Задание на семестр:

1. Найдем X,Y,Z; Если известно h,b,L;
2. Придумать XYZ для 24 х спутников Глонасс, R=6400км+19100, размещение равномерное
3. Вывести алгоритм расчета видимости
4. Расчет H, Dn, СКО

01.10.2020

## Описание движения спутников ГЛОНАСС по орбитам

Координаты центра спутника изменяются в зависимости от положения спутника на орбите

Координаты орбит:

|  |  |
| --- | --- |
|  | (1.1) |
| Где: |  |

 - угол перигея,

 - истинное значение аномалии

 - долгота восходящего узла

Значение  и  берутся из приложение 1.

Уравнение орибты спутника в орбитальной плоскости в полярной системе координат с центром, совпадающим с центром Земли:

, (1.2)

где p- фокальный параметр,

e- эксцентриситет,

 - угол между положительным направлением полярной оси и фокальной осью ( при  = 0 полярная ось направлена от центра к ближайшей вершине кривой, а при  - в противоположную сторону, в дальнейшем расчете полагаем, что  = 0)

 - угол наклона орбиты.

## Расчет выбранной точки в декартовой системе координат

По заданию, необходимо определить координаты выбранной точки: (55.819715, 37.611680) - координаты наблюдателя.

Переведем координаты точки из геодезической системы координат (H,L,B) в декартовую систему координат (X,Y,Z) , для этого используем формулы:

 ,

,

.

Где  - радиус кривизны в точке местной вертикали.

a – параметр сжатия эллипсоида;

H= 200 м; - высота над уровнем моря потребителя

L=37.611680; - долгота

B=55.819715 – широта.

Тогда координаты примут вид: (2838598.372491, 2186937.017082, 5276631.111979)

1. Расчет видимости спутников

На рисунке 1 представлен угол (α) между нормалью к потребителю и спутником, зная который, можно определить угол места (90° - α), по которому можно определить наличие видимости навигационного спутника.

Y0

Z1

α

НС

X0

П

Рисунок 1 − Угол между нормалью к цели и спутником

Исходя из наших данных, для расчета видимости спутников введем вектор единичной длины в направление на НКА, тогда он будет равен:

|  |  |
| --- | --- |
|  | (3.1) |

Где 

Так как данные координат даны в геодезической системе координат для расчета необходим перевести в топоцентрические координаты (ENU) *[Переведем координаты точки из геодезической системы координат (H,L,B) в декартовую систему координат (X,Y,Z) ]* – и коэффициентом для перевода из одной системы в другую будет являться матрица :

|  |  |
| --- | --- |
|  | (3.2) |

Теперь умножим вектор единичной длины в направление на НКА, который является вектором столбцом с размером , на получившийся коэффициент перевода, то есть матрицу  с размером , и в ответе получим вектор столбец  с размером :

|  |  |
| --- | --- |
|  | (3.3) |

Тогда угол между нормалью к потребителю и спутником определяется по  или z составляющей вектора  (которая рассчитывается по формуле 3.3):

|  |  |
| --- | --- |
|  | (3.4) |

Если (90-, то спутник находится в области видимости.

1. Нахождения оценки координаты потребителя с учетом ошибок SISRE

Из пункта 1.2.1 следует, что решением данной задачи является матрица  на диагонали которой будут лежать дисперсии компонент вектора состояний .

И матрица принимает вид: 

Для решение данной задачи необходимо найти вектор H, Dn

Матрица **H** принимает вид:

|  |  |
| --- | --- |
|  | (4.1) |

где:

|  |  |
| --- | --- |
|  | (4.2) |

Матрица **** равна:

|  |  |
| --- | --- |
|  | (4.3) |

где n зависит от количества видимых спутников

Матрица  будет равна:



Тогда выражение примет вид:





На диагонали лежат дисперсии компонент вектора состояний .

**ГЛАВА 2. ДОПОЛНЕНИЯ К ПРОГРАММЕ**

*По пункту 1.*

Для ввода значения H,L,B воспользуемся отдельным классом «wxTextCtrl»

Класс wxTextCtrl – это текстовое поле, куда можно вводить и выводить значения

Воспользуемся в нем функцией ввода значения и перевод в тип данных – double «GetValue().ToDouble()»

Расчет координат спутника

Для расчета координат создадим новую функцию «ephemerids» которая будет рассчитывать в определенных момент времени координаты выбранного спутника (n).

Так как нам необходимо возвращать из этой функции данные координат, значит функция примет вид int ephemerids(int n, int t). Для вывода трех значений – X,Y,Z воспользуемся дополнительным типом данных – структуры.

Значения долготы восходящего узла и зенитного угла для 24-х спутников (берется из учебника Глонасс) (приложение 1).

Чтобы не загружать код программы, а также для удобства чтения кода, функция для нахождения координат спутника будет создана в отдельном файле - «ephemerids.c» где будет лежать основной алгоритм расчета, а также создадим файл «ephemerids.h», который будет являться набором объявлений.

*По пункту 3.*

Для расчета видимости спутников необходимо знать, под каким углом к потребителю находится спутник. Для этого напишем алгоритм нахождения угла, основанный на пункте 3 «Расчет видимости спутников».

Чтобы не загружать код программы, а также для удобства чтения кода, функция для расчета видимости спутников будет создана в отдельном файле - «angle.c» где будет лежать основной алгоритм расчета, а также создадим файл «angle.h», который будет являться набором объявлений.

Функция «angle» примет тип данных double, так как в основном у нас все значения не целые.

XYZ🡪 ENU: С

ENU 🡪XYZ: С^-1 = C^T

R= вектор единичной длины, в напр. На НКА

Renu = C\*R=(Re, Rn, Ru)



*По пункту 4*

Для работы с матрицами воспользуемся встроенной библиотекой: “Armadillo”.

Объявим матрицу : “mat Dn;”

Чтобы не появились аномальные числа выделим необходимый нам массив данных и приравняем его к нулю: “Dn.zeros(inum, inum);”

Матрица Dn приняла вид: 

Заполним диагональ матрицы значениями: ( из п. 1.2.1).

Найдем обратную матрицу “Dn” и запишем ее в матрицу “Dninv”.

Так как матрица квадратная, воспользуемся функцией “inv”.

Функция Y = inv(A) вычисляет матрицу, обратную квадратной матрице A. В случаях, когда матрица A плохо масштабирована или близка к вырожденной, выдаются сообщения.

Следовательно, “Dninv = inv(Dn)”;

Объявим матрицу H: “mat H(inum, 4); H.zeros();” создается матрицу размером (inum, 4).

…

Теперь найдем матрицу **СКО:** “mat sko = Dninv \*H;”

Для вывода данных СКО добавим новый блок ctrltext…

**Библиотека “Armadillo”**

Armadillo - это программная библиотека линейной алгебры с открытым исходным кодом для языка программирования C++ . Он нацелен на обеспечение эффективных и оптимизированных базовых вычислений. Синтаксис намеренно похож на Matlab.

Поддерживаются целые числа, числа с плавающей точкой и комплексные числа, а также подмножество тригонометрических и статистических функций. Различные разложения матриц обеспечиваются за счет необязательной интеграции с LAPACK или одной из его высокопроизводительных замен (например, многопоточных библиотек MKL или ACML).

Подход с отложенной оценкой используется (во время компиляции) для объединения нескольких операций в одну и уменьшения (или устранения) потребности во временных. Это достигается с помощью рекурсивных шаблонов и шаблонного метапрограммирования.

Приложение 1.

|  |  |  |
| --- | --- | --- |
| Номер спутника | , гр. | , гр. |
| 1 | 0 | 0 |
| 2 | -45 | 0 |
| 3 | -90 | 0 |
| 4 | -135 | 0 |
| 5 | -180 | 0 |
| 6 | -225 | 0 |
| 7 | -270 | 0 |
| 8 | -315 | 0 |
| 9 | 15 | 120 |
| 10 | -30 | 120 |
| 11 | -75 | 120 |
| 12 | -120 | 120 |
| 13 | -165 | 120 |
| 14 | -210 | 120 |
| 15 | -255 | 120 |
| 16 | -300 | 120 |
| 17 | 30 | 240 |
| 18 | -15 | 240 |
| 19 | -60 | 240 |
| 20 | -105 | 240 |
| 21 | -150 | 240 |
| 22 | -195 | 240 |
| 23 | -240 | 240 |
| 24 | -285 | 240 |