

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

КАФЕДРА МЕХАНИКИ УПРАВЛЯЕМОГО ДВИЖЕНИЯ

**Бойцова Анастасия Ивановна**

**Выпускная квалификационная работа бакалавра**

**Движение космического аппарата с солнечным  
парусом в  $L_1$  системы Солнце-Земля**

Направление 010400

Прикладная математика и информатика

Научный руководитель,  
кандидат физ.-мат. наук,  
доцент  
Шиманчук Д. В.

Санкт-Петербург

2018

# Содержание

Введение .....	3
Постановка задачи.....	7
Обзор литературы.....	8
Глава 1. Исследование поведения решения линеаризованной системы. 9	
1.1. Уравнения движения.....	9
1.2. Функция опасности .....	12
1.3. Построение закона управления .....	15
Глава 2. Солнечный парус .....	18
2.1. Солнечный парус.....	18
2.2. Управление солнечным парусом .....	20
2.3. Численное моделирование движения КА с солнечным парусом.....	22
Выводы .....	26
Заключение.....	27
Список литературы .....	28
Приложение .....	30

## Введение

Под солнечным парусом [6]-[8], [10] мы понимаем приспособление, которое использует давление солнечного света на зеркальную поверхность для приведения в движение космического аппарата (КА).

Задачи, которые выполняются с помощью солнечного паруса:

- передвижение по солнечной системе;
- освещение Земли в полярную ночь;
- предсказание магнитных бурь;
- предупреждение о возникновении вспышек и катаклизм на Солнце;
- получение информации о состоянии солнечной атмосферы, глубинных слоях Солнца, солнечном ветре и об активности солнечной короны;
- осмотр комет, изучение звезд;
- изучение реликтового излучения;
- изучение космических лучей и многое другое.

Идея создания КА, использующего солнечный парус, разрабатывалась советским ученым Фридрихом Цандером. Он исходил из того, что частицы солнечного света (фотоны) имеют импульс и передают его любой освещаемой поверхности, создавая давление. Давление, которое создается фотонами заставит КА двигаться от Солнца, при этом не будет расходоваться ракетное топливо. При изменении угла расположения конструкции, можно корректировать направление полета. При большом удалении от Солнца фотонный поток слабеет пропорционально квадрату расстояния, а на границе системы упадет до 0.

Для того, чтобы обеспечить стабильный поток света и начальный разгон паруса, необходимы мощные лазерные установки. Основным минусом солнечного паруса является то, что он может двигать КА только в противоположную от Солнца сторону.

Материал, из которого сделан солнечный парус, должен быть

максимально легким и прочным. На данный момент самыми перспективными вариантами являются каптон и милар — тончайшие полимерные пленки с алюминиевым покрытием.

В настоящее время основной проблемой является развертывание солнечного паруса. Для этого приходится использовать дополнительные механические устройства.

В 1993 г. в рамках проекта «Знамя-2» на борту корабля «Прогресс М-15», пристыкованного к орбитальной станции «Мир» было произведено первое развертывание 20-метрового солнечного паруса. Задача этого эксперимента – освещение участка земной поверхности отраженным светом. В результате было создано яркое пятно 8 км в ширину.

Движение КА будем рассматривать в рамках задачи трех тел [1]-[3] — задача небесной механики, состоящая в определении относительного движения трех тел взаимодействующих по закону Всемирного тяготения. Круговая ограниченная задача трех тел предполагает, что одно из тел имеет бесконечно малую массу по сравнению с массами двух других тел, имеющих соизмеримые массы, не оказывает воздействия на их движение и два массивных тела движутся относительно своего центра масс по круговым орбитам.

Точки либрации (точки Лагранжа) [1]-[3] представляют собой частный случай при решении ограниченной круговой задачи трех тел. Существует пять точек либрации ( $L_i, i=\overline{1,5}$ ), в которых третье тело с бесконечно малой массой может оставаться неподвижным во вращающейся системе отсчета, связанной с массивными телами. В этих точках притяжение двух тел уравновешивает друг друга. Если поместить КА в любую из этих точек, то он останется в состоянии покоя. На рис. 1 показано расположение этих точек в системе Солнце-Земля, где  $M_1$  – Земля, а  $M_2$  – Солнце.

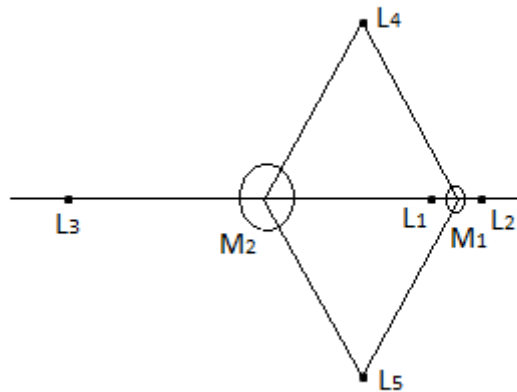


Рис. 1. Точки либрации

$L_1$ ,  $L_2$ ,  $L_3$  – называются прямолинейными точками либрации, расположенными на прямой, соединяющие центры масс Солнца и Земли. Эти точки являются неустойчивыми стационарными решениями уравнений движения. Спустя какое-то время КА, помещенный в эти точки будет покидать их окрестности и возникнет вопрос управления движения.

$L_4$  и  $L_5$  – называются треугольными точками либрации, образующие с телами равносторонние треугольники. Эти точки — устойчивые. КА, находящийся в их окрестности будет сохранять определенное положение относительно Солнца и Земли. В реальных условиях при различных возмущениях КА покинет окрестность любой из этих точек, но удержать его здесь можно с наименьшими затратами энергии, чем в любой другой точке пространства.

В точке  $L_1$  системы Солнце-Земля можно размещать космическую солнечную обсерваторию. Здесь она никогда не попадет в тень Земли, а, следовательно, наблюдения можно вести непрерывно. В  $L_1$  системы Солнце-Земля запущены:

- КА WIND, предназначенный для изучения солнечного ветра и его взаимодействия с поверхностью Земли;
- КА SOHO, для наблюдения за Солнцем;

- Advanced Composition Explorer, для исследования Солнца, энергетических частиц солнечного ветра и галактической материи.

В остальных точках:

- КА НАСА WMAP, изучающий реликтовое излучение ( $L_2$ );
- космические телескопы «Гершель», «Планк» и «Gaia» ( $L_2$ );
- два аппарата STEREO совершили транзит через точки  $L_4$  и  $L_5$ .

## Постановка задачи

В данной выпускной квалификационной работе рассматривается движение КА с использованием солнечного паруса в окрестности точки либрации  $L_1$ .

Пусть КА с солнечным парусом движется в окрестности точки либрации  $L_1$  системы Солнце-Земля согласно математической модели круговой ограниченной задачи трех тел [2]. Поставим задачу удержания КА с помощью солнечного паруса в окрестности коллинеарной точки либрации (неустойчивого стационарного решения уравнений движений). Для решения этой задачи будут пройдены следующие этапы:

1. Анализ поведения решений в окрестности точки либрации  $L_1$  системы Солнце-Земля.
2. Обоснование критерия удержания КА с солнечным парусом.
3. Построение законов управления для удержания КА в окрестности точки  $L_1$ .
4. Определение допустимого угла ориентации паруса.
5. Численное моделирование движения КА с солнечным парусом.

Целью работы является выявление зависимости ориентации солнечного паруса и времени его удержания в окрестности точки либрации  $L_1$ .

## Обзор литературы

В работах [1]-[3] показан вывод уравнений движения ограниченной задачи трех тел, а также найдены точки либрации  $L_i$ ,  $i=\overline{1,5}$  и проведен анализ устойчивости этих точек в линейном приближении.

В статьях [4] и [16] приводится понятие функции опасности, а также в [4] представлены законы управления, обеспечивающие удержание КА в окрестности коллинеарной точки либрации  $L_1$ .

В работе [5] доказано существование стабилизирующего управления для КА в окрестности коллинеарной точки либрации  $L_1$ .

В статьях [6]-[8] идет речь о солнечном парусе, о его преимуществах и недостатках. А также рассказана специфика управления солнечным парусом.

В статье [9] показан вывод уравнений движения солнечного паруса и найдены его положения равновесия. А также получено параметрическое семейство оригинальных решений в виде асимптотического ряда в области между Солнцем и Землей.

В работе [14] описана миссия NanoSail-D, цель которой состояла в развертывании солнечного паруса на орбите Земли.



# Глава 1. Исследование поведения решения линеаризованной системы

## 1.1. Уравнения движения

Пусть  $M_{1x_1x_2x_3}$  – вращающаяся (геоцентрическая) система координат. Ось  $M_{1x_1}$  направлена на Солнце. Движение  $M_1$  и  $M_2$  направлено против часовой стрелки.

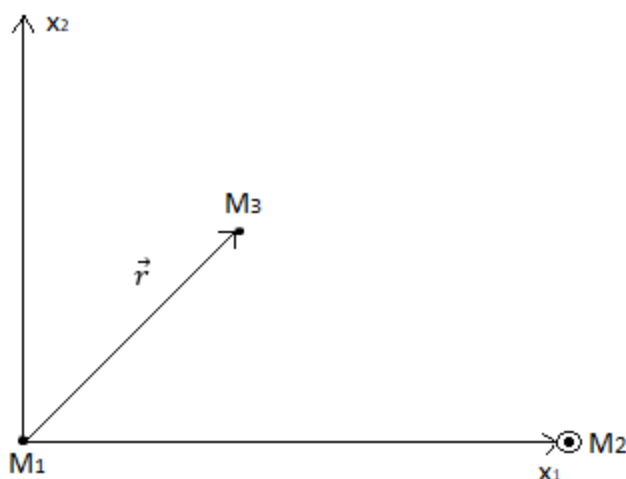


Рис. 2. Введенная система координат

Уравнения управляемого движения КА во вращающейся системе координат  $M_{1x_1x_2x_3}$ , в рамках хилловского приближения плоской круговой ограниченной задачи трех тел можно представить следующим образом [5]:

$$\begin{cases} \dot{x}_1 = x_2 + y_1, \\ \dot{x}_2 = -x_1 + y_2, \\ \dot{y}_1 = -\frac{3x_1}{\|x\|} + 2x_1 + y_2 + u_1, \\ \dot{y}_2 = -\frac{3x_2}{\|x\|} - x_2 - y_1 + u_2, \end{cases} \quad (1)$$

где  $x = (x_1, x_2)$  – вектор координат центра масс КА с солнечным парусом,  $\|*\|$  – евклидова норма,  $y = (y_1, y_2)$  – вектор скоростей КА,  $u = (u_1, u_2)$  – вектор управляющих воздействий, где компоненты представляют собой ускорения. Точка либрации  $L_1$  в системе  $M_{1x_1x_2x_3}$  неподвижна и имеет следующие координаты  $x^* = (1, 0)$  и  $y^* = (0, 1)$ .

В принятой модели за единицу расстояния принимается величина,

равная  $10^{-2}$  а.е.  $\approx 1.5 \cdot 10^6$  км. За единицу времени – 58.0916 суток (Земля совершает полный оборот вокруг Солнца за одну единицу). За единицу скорости – 298.659 м/с и за единицу ускорения –  $5.94649 \cdot 10^{-5}$  м/с<sup>2</sup>.

Общий вид системы (1):

$$\begin{cases} \frac{dx_i}{dt} = f_i(x, y), \\ \frac{dy_i}{dt} = g_i(x, y, u), \end{cases} \quad i = \overline{1,2}. \quad (2)$$

Приведем уравнения системы (1) к линеаризованному виду в окрестности точки  $L_1$ :

$$\frac{d\Delta x_1}{dt} = \left( \frac{\partial f_1}{\partial x_2} \right)_{x^*} (x_2 - 0) + \left( \frac{\partial f_1}{\partial y_1} \right)_{y^*} (y_1 - 0) = x_2 + y_1, \quad (3)$$

$$\frac{d\Delta x_2}{dt} = \left( \frac{\partial f_2}{\partial x_1} \right)_{x^*} (x_1 - 1) + \left( \frac{\partial f_2}{\partial y_2} \right)_{y^*} (y_2 - 1) = -x_1 + y_2, \quad (4)$$

$$\begin{aligned} \frac{d\Delta y_1}{dt} &= \left( \frac{\partial g_1}{\partial x_1} \right)_{x^*} (x_1 - 1) + \left( \frac{\partial g_1}{\partial y_2} \right)_{y^*} (y_2 - 1) + \left( \frac{\partial g_1}{\partial u_1} \right)_{u_1=0} (u_1 - 0) = \\ &= \left( 2 - \frac{3}{\|x\|^3} + \frac{9x_1}{\|x\|^5} \right)_{x^*} (x_1 - 1) + (y_2 - 1) + u_1 = 8(x_1 - 1) + (y_2 - 1) + u_1, \end{aligned} \quad (5)$$

$$\begin{aligned} \frac{d\Delta y_2}{dt} &= \left( \frac{\partial g_2}{\partial x_2} \right)_{x^*} (x_2 - 0) + \left( \frac{\partial g_2}{\partial y_1} \right)_{y^*} (y_1 - 0) + \left( \frac{\partial g_2}{\partial u_2} \right)_{u_2=0} (u_2 - 0) = -4x_2 - \\ &- y_1 + u_2, \end{aligned} \quad (6)$$

Система (3)-(6) равносильна следующей системе:

$$\dot{\mathbf{z}} = A\mathbf{z} + \mathbf{u}, \quad (7)$$

$$\mathbf{z} = \begin{pmatrix} x_1 - 1 \\ x_2 \\ y_1 \\ y_2 - 1 \end{pmatrix}, A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ -1 & 0 & 0 & 1 \\ 8 & 0 & 0 & 1 \\ 0 & -4 & -1 & 0 \end{pmatrix}, \mathbf{u} = \begin{pmatrix} 0 \\ 0 \\ u_1 \\ u_2 \end{pmatrix}.$$

Исследуем неуправляемую систему, где  $u_1 = u_2 = 0$ . Находим собственные значения матрицы  $A$ :

$$|A - \lambda E| = 0, \quad (8)$$

$$\lambda^4 - 2\lambda^2 - 27 = 0, \quad (9)$$

$$\lambda_{1,2} = \pm \sqrt{1 + 2\sqrt{7}}; \lambda_{3,4} = \pm i\sqrt{2\sqrt{7} - 1}. \quad (10)$$

Так как  $\lambda_1 > 0$ , следовательно точка  $L_1$  - неустойчивая. Неустойчивость

положения точки  $L_1$  означает, что КА, находящийся в этой точке со временем непременно удалится из этой точки.

Общее решение однородной неуправляемой системы уравнений (7) имеет такой вид:

$$\mathbf{z} = \sum_{i=1}^4 c_i \mathbf{b}^i e^{\lambda_i t}, \quad (11)$$

где  $\mathbf{b}^i = (y_{1i}, y_{2i}, y_{3i}, y_{4i})^T$  - собственный вектор-столбец матрицы  $A$ . Этот вектор определяется из системы:

$$A\mathbf{b}^i = \lambda_i \mathbf{b}^i, i = \overline{1,4}, \quad (12)$$

при  $i = 1$  можно взять вектор с компонентами:

$$\mathbf{b}^1 = \left( \frac{2}{\lambda_1 - 7}, \frac{9 - \lambda_1^2}{(\lambda_1^2 - 7)\lambda_1}, \frac{\lambda_1^2 + 9}{(\lambda_1^2 - 7)\lambda_1}, -1 \right)^T. \quad (13)$$

Исследуем вопрос поведения решения системы.

При  $c_1 = 0$  решение системы (11) имеет такой вид:

$$\mathbf{z} = c_3 \mathbf{b}^3 e^{\lambda_3 t} + c_4 \mathbf{b}^4 e^{\lambda_4 t}, \quad (14)$$

что указывает на колебательное движение в окрестности точки либрации  $L_1$  при  $t \rightarrow \infty$ , т.к.  $\lambda_3$  и  $\lambda_4$  – чисто мнимые числа.

При  $c_1 > 0$  и  $c_1 < 0$  решение системы (11) выглядит так:

$$\mathbf{z} = c_1 \mathbf{b}^1 e^{\lambda_1 t} + c_3 \mathbf{b}^3 e^{\lambda_3 t} + c_4 \mathbf{b}^4 e^{\lambda_4 t}, \quad (15)$$

где первое слагаемое отвечает за удаление решения от точки либрации  $L_1$  при  $t \rightarrow \infty$ .

## 1.2. Функция опасности

Функция опасности [4], [16] служит для определения удаленности КА от точки  $L_1$ . Функция опасности – специальная функция фазовых переменных, которая имеет вид:

$$d_1 \triangleq \mathbf{b}_1 \mathbf{z}, \quad (16)$$

где  $\mathbf{b}_i = (x_{i1}, x_{i2}, x_{i3}, x_{i4})$  – собственный вектор-строка матрицы  $A$ , соответствующий собственному значению  $\lambda_i$  системы (1). Его можно найти из следующей системы:

$$\mathbf{b}_i A = \lambda_i \mathbf{b}_i \quad i = \overline{1,4}, \quad (17)$$

при  $i = 1$ . В качестве такого вектора можно взять вектор с компонентами:

$$\mathbf{b}_1 = \left( \lambda_1^2 + 5, \frac{\lambda_1^2 - 3}{\lambda_1}, \frac{\lambda_1^2 + 3}{\lambda_1}, 2 \right). \quad (18)$$

Исследуем вопрос поведения функции опасности на решениях (11) линейной неуправляемой системы (7). Для этого покажем в каком отношении находятся между собой собственные вектора-столбцы  $\mathbf{b}^i$  и вектора-строки  $\mathbf{b}_i$ . Из (12) и (17) можем определить следующие матричные равенства:

$$A \begin{pmatrix} y_{11} y_{21} y_{31} y_{41} \\ y_{12} y_{22} y_{32} y_{42} \\ y_{13} y_{23} y_{33} y_{43} \\ y_{14} y_{24} y_{34} y_{44} \end{pmatrix} = \begin{pmatrix} y_{11} y_{21} y_{31} y_{41} \\ y_{12} y_{22} y_{32} y_{42} \\ y_{13} y_{23} y_{33} y_{43} \\ y_{14} y_{24} y_{34} y_{44} \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 \\ 0 & 0 & 0 & \lambda_4 \end{pmatrix} \Leftrightarrow AY = YL, \quad (19)$$

$$\begin{pmatrix} x_{11} x_{12} x_{13} x_{14} \\ x_{21} x_{22} x_{23} x_{24} \\ x_{31} x_{32} x_{33} x_{34} \\ x_{41} x_{42} x_{43} x_{44} \end{pmatrix} A = \begin{pmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 \\ 0 & 0 & 0 & \lambda_4 \end{pmatrix} \begin{pmatrix} x_{11} x_{12} x_{13} x_{14} \\ x_{21} x_{22} x_{23} x_{24} \\ x_{31} x_{32} x_{33} x_{34} \\ x_{41} x_{42} x_{43} x_{44} \end{pmatrix} \Leftrightarrow XA = LX, \quad (20)$$

где строки матрицы  $X$  – координаты собственных векторов-строк, а столбцы матрицы  $Y$  – координаты собственных векторов-столбцов. Ясно, что матрицы  $X$  и  $Y$  невырожденные.

Если (19) умножить слева на матрицу  $X$ , а (20) умножить справа на матрицу  $Y$ , то получаем соответственно:

$$XAY = XYL, \quad (21)$$

$$XAY = LXY. \quad (22)$$

Из (21), (22) следует, что матрица  $XY$  – диагональная. Учитывая невырожденность матриц  $X$  и  $Y$  из компонент матрицы  $XY$ , следует, что:

$$\begin{cases} \mathbf{b}_i \mathbf{b}^j = 0, & i \neq j, \quad i, j = \overline{1,4}, \\ \mathbf{b}_i \mathbf{b}^j \neq 0, & i = j = \overline{1,4}. \end{cases} \quad (23)$$

При условии (23) для значения функции опасности на решении линейной системы (11) получаем:

$$d_1 = \mathbf{b}_1 \sum_{i=1}^4 c_i \mathbf{b}^i e^{\lambda_i t} = \tilde{c}_1 e^{\lambda_1 t}, \quad (24)$$

где  $\tilde{c}_1 = c_1 \mathbf{b}_1 \mathbf{b}^1$  – некоторый вещественный коэффициент. Равенство (24) описывает поведение функции  $d_1$  на решениях линейной системы (11), т. е. если решение (11) неограниченно удаляется из окрестности точки либрации  $L_1$ , то  $d_1 \rightarrow \infty$  при  $t \rightarrow \infty$ , с другой стороны, если решение (11) ограничено в окрестности точки либрации  $L_1$  ( $c_1 = 0$ ), то  $d_1 = 0$ .

Далее, из вида первой компоненты решения (11):

$$c_1 \begin{pmatrix} y_{11} \\ y_{21} \\ y_{31} \\ y_{41} \end{pmatrix} e^{\lambda_1 t} \quad (25)$$

следует, что в линейном приближении направление ухода из окрестности коллинеарной точки либрации  $L_1$  определяется векторами:

$$\begin{pmatrix} y_{11} \\ y_{21} \end{pmatrix} \text{ и } \begin{pmatrix} -y_{11} \\ -y_{21} \end{pmatrix},$$

следовательно, направление удаления КА из окрестности определяется знаком коэффициента  $c_1$ , который можно найти по начальным данным. Исследуем вопрос: как связаны направление ухода КА и знак функции опасности?

Примем, что  $\mathbf{b}_1 \mathbf{b}^1 > 0$ , а направление ухода КА к Солнцу, определяется  $c_1 > 0$ , к Земле –  $c_1 < 0$ . В качестве таких векторов можно взять вектор-столбец  $\mathbf{b}^1$  с компонентами (13), вектор-строка  $\mathbf{b}_1$  с компонентами (18). Тогда из (24) следует, что с точностью до линейного приближения  $d_1 > 0$ , если КА уходит к Солнцу, а при  $d_1 < 0$  уходит к Земле.

На рис. 3 показаны возможные направления ухода (влияние неустойчивости).

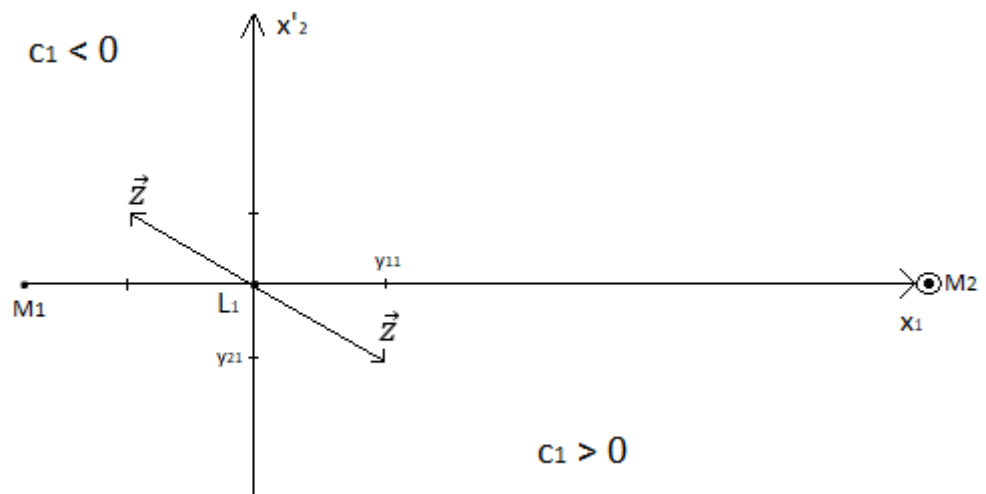


Рис. 3. Направление ухода решения

### 1.3. Построение закона управления

Перейдем к рассмотрению управляемой системы (7). Для производной функции опасности в силу системы (7) справедливо:

$$\dot{d}_1 = \lambda_1 d_1 + x_{13} u_1 + x_{14} u_2. \quad (26)$$

Откуда для удержания КА получаем задачу нахождения управляющих воздействий, для которых решение уравнения (26)  $d_1 \rightarrow 0$  при  $t \rightarrow \infty$ .

В качестве примера можно взять управления такого вида:

$$\begin{cases} u_1 = k_1 d_1, \\ u_2 = k_2 d_1, \end{cases} \quad k_2 < \frac{-\lambda_1 - x_{13} k_1}{x_{14}}, \quad (27)$$

где  $k_1$  и  $k_2$  – вещественные константы.

Реализован алгоритм решения задачи Коши на базе построенной схемы 2-х-этапного явного метода Рунге-Кутты 2-го порядка с автоматическим выбором шага, с заданной максимально допустимой локальной погрешностью ( $\varepsilon = 10^{-5}$ ). Численное моделирование реализовано на языке Java.

На рис. 4 показан график движения КА на промежутке времени  $t = [0,99]$ , где начальные данные  $x_0 = (0,99, 0)$  и  $y_0 = (0, 1)$ . А также используются управления (27), с коэффициентами  $k_1 = -0,01$  и  $k_2 = -1,5$ .

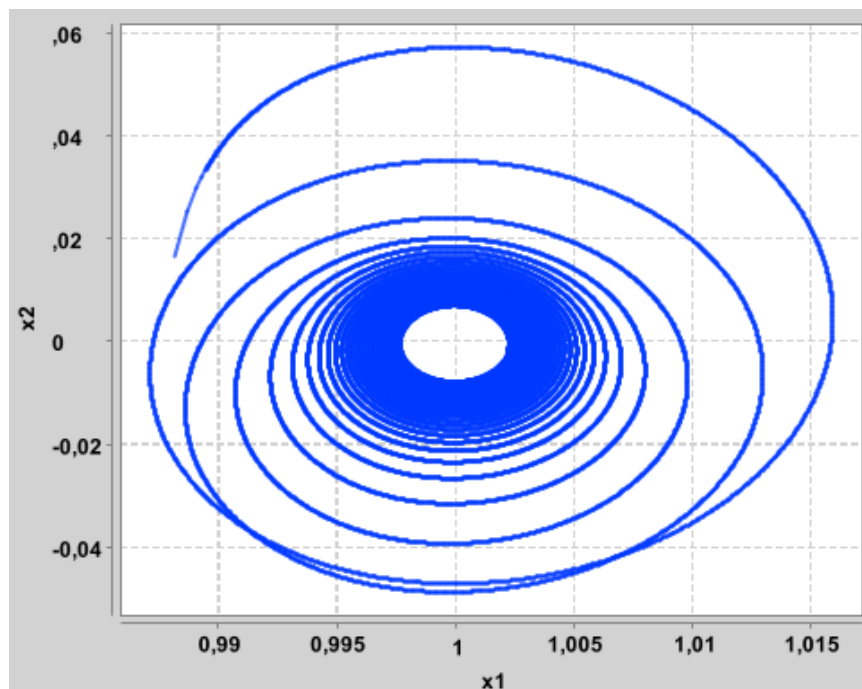


Рис. 4. Движение КА

На рис. 5 показано, как себя ведет функция опасности  $d_1$ .

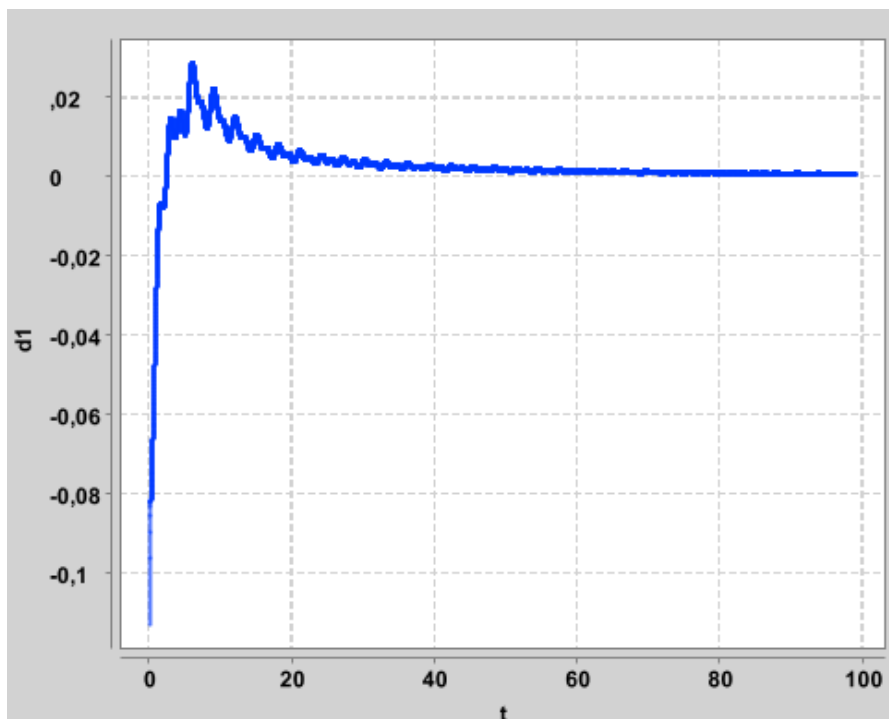


Рис. 5. Функция опасности

И управления  $u_1$  и  $u_2$  на рис. 6 и рис. 7 соответственно.

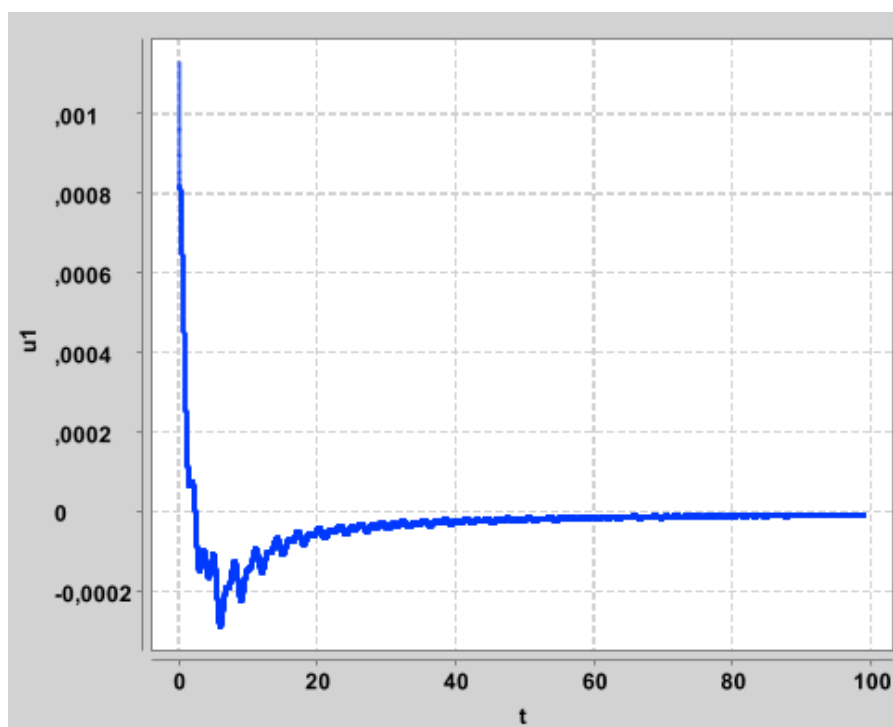


Рис. 6. Управление  $u_1$



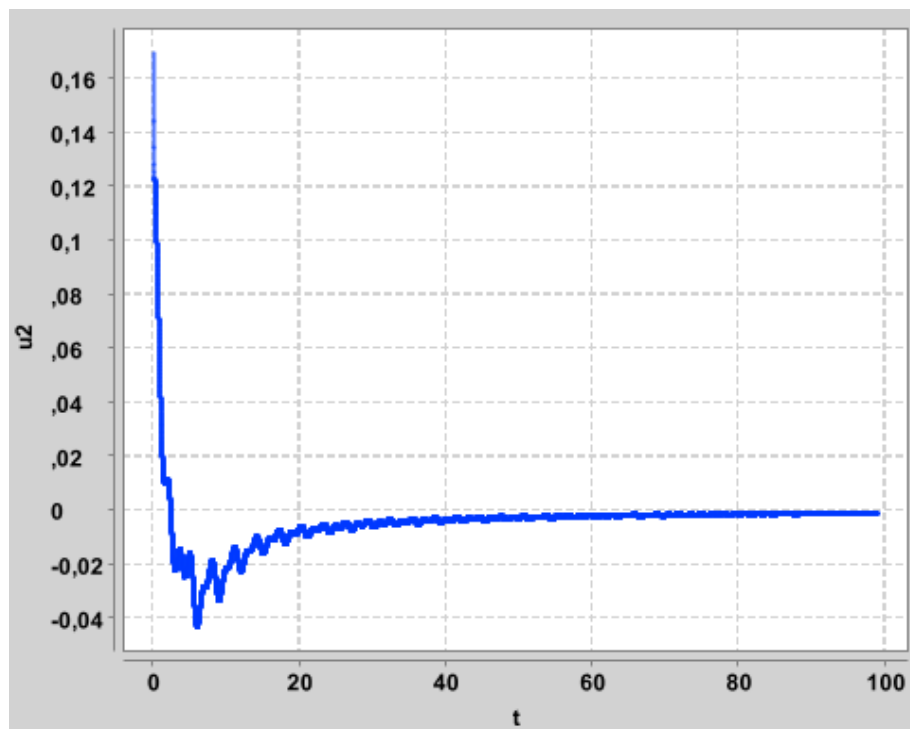


Рис. 7. Управление  $u_2$

# Глава 2. Солнечный парус

## 2.1. Солнечный парус

Солнечный парус [6]-[8], [10] с массой  $m$  представляет собой зеркальную поверхность с площадью  $S$ , которая обладает отражательными свойствами. На солнечный парус действует давление солнечного света  $P = 4.56 \cdot 10^{-6} \text{ Н/м}^2$ . Накапливающиеся фотоны дважды обмениваются импульсом с солнечным парусом: при поглощении и при излучении обратно. Парус передает это силовое воздействие КА. Если солнечный парус расположить под углом  $\alpha \in [-\frac{\pi}{2}, \frac{\pi}{2}]$  к потоку солнечных лучей  $-\vec{i}_1$ , тогда импульс, который передается солнечному парусу со стороны фотонов, будет направлен перпендикулярно светоотражающей поверхности. Направление вектора тяги будет противоположно направлению нормали  $\vec{n}$  к солнечному парусу. При повороте солнечного паруса, появляется возможность управлять направлением вектора тяги. При  $\alpha = -\frac{\pi}{2}$  или  $\alpha = \frac{\pi}{2}$  нормаль  $\vec{n}$  к солнечному парусу расположена перпендикулярно к потоку солнечных лучей, следовательно нет никакой тяги. Меняя направление нормали, можно изменять расположение солнечного паруса в пространстве. На рис. 8 обозначены силы, действующие на солнечный парус.

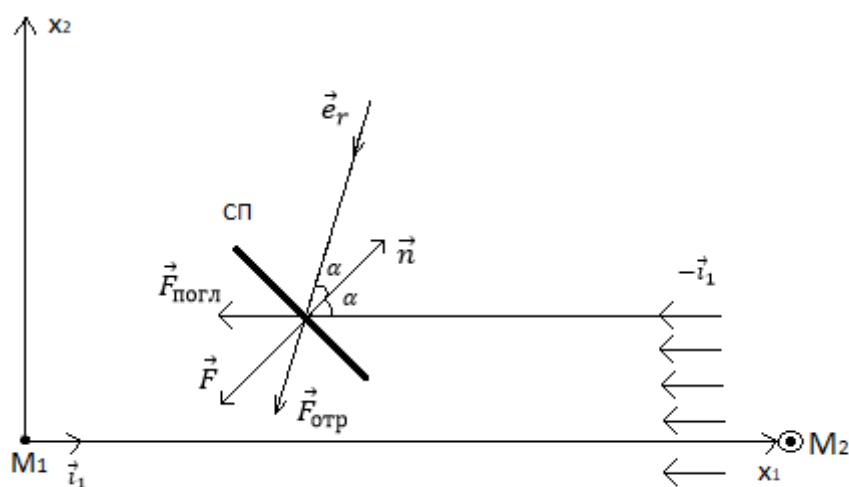


Рис. 8. Механика КА с солнечным парусом

На солнечный парус действует поглощающая сила, сонаправленная с потоком солнечных лучей ( $\mathbf{F}_{\text{погл}} \uparrow \uparrow -\mathbf{i}_1$ ):

$$\mathbf{F}_{\text{погл}} = -\frac{PS}{m} \langle -\mathbf{i}_1, \mathbf{n} \rangle \mathbf{i}_1. \quad (28)$$

В силу отражающих свойств паруса, на него действует отражающаяся сила, направленная под углом  $\alpha$  к нормали  $\mathbf{n}$ :

$$\mathbf{F}_{\text{отр}} = \frac{PS}{m} \langle -\mathbf{i}_1, \mathbf{n} \rangle \mathbf{e}_r. \quad (29)$$

Сила тяги, действующая на солнечный парус, которая находится суммарно из (27) и (28):

$$\mathbf{F} = -\frac{2PS}{m} \langle -\mathbf{i}_1, \mathbf{n} \rangle^2 \mathbf{n}. \quad (30)$$

Векторы имеют соответствующие координаты:

$$\mathbf{i}_1 = (1,0), \quad \mathbf{i}_2 = (0,1), \quad \mathbf{n} = (\cos\alpha, \sin\alpha). \quad (31)$$

(30) – управляющие силы, которые имеют вид (спроецированы на  $M_{1x1}$  и  $M_{1x2}$  соответственно):

$$\begin{cases} u_1 = -\frac{2PS}{am} \cos^3\alpha, \\ u_2 = -\frac{2PS}{am} \cos^2\alpha \sin\alpha, \end{cases} \quad (32)$$

где  $\langle -\mathbf{i}_1, \mathbf{n} \rangle^2 = \cos^2\alpha$  и  $a$  – единица ускорения в принятой системе единиц.

## 2.2. Управление солнечным парусом

Для управляемой системы (7) имеем управления вида (31). Подставляя их в производную для функции опасности (25) получаем следующее:

$$\dot{d}_1 = \lambda_1 d_1 - f(\alpha), \quad (33)$$

где  $f(\alpha) = (x_{13} \frac{2PS}{am} \cos^3 \alpha + x_{14} \frac{2PS}{am} \cos^2 \alpha \sin \alpha)$ .

Рассматриваем случай, когда  $d_1 > 0$ , при котором КА с солнечным парусом уходит к Солнцу.

Для уменьшения функции опасности, необходимо наложить ограничение на производную функции опасности, а именно:

$$\dot{d}_1 < 0. \quad (34)$$

Определим в каком промежутке лежит угол  $\alpha$ , который обеспечит удержание КА с солнечным парусом. Определим его из условия:

$$\lambda_1 d_1^0 - f(\alpha) < 0, \quad (35)$$

где  $d_1^0 = d_1^0(x_{10}, x_{20}, y_{10}, y_{20})$ , а  $(x_{10}, x_{20}, y_{10}, y_{20})$  начальное положение паруса.

Ясно, что одна из границ этого промежутка может быть определена из соотношения:

$$f(\underline{\alpha}) = \max_{\alpha \in [-\frac{\pi}{2}, \frac{\pi}{2}]} f(\alpha). \quad (36)$$

Следует заметить, что если неравенство (35) не выполняется для  $\alpha = \underline{\alpha}$ , то оно не выполняется и для любого другого  $\alpha$ , то есть КА с солнечным парусом уходит из окрестности коллинеарной точки либрации  $L_1$ , что обусловлено как начальным положением паруса, так и его техническими характеристиками –  $S$  и  $m$ .

С другой стороны, для определения второй границы  $\bar{\alpha}$  будем рассматривать область  $\alpha > \underline{\alpha}$ , для которого можно определить условие:

$$\lambda_1 d_1^0 - f(\bar{\alpha}) = 0. \quad (37)$$

Таким образом, из условий (36)-(37) получаем, что угол  $\alpha$ , который

обеспечивает удержание КА с солнечным парусом с точностью до линейного приближения лежит в отрезке  $[\underline{\alpha}, \bar{\alpha}]$ .

На рис. 9 изображено возможная ориентация солнечного паруса.

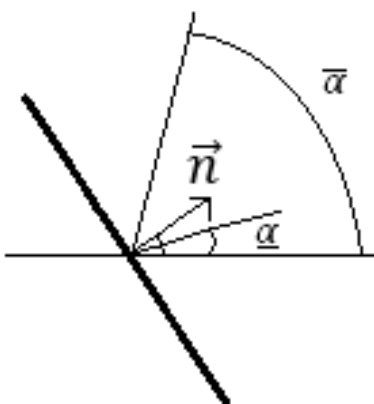


Рис. 9. Возможное положение солнечного паруса

## 2.3. Численное моделирование движения КА с солнечным парусом

Для примера возьмем следующие параметры для солнечного паруса:  $S = 400 \text{ м}^2$  и  $m = 300 \text{ кг}$ . И начальные данные  $x_0 = (1.01, 0)$  и  $y_0 = (0, 1)$ , где функция опасности в начальный момент  $d_1^0 = 0.113$ .

На рис. 10 рассмотрен случай, когда в уравнениях движения (1) управления отсутствуют или в случае управляемой системы  $\alpha = \frac{\pi}{2}$ . Траектория движения в этом случае уходит к Солнцу.

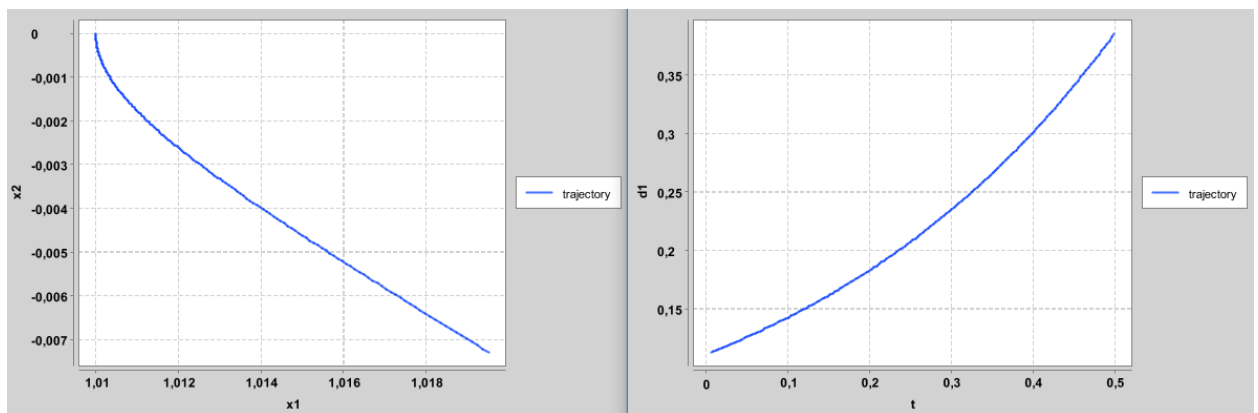


Рис. 10. Траектория движения КА и график функции опасности

На рис. 11 построен график функции  $f(\alpha)$  при  $\alpha = \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ .

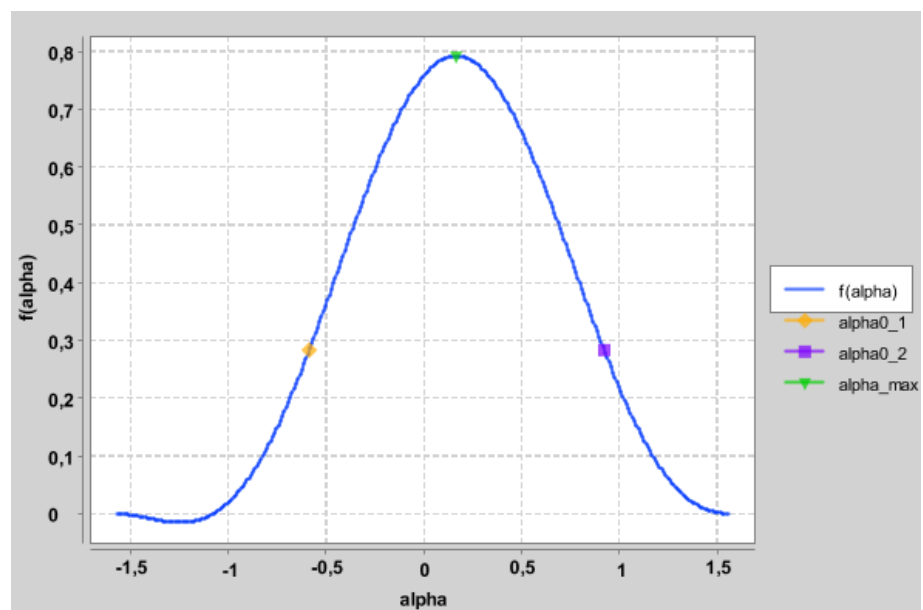


Рис. 11. График функции  $f(\alpha)$

Из графика видно, что в соответствии с представленными теоретическими результатами угол  $\alpha \in [0.168, 0.925]$ .

На рис. 12-16 построены траектории движения и соответствующие функции опасности при различных углах  $\alpha \in [0.168, \frac{\pi}{2}]$ .

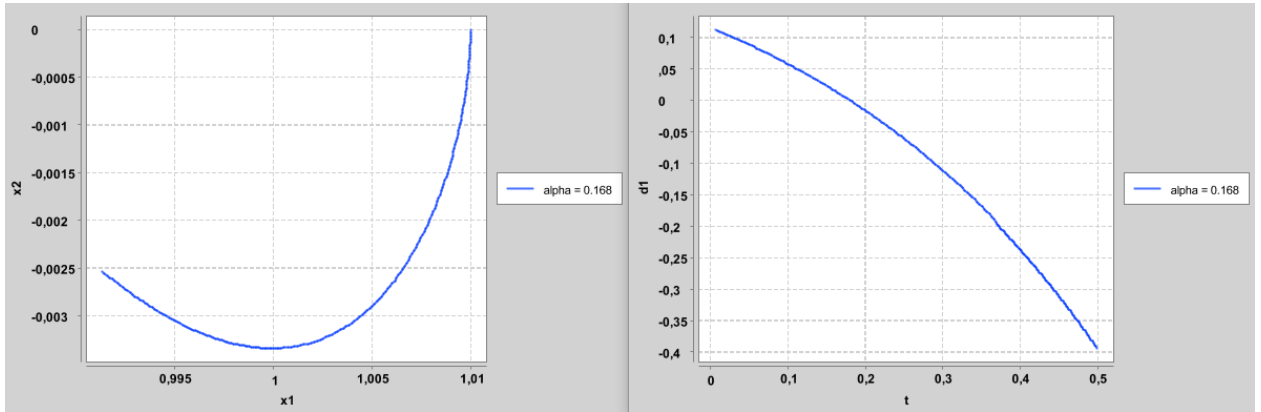


Рис. 12. Траектория движения КА и график функции опасности при  $\alpha = 0.168$   
 Функция опасности  $d_1(t) > 0$  при  $t \in [0, 0.173]$ .

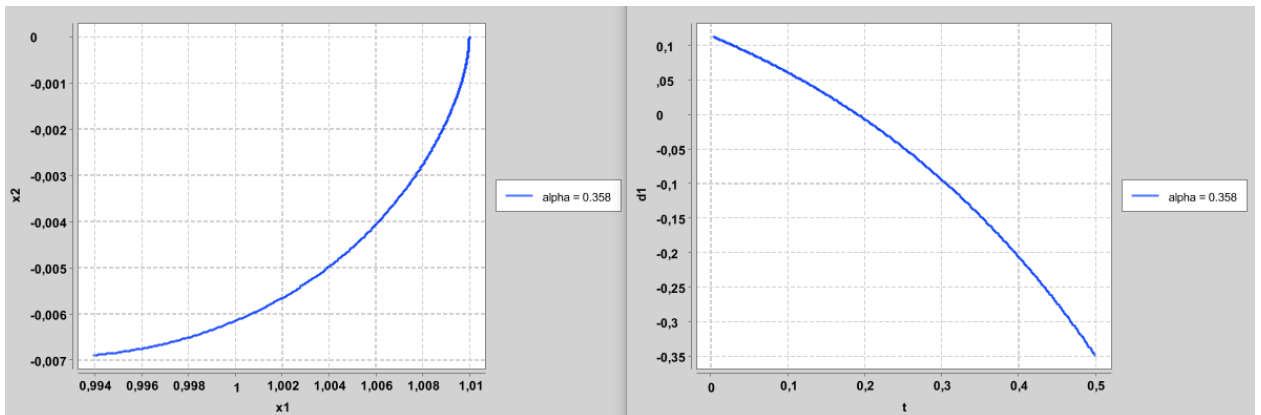


Рис. 13. Траектория движения КА и график функции опасности при  $\alpha = 0.358$   
 Функция опасности  $d_1(t) > 0$  при  $t \in [0, 0.189]$ .

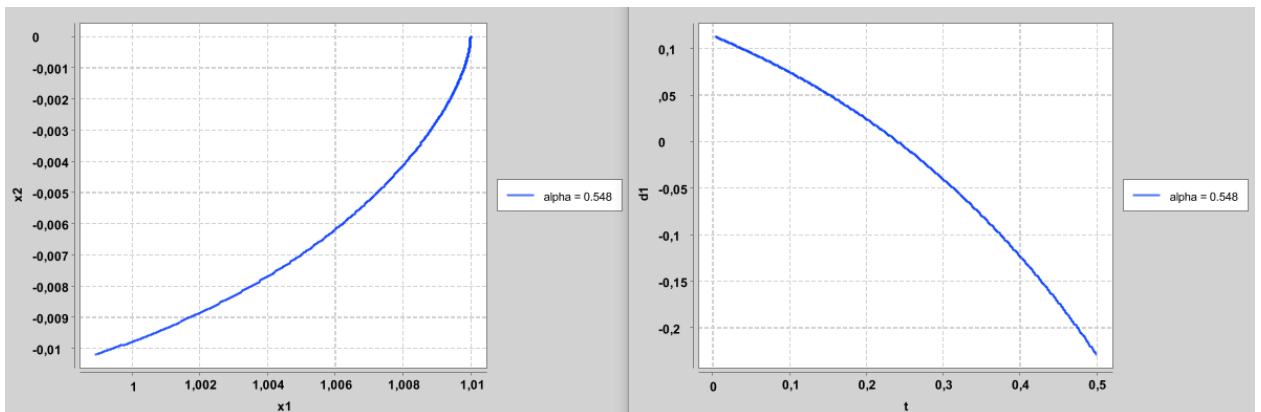


Рис. 14. Траектория движения КА и график функции опасности при  $\alpha = 0.548$

Функция опасности  $d_1(t) > 0$  при  $t \in [0, 0.238]$ .

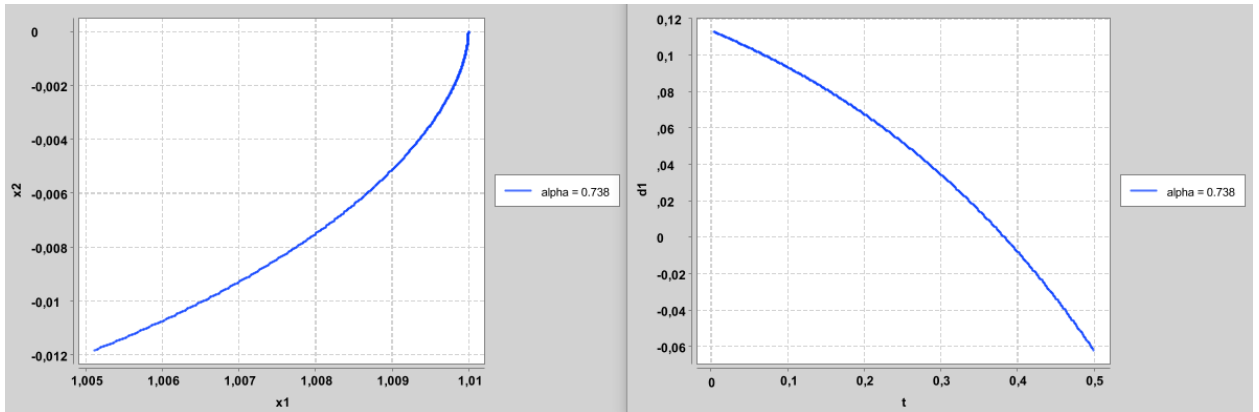


Рис. 15. Траектория движения КА и график функции опасности при  $\alpha = 0.738$

Функция опасности  $d_1(t) > 0$  при  $t \in [0, 0.381]$ .

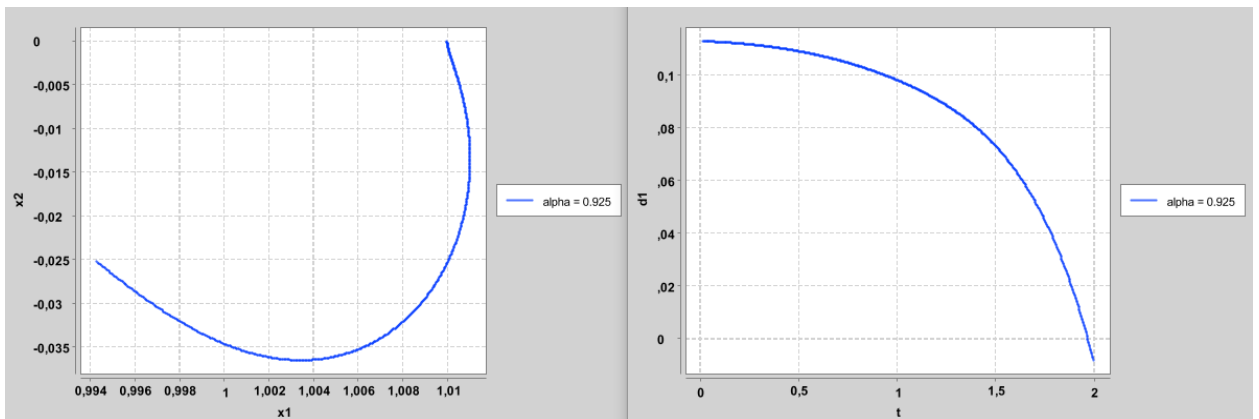


Рис. 16. Траектория движения КА и график функции опасности при  $\alpha = 0.925$

Функция опасности  $d_1(t) > 0$  при  $t \in [0, 1.954]$ .

На рис. 17, показано, что при  $\alpha > \bar{\alpha}$ , траектория движения уходит к Солнцу.

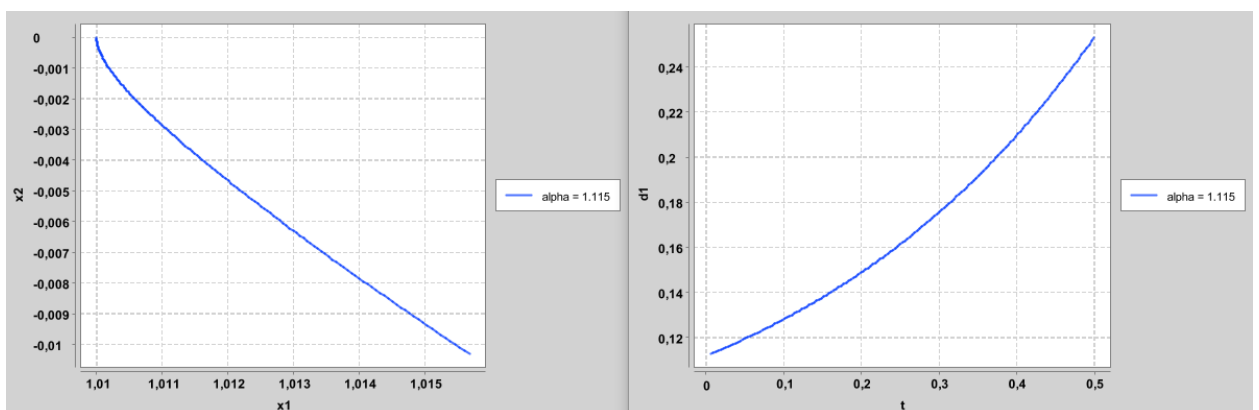


Рис. 17. Траектория движения КА и график функции опасности при  $\alpha = 1.115$

Таким образом, из вида рис. 12-16 можно сказать, что  $t \in [t_0, t_1]$  – время



удержания КА с солнечным парусом в окрестности точки либрации  $L_1$ , при котором  $d_1(t) > 0$ . После того, как функция опасности обращается в ноль в момент времени  $t_1$ , КА начинает ускоряться, то есть КА начинает уходить в сторону Земли с ускорением, которое ему сообщает парус. Однако, если до момента  $d_1(t_1) = 0$  удастся сориентировать парус так, что  $\alpha = \frac{\pi}{2}$ , то КА снова начнет уходить к Солнцу. Также из вида графиков рис. 12-16 можно видеть зависимость, что при увеличении  $\alpha \in [0.168, 0.925]$ , увеличивается время удержания КА, что объясняется величиной сообщаемого КА ускорения парусом. Также следует заметить (см. рис. 11), что для промежутка  $\alpha \in \left[-\frac{\pi}{2}, 0.168\right]$  будут иметь место аналогичные рассуждения.

## Выводы

Для анализа поведения решений уравнений движений КА были рассмотрены линеаризованные уравнения плоской математической модели управляемого движения КА во вращающейся системе координат в рамках хилловского приближения круговой ограниченной задачи трех тел. На основании этой линеаризованной модели строится функция опасности, которая оказывается весьма эффективным средством в проблеме движения КА в окрестности точки либрации.

Для определения ориентации солнечного паруса, обеспечивающего удержание КА в окрестности точки либрации  $L_1$ , было показано, что угол  $\alpha$  с точностью до линейного приближения должен удовлетворять ограничению (35), иначе КА будет уходить в сторону Солнца. Определен промежуток  $[\underline{\alpha}, \bar{\alpha}]$ , который представляет собой допустимую область управления для удержания КА при заданных  $S, m$ .

Проведен численный анализ траекторий с законами управления, которые уменьшают функцию опасности и обеспечивают удержание КА с солнечным парусом в окрестности точки либрации  $L_1$ . В результате на рис. 12-17 получены траектории движения КА с солнечным парусом в окрестности  $L_1$ . По поведению функции опасности на траекториях движения с различным выбором допустимых углов  $\alpha$  можно судить о времени удержания КА с солнечным парусом в окрестности точки либрации  $L_1$ .

Листинг программы на языке Java содержится в приложении, в котором содержится реализация алгоритма решения задачи Коши на базе построенной схемы 2-х-этапного явного метода Рунге-Кутты 2-го порядка с автоматическим выбором шага, с заданной максимально допустимой локальной погрешностью ( $\epsilon = 10^{-5}$ ).

## Заключение

В результате проведенного исследования:

1. Показано куда распространяются неустойчивые траектории движения в окрестности  $L_1$ . Возможные направления ухода КА.
2. Показана связь неустойчивых траекторий в окрестности точки либрации со знаком функции опасности.
3. Представлены законы управления движением КА с солнечным парусом в окрестности точки либрации  $L_1$ .
4. Дана оценка для угла  $\alpha$  наклона нормали паруса в виде отрезка значений, выбор которых с точностью до линейного приближения обеспечивает удержание КА в окрестности точки либрации.
5. Проведено численное моделирование возможных траекторий движения КА с солнечным парусом при разных допустимых углах. Найдены промежутки времени удержания КА с солнечным парусом.

## Список литературы

1. Охоцимский Д. Е., Сихарулидзе Ю. Г. Основы механики космического полета: Учеб. пособие. - М.: Наука, 1990. 448 с.
2. Маркеев А. П. Точки либрации в небесной механике и космодинамике. М.: Наука, 1978. 312 с.
3. Дубошин Г. Н. Небесная механика. Основные задачи и методы. М.: Наука, 1968. 800 с.
4. Шиманчук Д. В. Моделирование орбитального управляемого движения космического аппарата в окрестности коллинеарной точки либрации  $L_1$  // Вестник СПбГУ. Сер. 10. Прикладная математика, информатика, процессы управления. Вып. 3, 2010. С. 86–92.
5. Шмыров В. А. Стабилизация управляемого орбитального движения космического аппарата в окрестности коллинеарной точки либрации  $L_1$  // Вестник СПбГУ. Сер. 10. Прикладная математика, информатика, процессы управления. Вып. 2, 2005. С. 193–199.
6. Поляхова Е. Н., Королев В. С. Задачи управления космическим аппаратом с солнечным парусом.
7. Королев В. С., Поляхова Е. Н., Потоцкая И. Ю. Комплексный подход к проблеме движения космического аппарата с солнечным парусом // Universum: Технические науки: электрон. науч. журн. № 10(31), 2016.
8. Королев В. С., Поляхова Е. Н. Проблемы устойчивости движения космического аппарата с солнечным парусом // Технические науки – от теории к практике: сб. ст. по матер. LXI междунар. науч.-практ. конф. №8(56). – Новосибирск: СибАК, 2016. С. 21–34.
9. Кабанов С. М., Фридлиндер Г. В. Определение пространственных стационарных точек типа POLAR-SITTER в круговой задаче трех тел с использованием солнечного паруса. // Технические науки: Вып. 5 – Известия ТулГУ, 2016. С. 116–130.

10. Поляхова Е. Н. Космический полет с солнечным парусом. М.: Наука, 1986.
11. Farres A. Catalogue on the Dynamics of a Solar Sail around  $L_1$  and  $L_2$ . 2017.
12. Ишков С. А., Старинова О. Л. Оптимизация и моделирование движения космического аппарата с солнечным парусом. Самара: Известия СНЦ РАН, 2005. С. 99-106.
13. Старинова О. Л., Горбунова И. В. Оптимизация гелиоцентрического движения космического аппарата с солнечным парусом. Самара: Изд-во СНЦ РАН, 2015. С. 168-171.
14. Кошкин. Н. И., Лопаченко В. В. NanoSail-D – первый парусник на орбите Земли. 2011.
15. Проект SOHO. <https://sohowww.nascom.nasa.gov/>
16. Шмырова А. С., Шмыров В. А. Об одном семействе критериев качества в задаче стабилизации движения в окрестности коллинеарной точки либрации. Труды МАИ. Выпуск № 84. С. 1-16.

## Приложение

```
import org.knowm.xchart.QuickChart;
import org.knowm.xchart.SwingWrapper;
import org.knowm.xchart.XYChart;

public class Main {
    static double lambda = Math.sqrt(1 + 2 * Math.sqrt(7));
    static double b01 = lambda * lambda + 5;
    static double b02 = (lambda * lambda - 3) / lambda;
    static double b03 = (lambda * lambda + 3) / lambda;
    static double b04 = 2.0;
    static double z = 2.0;
    static double b1 = 1.0 / 2 * z;
    static double b2 = 1 - b1;
    static double eps = 0.000001;
    static double s = 1.0;

    static double P = 4.56 * Math.pow(10, -6);
    static double S = 400;
    static double m = 300;
    static double a = 5.95 * Math.pow(10, -5);
    static double alpha;

    static double norma(double q, double w, double e, double r) {
        return Math.sqrt(q * q + w * w + e * e + r * r);
    }
    //Функция опасности
    static double d1(double x1, double x2, double y1, double y2) {
        return b01 * (x1 - 1) + b02 * x2 + b03 * y1 + b04 * (y2 - 1);
    }
    //Правые части уравнений
    static double f1(double x1, double x2, double y1, double y2) {
        return x2 + y1;
    }
    static double f2(double x1, double x2, double y1, double y2) {
        return -x1 + y2;
    }
    static double f3(double x1, double x2, double y1, double y2) {
        return ((-3 / (Math.pow(norma(x1, x2, 0, 0), 3))) + 2) * x1 + y2 -
(2.0 * P * S * Math.pow(Math.cos(alpha), 3)) / (a * m);
    }
    static double f4(double x1, double x2, double y1, double y2) {
        return ((-3 / (Math.pow(norma(x1, x2, 0, 0), 3))) - 1) * x2 - y1 -
(2.0 * P * S * Math.pow(Math.cos(alpha), 2) * Math.sin(alpha)) / (a * m);
    }
    static double k11(double h, double x1, double x2, double y1, double y2) {
        return h * f1(x1, x2, y1, y2);
    }
    static double k21(double h, double x1, double x2, double y1, double y2) {
        return h * f2(x1, x2, y1, y2);
    }
    static double k31(double h, double x1, double x2, double y1, double y2) {
        return h * f3(x1, x2, y1, y2);
    }
    static double k41(double h, double x1, double x2, double y1, double y2) {
        return h * f4(x1, x2, y1, y2);
    }
    static double k12(double h, double x1, double x2, double y1, double y2) {
        return h * f1(x1 + z * k11(h, x1, x2, y1, y2), x2 + z * k21(h, x1,
```

```

x2, y1, y2), y1 + z * k31(h, x1, x2, y1, y2), y2 + z * k41(h, x1, x2, y1,
y2));
}
static double k22(double h, double x1, double x2, double y1, double y2) {
return h * f2(x1 + z * k11(h, x1, x2, y1, y2), x2 + z * k21(h, x1,
x2, y1, y2), y1 + z * k31(h, x1, x2, y1, y2), y2 + z * k41(h, x1, x2, y1,
y2));
}
static double k32(double h, double x1, double x2, double y1, double y2) {
return h * f3(x1 + z * k11(h, x1, x2, y1, y2), x2 + z * k21(h, x1,
x2, y1, y2), y1 + z * k31(h, x1, x2, y1, y2), y2 + z * k41(h, x1, x2, y1,
y2));
}
static double k42(double h, double x1, double x2, double y1, double y2) {
return h * f4(x1 + z * k11(h, x1, x2, y1, y2), x2 + z * k21(h, x1,
x2, y1, y2), y1 + z * k31(h, x1, x2, y1, y2), y2 + z * k41(h, x1, x2, y1,
y2));
}

static double f(double alpha) {
return (2.0 * P * S * (b03 * (Math.pow(Math.cos(alpha), 3)) + b04 *
Math.pow(Math.cos(alpha), 2) * Math.sin(alpha)) / (a * m));
}

//Определение границ промежутка, в котором лежит альфа
public static void Solution_1() {
double alpha1[] = new double[500];
double f[] = new double[500];
int k = 0;
double q = Math.PI / 2.0;
for (double j = -q; j < q; j = j + 0.01) {
alpha1[k] = j;
k++;
}
k = 0;
for (double j = -q; j < q; j = j + 0.01) {
f[k] = f(j);
k++;
}
//Поиск максимального значения альфа
double f_max = -10;
double alpha_max = -2;
for (double j = -q; j < q; j = j + 0.00001) {
if (f(j) > f_max) {
f_max = f(j);
alpha_max = j;
}
}
System.out.println("f_max = " + f_max);
System.out.println("alpha_max = " + alpha_max);
//Поиск минимального значения альфа
for (double alpha_ = -q; alpha_ < q; alpha_ = alpha_ + 0.000001) {
alpha = alpha_;

double pro_d1 = lambda * d1(1.01, 0, 0, 1) - f(alpha_);
if ((Math.abs(pro_d1) < 0.000001)) {
System.out.println("alpha0 = " + alpha_);
}
}
}
//График f(alpha) =====
double a[] = new double[k - 1];

```

```

    double b[] = new double[k - 1];

    for (int j = 0; j < k - 1; j++) {
        a[j] = alpha1[j];
    }
    for (int j = 0; j < k - 1; j++) {
        b[j] = f[j];
    }
    XYChart chart = QuickChart.getChart(" ", "alpha", "f(alpha)",
    "f(alpha)", a, b);
    new SwingWrapper(chart).displayChart();

    System.out.println("f(alpha0_1) = " + (f(-0.583)));
    System.out.println("f(alpha0_2) = " + (f(0.925)));

    chart.addSeries("alpha0_1", new double[]{-0.583}, new
double[] {0.28});
    chart.addSeries("alpha0_2", new double[] {0.925}, new double[] {0.28});
    chart.addSeries("alpha_max", new double[] {0.168}, new
double[] {0.792});
}
//Метод Рунге-Кутты
public static void Solution_2() {
    //Выбор начального шага
    double w = Math.pow((1.0 / (1)), s + 1) + Math.pow(Math.abs(f1(0, 0,
0, 0)), s + 1);
    double h = Math.pow(eps / w, 1.0 / (s + 1));
    int n = (int) (Math.floor(2 * 1) / h);
    double[] y1 = new double[14000 * n];
    double[] y2 = new double[14000 * n];
    double[] y3 = new double[14000 * n];
    double[] y4 = new double[14000 * n];
    double[] p1 = new double[14000 * n];
    double[] p2 = new double[14000 * n];
    double[] p3 = new double[14000 * n];
    double[] p4 = new double[14000 * n];
    double[] t_graf1 = new double[14000 * n];
    y1[0] = 1.01;
    y2[0] = 0;
    y3[0] = 0;
    y4[0] = 1;
    for (double alpha_ = 0.168; alpha_ <= 1.2; alpha_ = alpha_ + 0.19) {
        alpha = alpha_;
        int i = 1, k = 1;
        for (double t = h; t <= 0.5; t = t + h) {
            t_graf1[k - 1] = t;
            double H = h;
            double y_1, y_2, y_3, y_4, y1a, y2a, y3a, y4a;
            y_1 = y1[i - 1] + (b2 * k11(H / 2.0, y1[i - 1], y2[i - 1],
y3[i - 1], y4[i - 1])) + b1 * k12(H / 2.0, y1[i - 1], y2[i - 1], y3[i - 1],
y4[i - 1]));
            y_2 = y2[i - 1] + (b2 * k21(H / 2.0, y1[i - 1], y2[i - 1],
y3[i - 1], y4[i - 1])) + b1 * k22(H / 2.0, y1[i - 1], y2[i - 1], y3[i - 1],
y4[i - 1]));
            y_3 = y3[i - 1] + (b2 * k31(H / 2.0, y1[i - 1], y2[i - 1],
y3[i - 1], y4[i - 1])) + b1 * k32(H / 2.0, y1[i - 1], y2[i - 1], y3[i - 1],
y4[i - 1]));
            y_4 = y4[i - 1] + (b2 * k41(H / 2.0, y1[i - 1], y2[i - 1],
y3[i - 1], y4[i - 1])) + b1 * k42(H / 2.0, y1[i - 1], y2[i - 1], y3[i - 1],
y4[i - 1]));

```



```

        y1a = y_1 + (b2 * k11(H / 2.0, y_1, y_2, y_3, y_4) + b1 *
k12(H / 2.0, y_1, y_2, y_3, y_4));
        y2a = y_2 + (b2 * k21(H / 2.0, y_1, y_2, y_3, y_4) + b1 *
k22(H / 2.0, y_1, y_2, y_3, y_4));
        y3a = y_3 + (b2 * k31(H / 2.0, y_1, y_2, y_3, y_4) + b1 *
k32(H / 2.0, y_1, y_2, y_3, y_4));
        y4a = y_4 + (b2 * k41(H / 2.0, y_1, y_2, y_3, y_4) + b1 *
k42(H / 2.0, y_1, y_2, y_3, y_4));

        y1[i] = y1[i - 1] + (b2 * k11(H, y1[i - 1], y2[i - 1], y3[i -
1], y4[i - 1]) + b1 * k12(H, y1[i - 1], y2[i - 1], y3[i - 1], y4[i - 1]));
        y2[i] = y2[i - 1] + (b2 * k21(H, y1[i - 1], y2[i - 1], y3[i -
1], y4[i - 1]) + b1 * k22(H, y1[i - 1], y2[i - 1], y3[i - 1], y4[i - 1]));
        y3[i] = y3[i - 1] + (b2 * k31(H, y1[i - 1], y2[i - 1], y3[i -
1], y4[i - 1]) + b1 * k32(H, y1[i - 1], y2[i - 1], y3[i - 1], y4[i - 1]));
        y4[i] = y4[i - 1] + (b2 * k41(H, y1[i - 1], y2[i - 1], y3[i -
1], y4[i - 1]) + b1 * k42(H, y1[i - 1], y2[i - 1], y3[i - 1], y4[i - 1]));
        //Оценка локальной погрешности
        p1[i] = (y1a - y1[i]) / (3.0);
        p2[i] = (y2a - y2[i]) / (3.0);
        p3[i] = (y3a - y3[i]) / (3.0);
        p4[i] = (y4a - y4[i]) / (3.0);
        //Автоматический выбор шага интегрирования
        if (norma(p1[i], p2[i], p3[i], p4[i]) > eps * 4.0) {
            t = t - H / 2.0;
            y1[i] = y1[i - 1] + (b2 * k11(H / 2.0, y1[i - 1], y2[i -
1], y3[i - 1], y4[i - 1]) + b1 * k12(H / 2.0, y1[i - 1], y2[i - 1], y3[i -
1], y4[i - 1]));
            y2[i] = y2[i - 1] + (b2 * k21(H / 2.0, y1[i - 1], y2[i -
1], y3[i - 1], y4[i - 1]) + b1 * k22(H / 2.0, y1[i - 1], y2[i - 1], y3[i -
1], y4[i - 1]));
            y3[i] = y3[i - 1] + (b2 * k31(H / 2.0, y1[i - 1], y2[i -
1], y3[i - 1], y4[i - 1]) + b1 * k32(H / 2.0, y1[i - 1], y2[i - 1], y3[i -
1], y4[i - 1]));
            y4[i] = y4[i - 1] + (b2 * k41(H / 2.0, y1[i - 1], y2[i -
1], y3[i - 1], y4[i - 1]) + b1 * k42(H / 2.0, y1[i - 1], y2[i - 1], y3[i -
1], y4[i - 1]));
        } else {
            if ((norma(p1[i], p2[i], p3[i], p4[i]) > eps) &&
(norma(p1[i], p2[i], p3[i], p4[i]) <= eps * 4.0)) {
                y1[i] = y1a;
                y2[i] = y2a;
                y3[i] = y3a;
                y4[i] = y4a;
                H = H / 2.0;
            } else {
                if ((norma(p1[i], p2[i], p3[i], p4[i]) >= eps / 8.0)
&& (norma(p1[i], p2[i], p3[i], p4[i]) <= eps)) {
                    y1[i] = y1[i];
                    y2[i] = y2[i];
                    y3[i] = y3[i];
                    y4[i] = y4[i];
                } else {
                    H = 2.0 * H;
                }
            }
        }
    }
    if ((t + H > 1) && (t < 1)) {
        H = -t + 1;
    }
    h = H / 2;

```

```

        y1[i] = y1a + p1[i];
        y2[i] = y2a + p2[i];
        y3[i] = y3a + p3[i];
        y4[i] = y4a + p4[i];
        k = i;
        i++;
    }
    y1[k + 1] = y1[k] + p1[k];
    y2[k + 1] = y2[k] + p2[k];
//Графики =====
    //Траектория движения КА
    double v1[] = new double[k-1];
    double v2[] = new double[k -1];
    for (int j = 0; j < k-1; j++) {
        v1[j] = y1[j];
        v2[j] = y2[j];
    }
    XYChart chart = QuickChart.getChart(" ", "x1", "x2", "alpha = " +
(alpha_), v1, v2);
    new SwingWrapper(chart).displayChart();
    //Функция опасности
    double d1[] = new double[k-1];
    double[] t_graf = new double[k-1];
    for (int j = 0; j < k-1; j++) {
        d1[j] = d1((y1[j] + p1[j]), (y2[j] + p2[j]), (y3[j] +
p3[j]), (y4[j] + p4[j]));
    }
    for(int j = 0; j < k-1; j++) {
        t_graf[j] = t_graf1[j];
    }
    XYChart chart1 = QuickChart.getChart(" ", "t", "d1", "alpha = " +
(alpha_), t_graf, d1);
    new SwingWrapper(chart1).displayChart();
    }
}
    public static void main(String[] args) {
        System.out.println("Поиск альфа: ");
        Solution_1();
        System.out.println("Траектории движения КА и графики функции
опасности: ");
        Solution_2();
    }
}

```