§2 Второй отборочный этап

2.1 Задачи по физике

Задача 2.1.1 (5 баллов)

Спутник находится на ретроградной круговой экваториальной орбите Земли на высоте 500 км.
Двигатель коррекции орбиты спутника имеет удельный импульс 344 с, тяга может быть направлена в любую сторону. Масса спутника 1000 кг.
Какую минимальную массу топлива должен затратить спутник, чтобы перейти на круговую орбиту с наклонением 175 градусов и высотой 350 км? Ответ укажите в килограммах, с точностью до килограмма.

Допущения:
Считать, что выше 300 км атмосфера никак не влияет на движение спутника.
Считать Землю идеальным шаром.
Прецессией вращения Земли пренебречь.
Маневры спутника совершаются дискретными импульсами. Длительность импульса по сравнению с периодом обращения спутника считать ничтожной.

* Все высоты даны от поверхности Земли.

Решение:

Начальные данные:
$\mu = 398.6 \cdot 10^{12}$ м^3/с^2 – гравитационный параметр
$m = 1000$ кг – масса спутника
$I = 344 \cdot 9.8$ м/с – удельный импульс двигателя
$r_1 = 6721 \cdot 10^3$ м – радиус конечной круговой орбиты
$r_0 = 6871 \cdot 10^3$ м – радиус начальной круговой орбиты
$i = \frac{175 \cdot \pi}{180}$ – наклонение орбиты

Ход решения:
Переход на эллиптическую орбиту осуществляется в два этапа:

1) Изменение наклонения орбиты
2) Двухимпульсный переход на круговую орбиту другого радиуса

Однако для экономии топлива будет целесообразным совместить изменение наклонения с первым импульсом перехода.

Скорость на начальной круговой орбите:

$$v_0 = \sqrt{\frac{\mu}{r_0}} = 7616.556 \text{ м/с}$$

Скорость на переходной эллиптической орбите:

$$v_1 = \sqrt{\mu\left(\frac{2}{r_0} - \frac{2}{r_0 + r_1}\right)} = 7574.412 \text{ м/с}$$

Вектор изменения скорости, требуемый для изменения наклонения орбиты и первого импульса торможения, определяем с помощью теоремы косинусов:

$$\Delta v_1 = \sqrt{{v_0}^2 + {v_1}^2 + 2v_0 v_1 \cdot \cos(\pi - i)} = 663.957 \text{ м/с}$$

Затрачиваемая масса топлива на данное изменение скорости:

$$\Delta m_1 = m\left(1 - \frac{1}{exp\left(\frac{\Delta v_1}{I}\right)}\right) = 178.366 \text{ кг}$$

Обозначим отношение векторов орбит следующим образом:

$$r = \frac{r_1}{r_0}$$

Изменение скорости, требуемое на второй импульс для перехода на круговую орбиту, рассчитывается следующим образом:

$$\Delta v_2 = \sqrt{\mu\left(\mu\left(\frac{2}{r_0} - \frac{2}{r_0 + r_1}\right)\right)} - \sqrt{\frac{\mu}{r_1}} = 42.378 \text{ м/с}$$

Затрачиваемая масса топлива:

$$\Delta m = (m - \Delta m_1)\left(1 - \frac{1}{exp\left(\frac{\Delta v}{I}\right)}\right) = 188.886 \text{ кг}$$

Ответ: 189 кг

Критерий оценки:
Решение на платформе Stepik принималось только от капитанов команд. Было введено дисконтирование баллов в зависимости от числа попыток:

$$n = \frac{a}{N}$$

где a – максимальное количество баллов, N – количество попыток, n – балл за задачу.

Задача 2.1.2 (5 баллов)

Спутник находится на ретроградной круговой экваториальной орбите Земли на высоте 500 км.
Двигатель коррекции орбиты спутника имеет удельный импульс 344 с, тяга может быть направлена в любую сторону. Масса спутника 1000 кг.

Какую минимальную массу топлива должен затратить спутник, чтобы перейти на эллиптическую орбиту с наклонением 174 градуса, аргументом перицентра 80 градусов, высотой в перигее 350 км, у которой восходящий узел совпадает с одной из точек исходной орбиты? Ответ укажите в килограммах, с точностью до килограмма.

Допущения:
Считать, что выше 300 км атмосфера никак не влияет на движение спутника.
Считать Землю идеальным шаром.
Прецессией вращения Земли пренебречь.
Маневры спутника совершаются дискретными импульсами. Длительность импульса по сравнению с периодом обращения спутника считать ничтожной.

* Все высоты даны от поверхности Земли.

Решение:

Начальные данные:
$\mu = 398.6 \cdot 10^{12}$ м^3/с^2 – гравитационный параметр
$m = 1000$ кг – масса спутника
$I = 344 \cdot 9.8$ м/с – удельный импульс двигателя
$r_1 = 6721 \cdot 10^3$ м – перигей эллиптической орбиты
$r_0 = 6871 \cdot 10^3$ м – радиус круговой орбиты
$i = \frac{174 \cdot \pi}{180}$ – наклонение орбиты
$\omega = \frac{80 \cdot \pi}{180}$ – аргумент перицентра

Ход решения:
Переход на эллиптическую орбиту осуществляется в два этапа:

1) Изменение наклонения орбиты
2) Переход с круговой орбиты на эллиптическую

Скорость на начальной круговой орбите:

$$v_0 = \sqrt{\frac{\mu}{r_0}} = 7616.556 \text{ м/с}$$

Вектор изменения скорости, требуемый для изменения наклонения орбиты, определяем по углу, на который необходимо повернуть орбиту. С помощью теоремы косинусов запишем:

$$\Delta v_i = v_0 \sqrt{2 \cdot (1 - \cos(\pi - i))} = 797.24 \text{ м/с}$$

Необходимым приращением скорости для изменения наклонения будет Δv_i.
Учитывая, что восходящий узел новой орбиты совпадает с одной из точек старой орбиты, запишем:

$$r_0 = a_1 \frac{1 - e_1^2}{1 + e_1 \cos \omega}$$

Далее, находим параметры орбиты из соотношения:

$$\begin{cases} r_0 = a_1 \dfrac{1 - e_1^2}{1 + e_1 \cos \omega} \\ r_1 = a_1(1 - e_1) \end{cases}$$

Отсюда:

$$a_1 = 6908.463 \cdot 10^3 \text{ м}$$
$$e_1 = 0.0271$$

Скорость в точке перехода:

$$v_1 = \sqrt{\mu \cdot \left(\frac{2}{r_0} - \frac{1}{a_1}\right)} = 7637.18 \text{ м/с}$$

Скорость в перигее:

$$v_p = \sqrt{\frac{\mu}{a_1} \cdot \frac{1 + e_1}{1 - e_1}} = 7804.592 \text{ м/с}$$

Из соотношения для определения перпендикулярной составляющей скорости:

$$v_1 \cdot \cos \varphi = v_p \frac{r_1}{r_0}$$

Находим $\cos \varphi$, где φ – угол между начальным и конечным векторами скорости:

$$\cos \varphi = \frac{v_p}{v_1} \cdot \frac{r_1}{r_0}$$

Далее с помощью теоремы косинусов определяем необходимое приращение скорости для перехода на эллиптическую орбиту:

$$\Delta v_1 = \sqrt{{v_0}^2 + {v_1}^2 - 2 \cdot v_0 \cdot \frac{v_p r_1}{r_0}} = 213.677 \text{ м/с}$$

Общее приращение скорости:

$$\Delta v = \sqrt{{\Delta v_i}^2 + {\Delta v_1}^2} = 825.378 \text{ м/с}$$

Затрачиваемая масса топлива:

$$\Delta m = m\left(1 - \frac{1}{exp\left(\frac{\Delta v}{I}\right)}\right) = 217.164 \text{ кг}$$

Ответ: 217 кг

Критерий оценки:

Решение на платформе Stepik принималось только от капитанов команд. Было введено дисконтирование баллов в зависимости от числа попыток:

$$n = \frac{a}{N}$$

где a – максимальное количество баллов, N – количество попыток, n – балл за задачу.

Задача 2.1.3 (15 баллов)

Спутник находится на ретроградной круговой экваториальной орбите Земли на высоте 500 км.
Двигатель коррекции орбиты спутника имеет удельный импульс 344 с, тяга может быть направлена в любую сторону. Масса спутника 1000 кг.

Перед спутником поставлена следующая задача: требуется сделать снимки городов Сурабая и Луанда.

Есть некоторые ограничения, с учётом которых он должен выполнить эту задачу:
1. В момент фотографирования необходимо находиться непосредственно над точкой съёмки.
2. Возможности оптики спутника таковы, что высота при съёмке должна быть не больше 400 км.
3. Чтобы избежать потерь на взаимодействие с атмосферой, спутник не может опускаться ниже 300 км.

Какую минимальную массу топлива должен затратить спутник, чтобы выполнить задачу? Опишите, какие маневры должен совершить спутник, чтобы выполнить задачу.

Допущения:
Считать, что выше 300 км атмосфера никак не влияет на движение спутника.
Считать Землю идеальным шаром.
Прецессией вращения Земли пренебречь.
Маневры спутника совершаются дискретными импульсами. Длительность импульса по сравнению с периодом обращения спутника считать ничтожной.

* Все высоты даны от поверхности Земли.

Решение:

1. Т.к. в задаче ничего не говорится о времени выполнения, то можем легко показать, что период любой орбиты можно сделать не кратным суткам, а значит задача превращается в поиск орбиты, у которой широты нужных нам точек находятся не дальше 400 и не ближе 300 км. Такая орбита всегда будет проходить над нужной широтой на необходимом расстоянии, а в какой-то момент из-за вращения Земли окажется в то же время и над нужной долготой. По той же причине можно не беспокоиться о том, будет ли искомая съёмка произведена днём.
2. Максимальная широта, на которую поднимается наш спутник равна углу поворота плоскости орбиты от изначальной. Чем больше угол поворота, тем при прочих равных больше нужен импульс.
3. Легко показать, что "экстремальными" орбитами являются те, у которых искомые широты либо ровно на высоте 400 км, либо в перигее. Таких траекторий всего несколько. Рассматриваем все возможные маневры для попадания на такие орбиты. Выбираем из них такой, где изменение импульса спутника было минимальным.

4. Краевыми случаями решения являются:

- случай перехода на эллиптическую орбиту, перигей которой лежит между заданными широтами;
- случай перехода на эллиптическую орбиту, перигей которой лежит строго на одной из широт.

5. Алгоритм решения задачи сводится к задаче минимизации, вытекающий из двух последовательных переходов:

- переход с круговой орбиты на эллиптическую;
- переход с опорной эллиптической орбиты на эллиптическую орбиту, удовлетворяющую условиям проведения съемки.

Используя любой математический пакет можно провести необходимые расчёты.

Код решения:

```
//численные данные
mu = 398.6*10^12 // гравитационный параметр
m = 1000 // масса спутника
Imp = 344*9.80665 // удельный импульс двигателя
lat1deg = 7.2431 // широта первого города
lat2deg = 8.853 // широта второго города
lat1 = 3.14159265359*lat1deg/180
lat2 = 3.14159265359*lat2deg/180
r1 = 6771*10^3
r0 = 6871*10^3
//задача перехода с исходной круговой орбиты на произвольную, заданную
fi и v
cospsi = Cos[lat2]*Cos[fi]
v0 = Sqrt[mu/r0]
dv1 = Sqrt(v0^2+v^2-2*v0*v*cospsi)
a1 = mu/(2*mu/r0-v^2)
e1 = Sqrt[1-r0^2*v^2*Cos[fi]*Cos[fi]/(mu*a1)]
omega1 = ArcCos[a1*(1-e1^2)/(e1*r0)-1/e1]
////////////////////////////////////////////
//переход на конечную орбиту с промежуточной, случай, когда перицентр
точно между нужными широтами
r = a1*(1-e1^2)/(1+e1*Cos[teta1])
v1= Sqrt[mu*(2/r-1/a1)]
rp1= a1*(1-e1)
vp1= Sqrt[mu*(2/rp1-1/a1)]
vtr1= vp1*rp1/r
vr1= Sqrt[v1^2-vtr1^2]
sina1= RealSign[Sin[teta1]]*vr1/v1
omega2= ArcCos[-Sin[lat1]/Sin[lat2]]/2
teta2= omega1 - omega2 + teta1
e2= (r1-r)/(r*Cos(teta2)-r1*Sin(omega2))
a2= r*(1+e2*Cos[teta2])/(1-e2^2)
v2= Sqrt[mu*(2/r-1/a2)]
rp2= a2*(1-e2)
vp2= Sqrt[mu*(2/rp2-1/a2)]
vtr2= vp2*rp2/r
vr2= Sqrt[v2^2-vtr2^2]
sina2= RealSign[Sin[teta2]]*vr2/v2
da= RealAbs[ArcSin[sina1]-ArcSin[sina2]]
dv2= Sqrt[v1^2+v2^2-2*v1*v2*Cos[da]]
////////////////////////////////////////////////
//переход на конечную орбиту с промежуточной, случай, когда перицентр
точно на одной из широт
r = a1*(1-e1^2)/(1+e1*Cos[teta1])
v1= Sqrt[mu*(2/r-1/a1)]
rp1= a1*(1-e1)
vp1= Sqrt[mu*(2/rp1-1/a1)]
vtr1= vp1*rp1/r
vr1= Sqrt[v1^2-vtr1^2]
sina1= RealSign[Sin[teta1]]*vr1/v1
```

```
omega2= 3.1415926/2
teta2= omega1 - omega2 + teta1
costeta3= Sin(lat1)/Sin(lat2)
e2= (r1-r)/(r*cos(teta2)-r1*costeta3)
a2= r*(1+e2*cos(teta2))/(1-e2^2)
v2= Sqrt(mu*(2/r-1/a2))
rp2= a2*(1-e2)
vp2= Sqrt(mu*(2/rp2-1/a2))
vtr2= vp2*rp2/r
vr2= Sqrt(v2^2-vtr2^2)
Sinalpha2= RealSign(Sin(teta2))*vr2/v2
dalpha= abs(aSin(Sinalpha1)-aSin(Sinalpha2))
dv2= Sqrt(v1^2+v2^2-2*v1*v2*cos(dalpha))
//////////////////
//финальные вычисления, функция, которую необходимо минимизировать:
dv = dv1 + dv2
m1= m/exp(dv/Imp)
dm= m - m1
```

2.2 Задача по информатике

Задача 2.2.1 (25 баллов)

Письма на Землю

Вася и Петя -- астронавты, находящиеся на космической станции “Дружба”. Станция выходит на связь с Землёй каждый день ровно в 16:00 и в состоянии передать сообщение, длиной не более 140 байт.

В том случае, если сообщение пишет Вася, сообщением является любовное письмо для Васиной возлюбленной Анжелы. Так как Анжела говорит по английски, то и письмо написано на английском языке с использованием латинских букв, а также пробелов, запятых, точек, вопросительных и восклицательных знаков. Вася старается быть кратким, но, тем не менее, его письма порой занимают все 140 байт допустимого сообщения.

Петя, в отличии от Васи, не обладает возлюбленной, и по этой причине сообщает на землю ценные экспериментальные данные. Ценные экспериментальные данные всегда представляют собой 28 четырехзначных чисел, разделенных запятой -- всего 139 символов.

Вася и Петя каждый день отдельно принимают решение о том, кто именно пошлет данные на землю, поэтому заранее предсказать, что именно будет в сообщении невозможно.

В результате разногласий между напарниками, начиная с некоторого момента на станции нарушилась тонкая юстировка передающего устройства. В результате, сообщения доходят на землю поврежденными: из каждых 8 байт сообщения от нуля до двух байтов подряд могут быть повреждены -- то есть в момент приёма содержать любой другой произвольный байт.

Ваша задача -- написать пару программ, кодировщик и декодер. При помощи кодировщика сообщение будет изменено на станции и передано. Затем, при передаче, сообщение будет повреждено. Затем, на земле, будет использован декодер для декодирования поврежденного сообщения. Если повреждения слишком существенны и восстановление невозможно, Вам потребуется сообщить об этом.

Описание входных и выходных данных

Кодировщик. В файле input.bin находится исходное сообщение. Размер файла не превышает 140 байт. Требуется записать закодированное сообщение в файл output.bin. Размер выходного файла должен быть равен в точности 140 байтам.
Пример:

input.bin	output.bin
I love You, Angela	SSBsb3ZlIFlvdSwgQW5nZWxhCg==

Декодировщик. В файле corrupted.bin находится принятое, возможно - поврежденное сообщение. Требуется записать восстановленное и декодированное сообщение в файл restored.bin. Размер каждого из файлов не должен превышать 140 байт. В случае невозможности восстановления, требуется написать строку “CANNOT BE RESTORED”.
Пример:

corrupted.bin	output.bin
SSBsb3ZlIFlvdSwgQW5nZWxhCg=2	I love You, Angela

Технические данные

Название Вашей программы должно быть letters.py. Язык программирования -- python 3.
Процедура проверки Вашего решения будет происходить следующим образом:

1. Будет создан файл input.bin. Ваша программа будет скопирована рядом с этим файлом.
2. Ваша программа будет запущена с параметром encode:

python3 letters.py encode

3. Затем выходной файл будет “исковеркан” по заранее определённой процедуре
4. Затем Ваша программа будет запущена с параметром decode:

python3 letters.py decode

5. После чего будет произведена проверка результатов.

Решение:

Исходное сообщение имеет длину до 140 байт:

- до 140 значимых символов в сообщении Васи.
- 112 значимых символов в сообщении Пети.

В сообщении содержится избыточная информация – реальной информации в каждом символе сообщения меньше, чем 8 бит:

- 6 бит на каждый символ в сообщении Васи.
- 4 бита на каждый символ в сообщении Пети.

Все сообщение можно сжать, и передать в меньшем количестве символов:

- 140*6/8 = 105 символов в сообщении Васи (простая кодировка каждого символа).
- 112*4/8 = 56 символов в сообщении Пети.

Если заполнить оставшиеся символы дополнительной проверочной информацией, то можно надеяться, что исходное сообщение удастся восстановить после искажения.

Пример программы

```
import reedsolo
import smaz
import sys
mode = sys.argv[1]
if mode == 'encode':
    f = open('input.bin', 'r')
    inp = f.read()
    f.close()
    print(inp)
    if inp[0].isdigit():
        inp = inp.replace(',', '')
        b = bytearray()
        while len(inp) > 0:
            num = int(inp[0:2])
            b.append(num)
            inp = inp[2:]
        rs = reedsolo.RSCodec(84)
        b = rs.encode(b)
    else:
        cmp = smaz.compress(inp)
        b = cmp.encode()
        rs = reedsolo.RSCodec(140-len(b))
        b = rs.encode(b)
    f = open('output.bin', 'wb')
    f.write(b)
    f.close()
else:
    f = open('corrupted.bin', 'rb')
    inp_b = f.read()
    f.close()
    out = ''
    try:
        rs = reedsolo.RSCodec(84)
        b = rs.decode(inp_b)
        i = 0
        while len(b) > 0:
            num = b[0]
            if num > 99:
                out = ''
                break
            num = str(num)
            if len(num) == 1:
                num = "0" + num
            out += num
            if i % 2 == 1:
                out += ','
            i += 1
            b = b[1:]
        out = out[:-1]
    except:
        pass
    if len(out) == 0:
        for i in range(139, 1, -1):
            try:
                rs = reedsolo.RSCodec(i)
                b = rs.decode(inp_b)
                out = smaz.decompress(b.decode())
                break
            except:
                continue
    print(out)
    f = open('output.bin', 'w')
    f.write(out)
    f.close()
```