Тестовое задание: задачи 1 и 9

12 октября 2023 г.

СОДЕРЖАНИЕ

1 ЗАДАНИЕ 1 2

2 ЗАДАНИЕ 9 4

1 ЗАДАНИЕ 1

Рассмотрим основные моменты, которые были исправлены в файле *Constellation.m*. Все дальнейшие замечания, по большей части, касаются логики кода, а не формальностей оформления. Указанные строки соответствуют файлу до исправлений.

Строки 8-11: константы необходимо убрать из класса по ряду причин. Во-первых, в том случае, если необходимо ими воспользоваться где-то в другом месте, обязательно сначала надо создать объект класса Constellation и только потом станут доступны константы. Во-вторых, если использовать константы, путём их прямого указания в других файлах, то между разными частями проекта может возникнуть несогласованность по точности и размерности используемых констант. Соответственно, предлагается решение с созданием mat-файла, из которого потом, по мере необходимости, подгружаются константы в код. Функция, генерирующая файл с константами - saveConstantToFile.m.

Строки 16-21:

```
function this = Constellation(varargin)
    if isempty(varargin)
        return
    end
    this.loadFromConfigFile(varargin{1});
end
```

использование *varargin* в данном случае нецелесообразно, так как далее потребуется string - в **строке 30** используется *strcmpi*, который при сравнении, например, двух единиц типа double или ‘1’ и 1, всё равно вернёт ноль. Соответственно, некорректный ввод в конструктор может привести к неправильному формированию класса (ненахождению нужной записи в файле).

Строки 29-34:

```
for i = 1:length(data)
    if strcmpi(data(i).name, code)
        dataThis = data(i);
        break
    end
end
```

использование *i* в качестве счётчика приведёт к переопределению стандартной переменной мнимой единицы. Аналогичные замечания к **строкам 41-53, 77-83**.

Строки 60-66:

```
for group = this.groups
    for i = 1:length(group{1})
        ending = shift + group{1}(i).totalSatCount - 1;
        this.state.elements(shift:ending,:) = this.
           getInitialElements(group{1});
        shift = ending + 1;
    end
end
```

использование второго for-цикла (вложенного) нецелесообразно, так как в каждую ячейку *this.groups* уже кладётся строго одна группа при расшифровке json-файла. Поэтому, считаю, нет необходмости переусложнять код.

Строки 69-85:

– Заменены имена переменных для массивов с оканчивающихся на ‘s’ (*raans*) на оканчивающиеся на ‘Array’ (*raanArray*).
– То, что носит смысл счётчика, заменено с *idx* на *counter*.
– Величина *raanIDX* приведена к правилу ‘lowerCamelCase’.

2 ЗАДАНИЕ 9

Для формирования равномерной сетки точек на сфере, используется часть реализации кода отсюда. После этого на сферу “натягивается” оболочка из треугольников с вершинами в заданных ранее точках. Так как для дальнейшей работы важно триангулировать сферу, то процедуры равномерного замощения сферы точками и разбиения на треугольные сектора совмещены в одну функцию.

Дальше реализован код, который напрямую считает вектора от центров секторов (за центр треугольника принят его центр масс) до геостационарных спутников и спутников орбитальной группировки.

Затем для каждого сектора определяются геостационарные спутники, которые из него видны, то есть угол между нормалью к сектору и вектором на геостационар меньше $\frac{\pi}{2}$.

Эта информация после используется для сокращения числа вычислений при определении угла из центра сектора между спутником и геостационаром - к невидимым геостационарам углы не считаются.

Дальше есть несколько путей развития кода - сокращение времени вычислений или сокращение затрат памяти. Я выбрал промежуточный вариант, когда в цикле проходятся по всем спутникам орбитальной группировки, не отсеивая “лишние”. Под “лишними” стоит понимать такие спутники группировки, вектор из центра сегмента на которые образует с нормалью к сегменту угол более $\frac{\pi}{2} + \alpha$, где α - критический угол (в условии текущей задачи это 2 градуса).

Если в цикле сокращать число спутников группировки, которые надо будет считать для каждого сегмента, то это может сократить затрачиваемые объёмы памяти, но может увеличить время работы из-за дополнительных операций.

Перед циклом сократить объём вычислений труднее, так как или необходимо создавать матрицы большой размерности (64000*2000*3 - для double размер такой матрицы $\sim$ 3Гб), или снова использовать цикл. А если памяти достаточно, то можно напрямую создать матрицы большой размерности, которые позволят без циклов посчитать все необходимые величины сразу для всех секторов, спутников и геостационаров.

После всех вычислений данные записываются в массив структур. В данном массиве каждая структура соответствует своему сектору на сфере - порядковый номер структуры в массиве совпадает с номером сегмента. Структуры содержат два поля: *satName* и *relGeoSatName*. В *satName* содержатся номера спутников орбитальной группировки, которые удовлетворяют поставленной задаче, а в *relGeoSatName* содержатся соответствующие номе-

рам из *satName* номера геостационарных спутников, с которыми КА орбитальной группы составляют угол менее 2 градусов при взгляде из центра сектора.