad): 0.1875115089E+001
Argument of Perigee(rad): 0.2578909397E+001
Mean Anom(rad): -0.2576633250E+001
Af0(s): -0.1335144043E-003
Af1(s/s): 0.000000000e+000
week: 1406

GLO*Week 1406 almanac for prn-57 NAUHN-11*****

ID: 57
Health: 255
Eccentricity: 0.000000000E+000
Time of Applicability(s): 0.7560000000E+005
Orbital Inclination(rad): 1.099557400e+000
Rate of Right Ascen(r/s): 0.7974461797E-008
SQRT(A) (m^{1/2}): 5158.728496
Right Ascen at Week(rad): -0.7875484200E+000
Argument of Perigee(rad): 0.000000000E+000
Mean Anom(rad): 0.000000000E+000
Af0(s): 0.000000000E+000
Af1(s/s): 0.000000000e+000
week: 1406

GLO*Week 1406 almanac for prn-58 NAUHN-5*****

ID: 58
Health: 000
Eccentricity: 0.3404617310E-003
Time of Applicability(s): 0.3732489360E+006
Orbital Inclination(rad): 1.129386187e+000
Rate of Right Ascen(r/s): 0.9622338149E-008
SQRT(A) (m^{1/2}): 5050.787180
Right Ascen at Week(rad): 0.1835017693E+001
Argument of Perigee(rad): -0.3073714077E+000
Mean Anom(rad): 0.3071654416E+000
Af0(s): 0.4959106445E-004
Af1(s/s): 0.000000000e+000
week: 1406

GLO*Week 1406 almanac for prn-59 NAUHn-10*****

ID: 59
Health: 000
Eccentricity: 0.3719329834E-003
Time of Applicability(s): 0.3376473750E+006
Orbital Incluation(rad): 1.129173398e+000
Rate of Right Ascen(r/s): 0.9619704708E-008
SQRT(A) (m^{1/2}): 5050.785985
Right Ascen at Week(rad): 0.1834816942E+001
Argument of Perigee(rad): 0.1688817024E+001
Mean Anom(rad): -0.1688078309E+001
Af0(s): -0.1525878906E-004
Af1(s/s): 0.000000000e+000
week: 1406

GLO*Week 1406 almanac for prn-60 NAUHn-3*****

ID: 60
Health: 255
Eccentricity: 0.8907318115E-003
Time of Applicability(s): 0.3429275310E+006
Orbital Incluation(rad): 1.131525397e+000
Rate of Right Ascen(r/s): 0.9648896251E-008
SQRT(A) (m^{1/2}): 5050.791913
Right Ascen at Week(rad): 0.1821550150E+001
Argument of Perigee(rad): 0.1979793906E+000
Mean Anom(rad): -0.1976292264E+000
Af0(s): 0.6866455078E-004
Af1(s/s): 0.000000000e+000
week: 1406

GLO*Week 1406 almanac for prn-61 NAUHn-2*****

ID: 61
Health: 000
Eccentricity: 0.1959800720E-002
Time of Applicability(s): 0.3479362190E+006
Orbital Incluation(rad): 1.131845951e+000
Rate of Right Ascen(r/s): 0.9653048520E-008
SQRT(A) (m^{1/2}): 5050.783839
Right Ascen at Week(rad): 0.1821728990E+001
Argument of Perigee(rad): -0.1511929780E+000
Mean Anom(rad): 0.1506034738E+000

Af0(s): 0.9918212891E-004
Af1(s/s): 0.000000000e+000
week: 1406

**** Week 1406 almaNAU for PRN-01 ****

ID: 62
Health: 0
Eccentricity: 0.0
Time of Applicability(s): 405504.0000
Orbital Incluation(rad): 0.9773843811
Rate of Right Ascen(r/s): -0.78174684861E-008
SQRT(A) (m^{1/2}): 5481.73472
Right Ascen at Week(rad): 0.0
Argument of Perigee(rad): 0.785398
Mean Anom(rad): 0.0
Af0(s): 0.34332275391E-004
Af1(s/s): 0.36379788071E-011
week: 1406

**** Week 1406 almaNAU for PRN-02 ****

ID: 63
Health: 0
Eccentricity: 0.91261863708E-002
Time of Applicability(s): 405504.0000
Orbital Incluation(rad): 0.9773843811
Rate of Right Ascen(r/s): -0.82974884808E-008
SQRT(A) (m^{1/2}): 5481.73472
Right Ascen at Week(rad): 0.0
Argument of Perigee(rad): 0.785398
Mean Anom(rad): 0.69813
Af0(s): -0.20027160645E-004
Af1(s/s): 0.00000000000E+000
week: 1406

**** Week 1406 almaNAU for PRN-03 ****

ID: 64
Health: 0
Eccentricity: 0.78430175781E-002
Time of Applicability(s): 405504.0000
Orbital Incluation(rad): 0.9773843811
Rate of Right Ascen(r/s): -0.82974884808E-008
SQRT(A) (m^{1/2}): 5481.73472

Right Ascen at Week(rad): 0.0
Argument of Perigee(rad): 0.785398
Mean Anom(rad): 1.3963
Af0(s): 0.71525573730E-004
Af1(s/s): 0.36379788071E-011
week: 1406

**** Week 1406 almaNAU for PRN-04 ****

ID: 65
Health: 0
Eccentricity: 0.74453353882E-002
Time of Applicability(s): 0.0
Orbital Incluation(rad): 0.9773843811
Rate of Right Ascen(r/s): -0.83089175283E-008
SQRT(A) (m^{1/2}): 5481.73472
Right Ascen at Week(rad): 0.0
Argument of Perigee(rad): 0.785398
Mean Anom(rad): 2.0944
Af0(s): 0.13637542725E-003
Af1(s/s): 0.10913936421E-010
week: 1406

**** Week 1406 almaNAU for PRN-05 ****

ID: 66
Health: 0
Eccentricity: 0.67987442017E-002
Time of Applicability(s): 405504.0000
Orbital Incluation(rad): 0.9773843811
Rate of Right Ascen(r/s): -0.80574784835E-008
SQRT(A) (m^{1/2}): 5481.73472
Right Ascen at Week(rad): 0.0
Argument of Perigee(rad): 0.785398
Mean Anom(rad): 2.7925
Af0(s): 0.49304962158E-003
Af1(s/s): 0.10913936421E-010
week: 1406

**** Week 1406 almaNAU for PRN-06 ****

ID: 67
Health: 0
Eccentricity: 0.59747695923E-002
Time of Applicability(s): 405504.0000

Orbital Incluation(rad): 0.9773843811
Rate of Right Ascen(r/s): -0.81603399109E-008
SQRT(A) (m^{1/2}): 5481.73472
Right Ascen at Week(rad): 0.0
Argument of Perigee(rad): 0.785398
Mean Anom(rad): -2.7925
Af0(s): 0.28324127197E-003
Af1(s/s): 0.36379788071E-010
week: 1406

**** Week 1406 almaNAU for PRN-07 ****

ID: 68
Health: 0
Eccentricity: 0.11693477631E-001
Time of Applicability(s): 405504.0000
Orbital Incluation(rad): 0.9773843811
Rate of Right Ascen(r/s): -0.81031946734E-008
SQRT(A) (m^{1/2}): 5481.73472
Right Ascen at Week(rad): 0.0
Argument of Perigee(rad): 0.785398
Mean Anom(rad): -2.0944
Af0(s): 0.48542022705E-003
Af1(s/s): 0.72759576142E-011
week: 1406

**** Week 1406 almaNAU for PRN-08 ****

ID: 69
Health: 0
Eccentricity: 0.95143318176E-002
Time of Applicability(s): 405504.0000
Orbital Incluation(rad): 0.9773843811
Rate of Right Ascen(r/s): -0.81831980059E-008
SQRT(A) (m^{1/2}): 5481.73472
Right Ascen at Week(rad): 0.0
Argument of Perigee(rad): 0.785398
Mean Anom(rad): -1.3963
Af0(s): -0.55313110352E-004
Af1(s/s): 0.00000000000E+000
week: 1406

**** Week 1406 almaNAU for PRN-09 ****

ID: 70

Health: 0
Eccentricity: 0.17513751984E-001
Time of Applicability(s): 405504.0000
Orbital Inclination(rad): 0.9773843811
Rate of Right Ascen(r/s): -0.83546337183E-008
SQRT(A) (m^{1/2}): 5481.73472
Right Ascen at Week(rad): 0.0
Argument of Perigee(rad): 0.785398
Mean Anom(rad): -0.69813
Af0(s): 0.57220458984E-005
Af1(s/s): 0.00000000000E+000
week: 1406

**** Week 1406 almaNAU for PRN-10 ****

ID: 71
Health: 0
Eccentricity: 0.0
Time of Applicability(s): 405504.0000
Orbital Inclination(rad): 0.9773843811
Rate of Right Ascen(r/s): -0.78174684861E-008
SQRT(A) (m^{1/2}): 5481.73472
Right Ascen at Week(rad): 2.09439
Argument of Perigee(rad): 0.785398
Mean Anom(rad): 0.0
Af0(s): 0.34332275391E-004
Af1(s/s): 0.36379788071E-011
week: 1406

**** Week 1406 almaNAU for PRN-11 ****

ID: 72
Health: 0
Eccentricity: 0.91261863708E-002
Time of Applicability(s): 405504.0000
Orbital Inclination(rad): 0.9773843811
Rate of Right Ascen(r/s): -0.82974884808E-008
SQRT(A) (m^{1/2}): 5481.73472
Right Ascen at Week(rad): 2.09439
Argument of Perigee(rad): 0.785398
Mean Anom(rad): 0.69813
Af0(s): -0.20027160645E-004
Af1(s/s): 0.00000000000E+000
week: 1406

**** Week 1406 almaNAU for PRN-12 ****

ID: 73
Health: 0
Eccentricity: 0.78430175781E-002
Time of Applicability(s): 405504.0000
Orbital Inclination(rad): 0.9773843811
Rate of Right Ascen(r/s): -0.82974884808E-008
SQRT(A) (m^{1/2}): 5481.73472
Right Ascen at Week(rad): 2.09439
Argument of Perigee(rad): 0.785398
Mean Anom(rad): 1.3963
Af0(s): 0.71525573730E-004
Af1(s/s): 0.36379788071E-011
week: 1406

**** Week 1406 almaNAU for PRN-13 ****

ID: 74
Health: 0
Eccentricity: 0.74453353882E-002
Time of Applicability(s): 405504.0000
Orbital Inclination(rad): 0.9773843811
Rate of Right Ascen(r/s): -0.83089175283E-008
SQRT(A) (m^{1/2}): 5481.73472
Right Ascen at Week(rad): 2.09439
Argument of Perigee(rad): 0.785398
Mean Anom(rad): 2.0944
Af0(s): 0.13637542725E-003
Af1(s/s): 0.10913936421E-010
week: 1406

**** Week 1406 almaNAU for PRN-14 ****

ID: 75
Health: 0
Eccentricity: 0.67987442017E-002
Time of Applicability(s): 405504.0000
Orbital Inclination(rad): 0.9773843811
Rate of Right Ascen(r/s): -0.80574784835E-008
SQRT(A) (m^{1/2}): 5481.73472
Right Ascen at Week(rad): 2.09439
Argument of Perigee(rad): 0.785398
Mean Anom(rad): 2.7925

Af0(s): 0.49304962158E-003
Af1(s/s): 0.10913936421E-010
week: 1406

**** Week 1406 almaNAU for PRN-15 ****

ID: 76
Health: 0
Eccentricity: 0.59747695923E-002
Time of Applicability(s): 405504.0000
Orbital Incluation(rad): 0.9773843811
Rate of Right Ascen(r/s): -0.81603399109E-008
SQRT(A) (m^{1/2}): 5481.73472
Right Ascen at Week(rad): 2.09439
Argument of Perigee(rad): 0.785398
Mean Anom(rad): -2.7925
Af0(s): 0.28324127197E-003
Af1(s/s): 0.36379788071E-010
week: 1406

**** Week 1406 almaNAU for PRN-16 ****

ID: 77
Health: 0
Eccentricity: 0.11693477631E-001
Time of Applicability(s): 405504.0000
Orbital Incluation(rad): 0.9773843811
Rate of Right Ascen(r/s): -0.81031946734E-008
SQRT(A) (m^{1/2}): 5481.73472
Right Ascen at Week(rad): 2.09439
Argument of Perigee(rad): 0.785398
Mean Anom(rad): -2.0944
Af0(s): 0.48542022705E-003
Af1(s/s): 0.72759576142E-011
week: 1406

**** Week 1406 almaNAU for PRN-17 ****

ID: 78
Health: 0
Eccentricity: 0.95143318176E-002
Time of Applicability(s): 405504.0000
Orbital Incluation(rad): 0.9773843811
Rate of Right Ascen(r/s): -0.81831980059E-008
SQRT(A) (m^{1/2}): 5481.73472

Right Ascen at Week(rad): 2.09439
Argument of Perigee(rad): 0.785398
Mean Anom(rad): -1.3963
Af0(s): -0.55313110352E-004
Af1(s/s): 0.00000000000E+000
week: 1406

**** Week 1406 almaNAU for PRN-18 ****

ID: 79
Health: 0
Eccentricity: 0.17513751984E-001
Time of Applicability(s): 405504.0000
Orbital Incluation(rad): 0.9773843811
Rate of Right Ascen(r/s): -0.83546337183E-008
SQRT(A) (m^{1/2}): 5481.73472
Right Ascen at Week(rad): 2.09439
Argument of Perigee(rad): 0.785398
Mean Anom(rad): -0.69813
Af0(s): 0.57220458984E-005
Af1(s/s): 0.00000000000E+000
week: 1406

**** Week 1406 almaNAU for PRN-19 ****

ID: 80
Health: 0
Eccentricity: 0.0
Time of Applicability(s): 405504.0000
Orbital Incluation(rad): 0.9773843811
Rate of Right Ascen(r/s): -0.78174684861E-008
SQRT(A) (m^{1/2}): 5481.73472
Right Ascen at Week(rad): -2.09439
Argument of Perigee(rad): 0.785398
Mean Anom(rad): 0.0
Af0(s): 0.34332275391E-004
Af1(s/s): 0.36379788071E-011
week: 1406

**** Week 1406 almaNAU for PRN-20 ****

ID: 81
Health: 0
Eccentricity: 0.91261863708E-002
Time of Applicability(s): 405504.0000

Orbital Incluation(rad): 0.9773843811
Rate of Right Ascen(r/s): -0.82974884808E-008
SQRT(A) (m^{1/2}): 5481.73472
Right Ascen at Week(rad): -2.09439
Argument of Perigee(rad): 0.785398
Mean Anom(rad): 0.69813
Af0(s): -0.20027160645E-004
Af1(s/s): 0.00000000000E+000
week: 1406

**** Week 1406 almaNAU for PRN-21 ****

ID: 82
Health: 0
Eccentricity: 0.78430175781E-002
Time of Applicability(s): 405504.0000
Orbital Incluation(rad): 0.9773843811
Rate of Right Ascen(r/s): -0.82974884808E-008
SQRT(A) (m^{1/2}): 5481.73472
Right Ascen at Week(rad): -2.09439
Argument of Perigee(rad): 0.785398
Mean Anom(rad): 1.3963
Af0(s): 0.71525573730E-004
Af1(s/s): 0.36379788071E-011
week: 1406

**** Week 1406 almaNAU for PRN-22 ****

ID: 83
Health: 0
Eccentricity: 0.74453353882E-002
Time of Applicability(s): 405504.0000
Orbital Incluation(rad): 0.9773843811
Rate of Right Ascen(r/s): -0.83089175283E-008
SQRT(A) (m^{1/2}): 5481.73472
Right Ascen at Week(rad): -2.09439
Argument of Perigee(rad): 0.785398
Mean Anom(rad): 2.0944
Af0(s): 0.13637542725E-003
Af1(s/s): 0.10913936421E-010
week: 1406

**** Week 1406 almaNAU for PRN-23 ****

ID: 84

Health: 0
Eccentricity: 0.67987442017E-002
Time of Applicability(s): 405504.0000
Orbital Inclination(rad): 0.9773843811
Rate of Right Ascen(r/s): -0.80574784835E-008
SQRT(A) (m^{1/2}): 5481.73472
Right Ascen at Week(rad): -2.09439
Argument of Perigee(rad): 0.785398
Mean Anom(rad): 2.7925
Af0(s): 0.49304962158E-003
Af1(s/s): 0.10913936421E-010
week: 1406

**** Week 1406 almaNAU for PRN-24 ****

ID: 85
Health: 0
Eccentricity: 0.59747695923E-002
Time of Applicability(s): 405504.0000
Orbital Inclination(rad): 0.9773843811
Rate of Right Ascen(r/s): -0.81603399109E-008
SQRT(A) (m^{1/2}): 5481.73472
Right Ascen at Week(rad): -2.09439
Argument of Perigee(rad): 0.785398
Mean Anom(rad): -2.7925
Af0(s): 0.28324127197E-003
Af1(s/s): 0.36379788071E-010
week: 1406

**** Week 1406 almaNAU for PRN-25 ****

ID: 86
Health: 0
Eccentricity: 0.11693477631E-001
Time of Applicability(s): 405504.0000
Orbital Inclination(rad): 0.9773843811
Rate of Right Ascen(r/s): -0.81031946734E-008
SQRT(A) (m^{1/2}): 5481.73472
Right Ascen at Week(rad): -2.09439
Argument of Perigee(rad): 0.785398
Mean Anom(rad): -2.0944
Af0(s): 0.48542022705E-003
Af1(s/s): 0.72759576142E-011
week: 1406

**** Week 1406 almaNAU for PRN-26 ****

ID: 87
Health: 0
Eccentricity: 0.95143318176E-002
Time of Applicability(s): 405504.0000
Orbital Inclination(rad): 0.9773843811
Rate of Right Ascen(r/s): -0.81831980059E-008
SQRT(A) (m^{1/2}): 5481.73472
Right Ascen at Week(rad): -2.09439
Argument of Perigee(rad): 0.785398
Mean Anom(rad): -1.3963
Af0(s): -0.55313110352E-004
Af1(s/s): 0.00000000000E+000
week: 1406

**** Week 1406 almaNAU for PRN-27 ****

ID: 88
Health: 0
Eccentricity: 0.17513751984E-001
Time of Applicability(s): 405504.0000
Orbital Inclination(rad): 0.9773843811
Rate of Right Ascen(r/s): -0.83546337183E-008
SQRT(A) (m^{1/2}): 5481.73472
Right Ascen at Week(rad): -2.09439
Argument of Perigee(rad): 0.785398
Mean Anom(rad): -0.69813
Af0(s): 0.57220458984E-005
Af1(s/s): 0.00000000000E+000
week: 1406

**** Week 1406 almaNAU for PRN-28 ****

ID: 89
Health: 0
Eccentricity: 0.10948657990E-001
Time of Applicability(s): 405504.0000
Orbital Inclination(rad): 0.9773843811
Rate of Right Ascen(r/s): -0.77717522961E-008
SQRT(A) (m^{1/2}): 5481.73472
Right Ascen at Week(rad): 0.0
Argument of Perigee(rad): 0.785398
Mean Anom(rad): 0.3

Af0(s): 0.34332275391E-004
Af1(s/s): 0.00000000000E+000
week: 1406

**** Week 1406 almaNAU for PRN-29 ****

ID: 90
Health: 0
Eccentricity: 0.93865394592E-002
Time of Applicability(s): 405504.0000
Orbital Incluation(rad): 0.9773843811
Rate of Right Ascen(r/s): -0.77946103911E-008
SQRT(A) (m^{1/2}): 5481.73472
Right Ascen at Week(rad): 2.09439
Argument of Perigee(rad): 0.785398
Mean Anom(rad): 0.2
Af0(s): 0.51975250244E-003
Af1(s/s): 0.14551915228E-010
week: 1406

**** Week 1406 almaNAU for PRN-30 ****

ID: 91
Health: 0
Eccentricity: 0.93865394592E-002
Time of Applicability(s): 405504.0000
Orbital Incluation(rad): 0.9773843811
Rate of Right Ascen(r/s): -0.77946103911E-008
SQRT(A) (m^{1/2}): 5481.73472
Right Ascen at Week(rad): -2.09439
Argument of Perigee(rad): 0.785398
Mean Anom(rad): 0.2
Af0(s): 0.51975250244E-003
Af1(s/s): 0.14551915228E-010
week: 1406

ПЕРЕЧЕНЬ ССЫЛОК

1. Бабак В. П., Конін В. В., Харченко В. П. Супутникова радіонавігація. – К.: Техніка, 2004. – 328 с.
2. *Глобальная* навигационная спутниковая система ГЛОНАСС (Интерфейсный контрольный документ, пятая редакция). – М.: Координационный научно – информационный центр Российской федерации, 2002. – 57 с.
3. *ГЛОНАСС*. Принципы построения и функционирования/Под ред. А. И. Перова, В. Н. Харисова. Изд. 3-е, перераб.- М.: Радиотехника, 2005.- 687 с.
4. *Гофман-Велленгоф В.* Глобальна система визначення місцеположення (GPS): Теорія і практика/ В. Гофман-Велленгоф, Г. Ліхтенеггер, Д. Коллінз / Пер. з англ.; За ред. акад. Я. С. Яцківа. – К.: Наук. думка, 1996. – 391 с.
5. *Дьяконов В. П.* MATLAB 6/6.1/6. 5+ Simulink 4/5. Основы применения. Полное руководство пользователя.- М. :СОЛОН Пресс, 2002.- 768 с.
6. *Дьяконов В. П.* MATLAB 6. 5 SP1/7. 0+ Simulink 5/6. Обработка сигналов и проектирование фильтров.- М. :СОЛОН Пресс, 2005.- 576 с.
7. *Руководство по всемирной геодезической системе (WGS-84).* – ICAO. Doc. 9674 - AN/946. – Монреаль, 1997. – 112 с.
8. *Соловьев Ю. А.* Системы спутниковой навигации. – М.: ЭКО – ТРЕНДЗ, 2000. – 268 с.
9. *Interface Control Document Global Positioning System (ICD-GPS-200C).* Wash., 1997. – 160 p.
10. OEM4 Family Installation and Operation User Manual Rev 11, Vol 1. NovAtel Inc., Canada, 2004.- 202 p.
11. OEM4 Family Installation and Operation User Manual Rev 11, Vol 2. NovAtel Inc., Canada, 2004.- 280 p.