

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
"ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ"

Т.К. Кацаран, Л.Ю. Кабанцова

**МЕТОД МАЛОГО ПАРАМЕТРА В ЗАДАЧАХ  
ОПТИМАЛЬНОГО УПРАВЛЕНИЯ**

Учебное пособие для вузов

Воронеж  
Издательский дом ВГУ  
2016

Утверждено научно-методическим советом факультета прикладной математики, информатики и механики 08.06.2016 г., протокол № 10

Рецензент канд. техн. наук, доцент кафедры математических методов исследования операций ВГУ Г.Д. Чернышова

Учебное пособие подготовлено на кафедре нелинейных колебаний факультета прикладной математики, информатики и механики Воронежского государственного университета.

Рекомендовано для магистров факультета прикладной математики, информатики и механики Воронежского государственного университета, специализирующихся по профилю "Оптимизация и оптимальное управление".

Для направления: 01.04.02 – Прикладная математика и информатика,

# Введение

Одной из характерных особенностей современной эпохи является всевозрастающее внимание к проблемам управления. Управление - это воздействие на систему, способное изменить текущее состояние и, следовательно, всё последующее развитие системы. Всюду, где имеется возможность активного участия человека, возникает проблема отыскания наилучшего или, как говорят, оптимального из возможных управлений. Математическая теория управления была создана в середине пятидесятых годов прошлого столетия и получила название теории оптимального управления. Выдающуюся роль при этом сыграл принцип максимума Понтрягина. Этот принцип представляет собой определённого типа необходимое условие, которое даёт возможность найти все допустимые управления, которые могут претендовать на роль оптимальных.

В теории оптимального управления за время её развития произошёл синтез идей и методов, с одной стороны, восходящих к классическому вариационному исчислению, с другой – вполне современных аналитических и численно-аналитических методов. Широко используется принцип Беллмана, согласно которому для любого начального состояния и выбранного начального управления последующее оптимальное управление совпадает с исходным оптимальным управлением относительно состояния, случившегося в результате применения оптимального управления. В процессе развития и применения теории оптимального управления задачи постепенно усложнялись. Точное решение задач оптимального управления может быть построено сравнительно редко и лишь для определённых классов задач. Одним из распространенных методов решения трудных задач является сведение их к более простым.

В то же время многие прикладные задачи оптимального управления в явном или неявном виде содержат малые параметры, характеризующие относительную малость тех или иных воздействий или факторов. Поэтому могут быть использованы приближённые методы, основанные на идее малого параметра. Введение малого параметра оправдано в тех случаях, когда невозмущенная задача (малый параметр равен нулю) может быть решена численно или аналитически значительно более просто, чем возмущенная. Это имеет место, например, если система близка к линейной, или к неуправляемой, или в ней выявляются быстрые и медленные переменные, допускающие разделение движений.

В настоящем учебном пособии даётся общая постановка задачи оптимального управления динамической системой, поведение которой описывается с помощью системы обыкновенных дифференциальных уравнений. Здесь формулируется вариант принципа максимума для неавтономных систем. Предполагается, что все функции, описывающие динамическую систему (дифференциальное уравнение, начальные и краевые условия, критерий качества) разлагаются в ряды по степеням малого параметра. Это даёт возможность построить алгоритм исследования слабоуправляемых систем, с использованием которого решена задача о полёте на максимальную дальность.

Метод малого параметра также применяется при решении задачи успокоения твёрдого тела за минимальное время при наличии возмущений. Специфика задачи состоит в том, что при отсутствии возмущений находится точное решение этой задачи (оптимальное управление, минимальное время и, при использовании эллиптических функций, оптимальные фазовые траектории), что даёт возможность численного на-

хождения следующего приближения.

В создании настоящего учебного пособия принимали участие магистры Вороникова Е. и Диденко Д., Харитонов В. и другие, специализирующиеся по профилю "Оптимизация и оптимальное управление".

## § 1. Метод малого параметра в теории управления

### 1.1 Постановка задачи оптимального управления

Рассмотрим систему дифференциальных уравнений

$$\dot{x} = f(t, x, u), \quad x(t_0) = a, \quad (1.1)$$

где  $x = (x_1, x_2, \dots, x_n)$  -  $n$ -мерный вектор фазовых координат,  $u = (u_1, u_2, \dots, u_m)$  -  $m$ -мерный вектор управляющих функций,  $f = (f_1, f_2, \dots, f_n)$  - заданная  $n$ -мерная векторная функция,  $t_0$  - начальный момент времени,  $a$  - вектор начального фазового состояния.

На управляющую векторную функцию  $u(t)$  наложено ограничение

$$u(t) \in U, \quad t \geq t_0, \quad (1.2)$$

где  $U$  - заданное замкнутое ограниченное множество в  $m$ -мерном пространстве. Функции  $u_i(t)$ ,  $i = 1, \dots, m$  предполагаются кусочно-непрерывными. Граничные условия в конце процесса управления  $T \geq t_0$  заданы в виде

$$h(x(T), T) = 0, \quad q(x(T), T) = 0, \quad (1.3)$$

и минимизируемый функционал (критерий качества) имеет вид

$$J = F(x(T), T). \quad (1.4)$$

Здесь  $h(x, t)$  и  $F(x, t)$  - заданные скалярные функции,  $q(x, t) = (q_1, q_2, \dots, q_r)$  - заданная  $r$ -мерная векторная функция, причем  $0 \leq r \leq n - 1$ . Функции  $h$  и  $q$  предполагаются непрерывно дифференцируемыми по  $x, t$ , а  $F$  - дважды непрерывно дифференцируемой. Первое равенство (1.3) служит условием определения момента времени  $T$  окончания процесса. Предполагается, что функция  $h$  такова, что при любой допустимой траектории  $x(t)$  функция  $h(x, t)$  монотонно зависит от  $t$  в некотором интервале времени. Поэтому уравнение  $h(x(t), t) = 0$  определяет для каждой допустимой траектории единственный момент времени  $T \geq t_0$ .

Задача оптимального управления состоит в определении управления  $u(t)$  и соответствующей оптимальной траектории  $x(t)$ , которые при  $t_0 \leq t \leq T$  удовлетворяют системе уравнений и начальному условию (1.1), граничным условиям (1.3), ограничениям на управление (1.2) и доставляют минимум функционалу (1.4).

### 1.2 Принцип максимума Понтрягина

Традиционно принцип максимума Понтрягина, представляющий необходимое условие оптимальности, используется для нахождения оптимального управления автономных систем. Управляемая система дифференциальных уравнений (1.1) не является автономной. В связи с этим введем дополнительные фазовые переменные,

определяемые уравнениями с начальными условиями.

$$\dot{x}_0 = f_0, \quad \dot{x}_{n+1} = 1, \quad x_0(t_0) = 0, \quad x_{n+1}(t_0) = t_0, \quad (1.5)$$

где

$$f_0 = \frac{dF}{dt} = \frac{\partial F}{\partial t} + \left( \frac{\partial F}{\partial x}, f \right).$$

Здесь и далее  $\partial/\partial x$  - оператор градиента по фазовым координатам,  $dF/dt$  - полная производная функции  $F$  вдоль траекторий системы (1.1), скобки  $(\cdot, \cdot)$  означают скалярное произведение двух векторов. Очевидно, что  $x_{n+1}(t) = t$  и поэтому аргумент  $t$  у функций  $f, f_0, h, q$  и  $F$  можно заменить на  $x_{n+1}$ . Тогда система (1.1), (1.5) будет автономной, а функционал (1.4) примет вид  $J = x_0(T)$ .

Введем вектор сопряженных переменных  $\psi(t) = (\psi_1, \psi_2, \dots, \psi_n)$ , а также сопряженные переменные  $\psi_{n+1}(t)$  и  $\psi_0(t)$ , причем положим  $\psi_0 = -1$ . Функция Гамильтона  $H'$  и сопряженные уравнения для системы (1.1), (1.5) примут вид

$$\begin{aligned} H' &= (\psi, f) + \psi_{n+1} - f_0 = \left( \psi - \frac{\partial F}{\partial x}, f \right) + \psi_{n+1} - \frac{\partial F}{\partial t}, \\ \dot{\psi}_k &= -\frac{\partial H'}{\partial x_k} = -\left( \psi - \frac{\partial F}{\partial x}, \frac{\partial f}{\partial x_k} \right) + \left[ \frac{\partial^2 F}{\partial t \partial x_k} + \left( \frac{\partial}{\partial x_k} \frac{\partial F}{\partial x}, f \right) \right], \\ &k = 1, 2, \dots, n. \end{aligned} \quad (1.6)$$

С учетом краевых условий (1.3) условия трансверсальности запишутся в виде

$$\begin{aligned} \psi &= \lambda \frac{\partial h}{\partial x} + \sum_{i=1}^r \lambda_i \frac{\partial q_i}{\partial x}, \\ \psi_{n+1} &= \lambda \frac{\partial h}{\partial t} + \sum_{i=1}^r \lambda_i \frac{\partial q_i}{\partial t}, \\ H' &= 0 \text{ при } t = T. \end{aligned} \quad (1.7)$$

Здесь  $\lambda, \lambda_i, i = 1, 2, \dots, r$ , - неизвестные постоянные параметры. Подставим выражение для  $\psi$  и  $\psi_{n+1}$  из (1.7) в равенство для  $H'$  в (1.6). Учитывая условие  $H' = 0$ , получим уравнения, которые затем разрешим относительно  $\lambda$

$$\lambda = \left( \frac{dF}{dt} - \sum_{i=1}^r \lambda_i \frac{dq_i}{dt} \right) \left( \frac{dh}{dt} \right)^{-1}, \quad t = T. \quad (1.8)$$

Введем далее видоизмененные сопряженные переменные и гамильтониан

$$\begin{aligned} p &= \psi - \frac{\partial F}{\partial x}, \quad p = (p_1, p_2, \dots, p_n), \\ H &= (p, f) = H' - \psi_{n+1} - \frac{\partial F}{\partial t}. \end{aligned} \quad (1.9)$$

Отметим, что выражение в квадратных скобках равенства (1.6) равно  $d(\frac{\partial F}{\partial x_k})/dt$ . Тогда выражения (1.6), (1.7) с учетом (1.9) можно записать в виде

$$\begin{aligned} \dot{p}_k &= -\left( p, \frac{\partial f}{\partial x_k} \right) = -\frac{\partial H}{\partial x_k}, \quad H = (p, f), \\ p &= \lambda \frac{\partial h}{\partial t} + \sum_{i=1}^r \lambda_i \frac{\partial q_i}{\partial x} - \frac{\partial F}{\partial x}, \quad t = T. \end{aligned} \quad (1.10)$$

Согласно принципу максимума, задача оптимального управления свелась к краевой задаче для двух  $n$ -мерных векторных функций  $x(t)$  и  $p(t)$ . Оптимальное управление  $u(t)$  определяется из условия супремума функции  $H'$  по  $u$ , что эквивалентно условию супремума функции  $H$ , то есть

$$H(p(t), x(t), t, u(t)) = \sup_{u \in U} H(p(t), x(t), t, u). \quad (1.11)$$

Следует отметить, что не всякое управление, найденное из принципа максимума, является оптимальным. С классической формулировкой принципа максимума Понтрягина и его применением можно познакомиться в работах [1], [3]-[5].

### 1.3 Задачи управления с малыми параметром. Слабоуправляемые системы

Прикладные задачи оптимального управления в явном и неявном виде содержат малые параметры, которые характеризуют относительную малость воздействий или факторов. В этих случаях используются асимптотические методы построения оптимальных управлений, основанные на идее малого параметра.

Введение малого параметра  $\varepsilon$  оправдано в тех случаях, когда невозмущенная задача (при  $\varepsilon = 0$ ) может быть решена аналитически или численно значительно проще, чем возмущенная.

Предположим, что функции  $f, h, a, q, F$  зависят от параметра  $\varepsilon$  и имеет место их разложение в виде рядов по степеням  $\varepsilon$

$$\begin{aligned} f &= f^0(x, t, u) + \varepsilon f^1(x, t, u) + \varepsilon^2 f^2(x, t, u) + \dots \\ a &= a^0 + \varepsilon a^1 + \varepsilon^2 a^2 + \dots \\ h &= h^0(x, t) + \varepsilon h^1(x, t) + \varepsilon^2 h^2(x, t) + \dots \\ q &= q^0(x, t) + \varepsilon q^1(x, t) + \varepsilon^2 q^2(x, t) + \dots \\ F &= F^0(x, t) + \varepsilon F^1(x, t) + \varepsilon^2 F^2(x, t) + \dots \end{aligned} \quad (1.12)$$

Здесь и в ниже приводимых выражениях индексы вверху указывают номер членов в разложениях по степеням  $\varepsilon$ , а нижние индексы - номер координаты вектора. Точками обозначены высшие члены разложений.

Пусть при  $\varepsilon = 0$  задача (1.1)-(1.4), где  $f, h, a, q, F$  определены в виде (1.12), имеет единственное решение, которое может быть построено. Тогда возникает вопрос о близости точного решения задачи (1.1)-(1.4), (1.12) и приближенного, полученного при  $\varepsilon = 0$ . Также возможен другой случай, когда в (1.12) функция  $f^0$  не зависит от управляющего воздействия. В этом случае система (1.1) называется слабоуправляемой.

Предположим, что функция  $f^0(x, t, u)$  в (1.12) не зависит от параметра  $u$ . Этот случай интересен тем, что в нулевом приближении управление вообще нельзя определить в принципе. Отменим, что возможен промежуточный случай, когда векторная функция  $f^0$  зависит не от всех управляющих функций  $u_i(t)$ , а только от некоторых от них.

## 1.4 Вывод уравнений нулевого и первого приближений

Перейдем к построению приближенного решения задачи оптимального управления для слабоуправляемой системы (1.1)–(1.4) в случае, когда функции  $f, h, q, F$  представимы в виде (1.12), где  $f^0$  не зависит от  $u$ . Искомые величины  $x, p, T, \lambda, \lambda_i$  и  $J$  ищем в виде разложений в ряд по степеням  $\varepsilon$ :

$$\begin{aligned} x &= x^0 + \varepsilon x^1(t) + \dots, \\ p &= p^0(t) + \varepsilon p^1(t) + \dots, \\ T &= T^0 + \varepsilon T^1 + \dots, \\ \lambda &= \lambda^0 + \varepsilon \lambda^1 + \dots, \\ J &= J^0 + \varepsilon J^1 + \dots \end{aligned} \quad (1.13)$$

Подставим равенства (1.12), (1.13) в уравнения (1.1), (1.3), (1.4), (1.8), (1.10), разложим полученные соотношения в ряды по степеням  $\varepsilon$  и приравняем коэффициенты при  $\varepsilon^0 = 1$  и  $\varepsilon^1 = \varepsilon$ . В нулевом приближении получим

$$\begin{aligned} \dot{x}^0 &= f^0(x^0, t), \quad x^0(t_0) = a^0, \\ h^0(x^0(T^0), T^0) &= 0, \quad q^0(x^0(T^0), T^0) = 0, \\ J^0 &= F^0(x^0(T^0), T^0), \\ p_k^0 &= - \left( p^0, \frac{\partial f^0(x^0(t), t)}{\partial x_k} \right), \quad k = 1, 2, \dots, n; \\ p^0(T^0) &= \lambda^0 \frac{\partial h^0}{\partial t} + \sum_{i=1}^r \lambda_i^0 \frac{\partial q_i^0}{\partial x} - \frac{\partial F^0}{\partial x}, \end{aligned} \quad (1.14)$$

$$\begin{aligned} \lambda^0 &= \left\{ \frac{\partial F^0}{\partial t} + \left( \frac{\partial F^0}{\partial x}, f^0 \right) - \sum_{i=1}^r \lambda_i^0 \left[ \frac{\partial q_i^0}{\partial t} + \left( \frac{\partial q_i^0}{\partial x}, f^0 \right) \right] \right\} \times \\ &\times \left\{ \left[ \frac{\partial h^0}{\partial t} + \left( \frac{\partial h^0}{\partial x}, f^0 \right) \right]^{-1} \right\}, \quad t = T^0. \end{aligned}$$

Для равенств (1.1), (1.3), (1.4) выпишем уравнения первого приближения:

$$\begin{aligned} x_k^1 &= \left( \frac{\partial f_k(x^0(t), t)}{\partial x}, x^1 \right) + f_k^1(x^0(t), t, u(t)), \quad k = 1, 2, \dots, n, \\ x^1(t_0) &= a^1, \\ \left[ \frac{\partial h^0}{\partial t} + \left( \frac{\partial h^0}{\partial x}, f^0 \right) \right] T^1 + \left( \frac{\partial h^0}{\partial x}, x^1(T^0) \right) + h^1 &= 0, \\ \left[ \frac{\partial q_i^0}{\partial t} + \left( \frac{\partial q_i^0}{\partial x}, f^0 \right) \right] T^1 + \left( \frac{\partial q_i^0}{\partial x}, x^1(T^0) \right) + q_i^1 &= 0, \\ J^1 &= \left[ \frac{\partial F^0}{\partial t} + \left( \frac{\partial F^0}{\partial x}, f^0 \right) \right] T^1 + \left( \frac{\partial F^0}{\partial x}, x^1(T^0) \right) + F^1. \end{aligned} \quad (1.15)$$

В последних трех равенства (1.15) все функции от  $x, t$  берутся при значениях  $x = x^0(T^0), t = T^0$ .

## 1.5 Построение приближенного решения

Перейдем к анализу уравнений (1.14), (1.15). Общее решение системы нулевого приближения  $\dot{x} = f^0(x, t)$  из (1.14) предполагается известным и заданным в явном виде

$$x = \phi(t, c), \phi = (\phi_1, \phi_2, \dots, \phi_n), c = (c_1, c_2, \dots, c_n). \quad (1.16)$$

Здесь  $\phi$  - векторная функция,  $c$  - вектор произвольных постоянных. Разрешая равенство (1.16) относительно произвольных постоянных  $c_i$ , получим

$$g(t, x) = c, g = (g_1, g_2, \dots, g_n), \quad (1.17)$$

Функции  $g_k, k = 1, 2, \dots, n$ , являются независимыми первыми интегралами системы нулевого приближения. Для траектории  $x^0(t)$  в нулевом приближении имеем задачу Коши, задаваемую первыми двумя равенствами (1.14), ее решение выражается через функции  $\phi, g$ , введенные равенствами (1.16), (1.17)

$$x^0(t) = \phi(t, c), c = g(a^0, t^0). \quad (1.18)$$

Момент  $T^0$  окончания процесса и функционал  $J^0$  в этом приближении определяются третьим и пятым равенствами (1.14), т.е. краевые условия  $q = 0$  нулевого приближения, выполняются автоматически.

Введем матрицы размерности  $n \times n$

$$\Phi(t, c) = \left\| \frac{\partial \phi_i}{\partial c_j} \right\|, G(t, c) = \left\| \frac{\partial g_i}{\partial x_j} \right\| \quad (1.19)$$

при

$$x = \phi(t, c), i, j = 1, 2, \dots, n.$$

Равенства (1.17) и (1.18) задают преобразования, переводящие вектор  $c$  в вектор  $x$  и обратно ( $x, c \in \mathbb{R}^n$ ). Матрицы (1.19), как матрицы Якоби для этих взаимно обратных преобразований, связаны соотношением  $\Phi = G^{-1}$ . Ранг матриц равен  $n$ .

Функция  $x^1(t)$  удовлетворяет линейной неоднородной системе (1.15) (первое равенство). Соответствующая однородная система является системой в вариациях для уравнений нулевого приближения (1.14) (первое равенство), которой удовлетворяет  $x^0(t)$ . Как известно из теории обыкновенных дифференциальных уравнений, матрица  $\Phi$  из (1.19) представляет собой фундаментальную матрицу для системы в вариациях. Пользуясь этим, запишем полученное при помощи метода вариации произвольных постоянных общее решение неоднородной системы (1.15)

$$x^1(t) = \Phi(t, c)b + \Phi(t, c) \int_{t_0}^t \Phi^{-1}(\tau, c) f^1(x^0(\tau), \tau, u(\tau)) d\tau.$$

Пользуясь равенством  $\Phi^{-1} = G$ , перепишем последнее выражение в виде

$$x^1(t) = \Phi(t, c)G(t_0, c)x^1(t_0) + \Phi(t, c) \int_{t_0}^t G(\tau, c) f^1(x^0(\tau), \tau, u(\tau)) d\tau. \quad (1.20)$$



Выразим величину  $T^1$  из третьего равенства (1.15) и подставим в четвертое равенство (1.15). Получим

$$\begin{aligned} \left[ \left( \frac{\partial q_i^0}{\partial x}, x^1(T^0) \right) + q_i^1 \right] \left[ \frac{\partial h^0}{\partial x} + \left( \frac{\partial h^0}{\partial x}, f^0 \right) \right] = \\ = \left[ \left( \frac{\partial h^0}{\partial x}, x^1(T^0) \right) + h^1 \right] \left[ \frac{\partial q_i^0}{\partial t} + \left( \frac{\partial q_i^0}{\partial x}, f^0 \right) \right], \quad i = 1, \dots, r. \end{aligned} \quad (1.21)$$

Вектор  $p^0$ , как следует из (1.14), удовлетворяет линейной однородной системе, которая является сопряженной по отношению к упоминавшейся выше системе в вариациях. Поэтому фундаментальная матрица для нее равна  $(\Phi^{-1})' = G'$ , где штрих означает транспонированную матрицу. Следовательно, общее решение системы (1.14) для  $p^0(t)$  имеет вид

$$p^0(t) = G'(t, c)s, \quad s = (s_1, s_2, \dots, s_n), \quad (1.22)$$

или в скалярной форме

$$p_k^0(t) = \sum_{i=1}^n \frac{\partial g_i}{\partial x_k} s_i, \quad k = 1, 2, \dots, n.$$

Здесь  $s$  — вектор произвольных постоянных. Подставляя решение (1.22) в условие (1.14) для  $p^0(T^0)$  и учитывая равенство  $(G')^{-1} = \Phi'$ , получим

$$s = \Phi'(T^0, c) \left( \lambda^0 \frac{\partial h^0}{\partial x} + \sum_{i=1}^r \lambda_i^0 \frac{\partial q_i^0}{\partial x} - \frac{\partial F^0}{\partial x} \right) \quad (1.23)$$

при  $t = T^0$ .

## 1.6 Оптимальное управление в первом приближении

Далее находим управление в первом приближении. Подставим в функцию  $H$  из (1.9) представления для  $\text{rnf}$  из (1.13) и (1.14) и разложим эту функцию в ряд по степеням  $\varepsilon$

$$H = (p, f) = (p^0, f^0(x^0, t)) + \varepsilon \left[ (p^0, \sum_{i=1}^n \frac{\partial f^0}{\partial x_i} x_i^1) + (p^1, f^0(x^0, t)) + (p^0, f^1(x^0, t, u)) \right] + \dots$$

Здесь точками обозначены члены порядка выше первого. Из выписанных слагаемых лишь последнее зависит от  $u$ . Поэтому определение максимума  $H$  по  $u$  в (1.11) сводится в первом приближении к максимизации этого последнего слагаемого:

$$(p^0(t), f^1(x^0(t), t, u(t))) = \sup_{u \in U} (p^0(t), f^1(x^0(t), t, u)) \quad (1.24)$$

Отметим, что согласно (1.24) управление  $u(t)$  зависит только от решений нулевого приближения  $x^0(t)$  и  $p^0(t)$ . Учитывая решение (1.22), условие (1.24) можно переписать в виде

$$(G' s, f^1) = \sum_{i,j=1}^n \frac{\partial g_i}{\partial x_j} s_i f_j^1(x^0(t), t, u) \xrightarrow{u \in U} \sup. \quad (1.25)$$

*Замечание.* Управление  $u(t)$ , определяемое соотношением (1.25), может и не быть близко к оптимальному в смысле метрики пространства  $\mathbb{C}$  (т.е. по максимуму модуля разности). Однако оно будет приближенно оптимальным в смысле минимизируемого функционала. При некоторых дополнительных предположениях относительно функций  $h^0(x, t)$ ,  $f^0(x, t)$ ,  $f^1(x, t)$  имеет место утверждение: существует  $\varepsilon^* > 0$ ,  $\alpha > 0$ , что при  $\varepsilon \in (0, \varepsilon^*]$  выполнено неравенство

$$0 \leq J_{u^0}^\varepsilon - J_{u^\varepsilon}^\varepsilon \leq \alpha \varepsilon^2,$$

где  $J_{u^0}^\varepsilon$  - значение критерия качества при нулевом управлении, найденное по формуле (1.24) и  $J_{u^\varepsilon}^\varepsilon$  - значение критерия качества при оптимальном управлении. С доказательством этого факта можно познакомиться в [2].

## 1.7 Алгоритм приближенного решения задачи оптимального управления для слабоуправляемых систем

На основании полученных в предыдущем параграфе результатов сформулируем алгоритм нахождения оптимального управления, оптимальной траектории, критерия качества и времени окончания процесса с точностью до параметра  $\varepsilon^2$ .

### Общий случай ( $r \neq 0$ )

- 1) Находим общее решение системы нулевого приближения, т.е. векторную функцию  $\phi$  (см. (1.16)).
- 2) Находим  $n$  независимых первых интегралов системы нулевого приближения, т.е. векторную функцию  $g$  (см. (1.17)).
- 3) находим частное решение  $x^0(t)$  системы нулевого приближения (см. (1.18)).
- 4) Находим матрицы Якоби  $\Phi$  и  $G$  (см. (1.19)).
- 5) Находим момент времени  $T^0$  как наименьший положительный корень уравнения  $h^0(x^0(T^0), T^0) = 0$ .
- 6) Вычисляем вектор  $x^0(T^0)$ .
- 7) Вычисляем нулевое приближение  $J^0$  к оптимальному значению критерия качества по формуле  $J^0 = F^0(x^0(T^0), T^0)$ .
- 8) Находим частные производные:

$$\begin{aligned} & \frac{\partial F^0}{\partial t}, \frac{\partial F^0}{\partial x}, \frac{\partial q_i^0}{\partial t}, \\ & \frac{\partial q_i^0}{\partial x_i}, i = 1, 2, \dots, r, \frac{\partial h^0}{\partial t}, \frac{\partial h^0}{\partial x}. \end{aligned} \tag{1.26}$$

- 9) Вычисляем значения частных производных (1.26) при  $x = x^0(T^0)$ ,  $t = T^0$ .

10) Вычисляем скалярные произведения:

$$\left(\frac{\partial F^0}{\partial x}, f^0\right), \left(\frac{\partial h^0}{\partial x}, f^0\right), \left(\frac{\partial q_i^0}{\partial x}, f^0\right), i = 1, 2, \dots, r.$$

11) Вычисляем выражения для  $\lambda^0$ , зависящее от параметров  $\lambda_1^0, \lambda_2^0, \dots, \lambda_r^0$ , используя последнюю из формул группы (1.14) и результаты вычислений в предыдущих пунктах.

12) Находим вектор  $s = (s_1, s_2, \dots, s_n)$ , зависящий от параметров  $\lambda_1^0, \lambda_2^0, \dots, \lambda_r^0$ , используя равенство (1.23) и результаты предыдущего шага.

13) Находим векторную функцию  $p_0(t)$ , зависящую от параметров  $\lambda_1^0, \lambda_2^0, \dots, \lambda_r^0$ , используя равенство (1.22).

14) Используя полученные выражения для  $p^0(t)$  и  $x^0(t)$ , вычисляем скалярное произведение

$$(p^0(t), f^1(x^0(t), t, u)). \quad (1.27)$$

15) Находим управляющую функцию  $u = u(t, \lambda_1^0, \lambda_2^0, \dots, \lambda_r^0)$ , при которой выражение (1.27) достигает максимального значения.

16) Проверяем принадлежность значений найденной в предыдущем пункте функции множеству  $U$ .

17) Подставим  $x^0(t)$  и  $u(t) = u(t, \lambda_1^0, \lambda_2^0, \dots, \lambda_r^0)$  в выражение (1.20) и, используя найденные в пункте 4) матрицы  $\Phi$  и  $G$ , найдем векторную функцию  $x^1(t) = x^1(t, \lambda_1^0, \lambda_2^0, \dots, \lambda_r^0)$ .

18) Вычислим вектор  $x^1(T^0) = x^1(T^0, \lambda_1^0, \lambda_2^0, \dots, \lambda_r^0)$ , используя результат предыдущего пункта.

19) Подставим величины  $t = T^0$ ,  $x^0 = x^0(T^0)$  и  $x^1(T^0) = x^1(T^0, \lambda_1^0, \lambda_2^0, \dots, \lambda_r^0)$  в равенство (1.21). Получим систему  $r$  уравнений с  $r$  неизвестными  $\lambda_1^0, \lambda_2^0, \dots, \lambda_r^0$ .

20) Находим решение  $\lambda_1^0, \lambda_2^0, \dots, \lambda_r^0$  полученной в предыдущем пункте системы уравнений.

21) Используя найденные значения  $\lambda_1^0, \lambda_2^0, \dots, \lambda_r^0$ , вычисляем векторные функции  $p^0(t)$ ,  $u(t)$ ,  $x^1(t)$  и величину  $\lambda^0$  по формулам, полученным в пунктах 13), 15), 17) и 11) соответственно.

22) Вычислим поправку  $T^1$  к моменту  $T^0$  окончания процесса, используя третью из формул группы (1.15) и найденные в предыдущих пунктах векторные функции  $x^0(t)$ ,  $x^1(t)$ , положив  $t = T^0$ .

23) Вычислим величину  $J^1$  - поправку к функционалу критерия качества из пятого равенства группы формул (1.15).

### Частный случай ( $r = 0$ )

В этом случае процесс вычисления значительно упрощается. Алгоритм решения включает пункты с 1) по 18), однако вектор  $s$  и векторные функции  $p^0(t)$ ,  $u(t)$ ,  $x^1(t)$ , полученные при реализации пунктов 12)-18), не зависят от параметров  $\lambda_1^0, \lambda_2^0, \dots, \lambda_r^0$ . В связи с этим необходимость вычисления этих параметров и зависящих от них величин, т.е. реализация пунктов 19) - 21), отпадает. Пункты 22) и 23) вычисления поправок  $T^1$  и  $J^1$  сохраняются.

## Контрольные вопросы

1. Назовите четыре пункта, из которых состоит постановка задачи.
2. Какими свойствами должны обладать функции в пунктах (1)–(4)?
3. Из какого условия находится время окончания процесса?
4. Сформулируйте принцип максимума.
5. Какое управление можно назвать допустимым и что есть допустимая фазовая траектория?
6. Какое управление и какая фазовая траектория называются оптимальными?
7. Какие системы дифференциальных уравнений называются слабоуправляемыми?
8. На каких идеях анализа основывается метод малого параметра?
9. В каких пунктах приведённого алгоритма используется условие трансверсальности?
10. Какой из пунктов приведённого алгоритма является наиболее трудно выполнимым при его реализации?
11. В каких случаях реализация алгоритма упрощается?

## § 2. Задача о полете на максимальную дальность

### 2.1 Постановка задачи и преобразование системы к нормальному виду

Рассмотрим летательный аппарат, не имеющий собственного двигателя, обладающий начальной скоростью  $V_0$ , направленной к горизонту под углом  $\theta$ . Пусть существуют два способа управления аппаратом - за счет изменения площади несущей поверхности  $S$  и за счет изменения угла атаки  $\alpha$ . Задача — достичь максимальной дальности полета. Для этого построим математическую модель системы.

Пусть материальная точка — центр тяжести объекта — движется в плоскости  $(X_1 \times X_2)$  и закон движения задан функциями координат  $x_1$  и  $x_2$ . На объект действуют три силы: сила тяжести  $m \vec{g}$ , сила сопротивления  $\vec{R}$  и подъемная сила  $\vec{Y}$ .

В таком случае, согласно второму закону Ньютона, уравнения движения системы в проекциях на оси имеют вид

$$\begin{cases} m\ddot{x}_1 = -R \cos \theta - Y \sin \theta, \\ m\ddot{x}_2 = -mg + R \sin \theta - Y \cos \theta, \end{cases}$$

где  $\theta$  — угол, который образует вектор скорости с осью  $x_1$ . Очевидно, что подъемная сила и сила сопротивления зависят от скорости движения объекта, угла атаки, аэродинамических характеристик крыла, площади несущей поверхности и плотности среды. Из аэродинамики известно, что модули подъемной силы и силы сопротивления задаются формулами

$$R = \frac{1}{2}\rho V^2 S C_x, \quad Y = \frac{1}{2}\rho V^2 S C_y,$$

где  $S$  — площадь несущей поверхности,  $\rho$  — коэффициент плотности атмосферы,  $V$  — модуль скорости,  $C_x, C_y$  — аэродинамические характеристики крыла, которые являются для каждого конкретного случая вполне конкретными функциями от угла атаки  $\alpha$ . Таким образом, система дифференциальных уравнений, описывающих движение летательного аппарата, имеет вид

$$\begin{cases} m\ddot{x}_1 = -\frac{1}{2}\rho V^2 S(C_x \cos \theta + C_y \sin \theta), \\ m\ddot{x}_2 = -mg + \frac{1}{2}\rho V^2 S(C_x \sin \theta - C_y \cos \theta). \end{cases}$$

Учитывая, что проекции скорости на оси равны  $\dot{x}_1 = V \cos \theta, \dot{x}_2 = V \sin \theta$  и понижая порядок системы введением дополнительных переменных, получим систему дифференциальных уравнений первого порядка

$$\begin{cases} \dot{x}_1 = x_3, \\ \dot{x}_2 = x_4, \\ \dot{x}_3 = -\varepsilon \rho V S(C_x x_3 + C_y x_4), \\ \dot{x}_4 = -1 + \varepsilon \rho V S(C_x x_4 - C_y x_3), \end{cases} \quad (2.1)$$

где  $V = \sqrt{x_3^2 + x_4^2}$ ,  $\varepsilon = V_0^2(2mg)^{-1}$  — малый параметр. Полагая  $V_0 = 1$  имеем начальные условия вида:

$$\begin{cases} t_0 = 0, \\ x_1(t_0) = 0, \\ x_2(t_0) = 0, \\ x_3(t_0) = V_0 \cos \theta = \cos \theta, \\ x_4(t_0) = V_0 \sin \theta = \sin \theta, \end{cases} \quad (2.2)$$

где  $0 < \theta < \frac{\pi}{2}$ . Нужно достичь максимальной дальности полета управляемого аппарата (величина  $x_1$ ) в момент времени  $T$ , когда высота  $x_2$  обратится в ноль. Таким образом, добавляется краевое условие

$$x_2(T) = 0, \quad (2.3)$$

а критерий качества принимает вид  $J = -x_1(T) \rightarrow \min$ .

Задача состоит в нахождении оптимальной траектории  $x(t) = (x_1(t), x_2(t), x_3(t), x_4(t))$ , которая при  $t_0 \leq t \leq T$  удовлетворяет системе (2.1), начальным условиям (2.2), крайнему условию (2.3) и доставляет минимум функционалу  $J$ .

## 2.2 Пакет Mathematica: полезная информация

Wolfram Mathematica (далее Mathematica) это пакет символьной математики, в котором заложено множество функций, а также есть открытая среда, позволяющая дополнять пакет своими собственными расширениями. Эта система является самой мощной вычислительной системой в мире.

Интерфейс пакета строится из нескольких базовых понятий: Тетрадь (Notebooks), Ячейка (Cell) и Палитра (Palletes). Тетрадью называется файл, с которым работает пользователь. В нем создаются и вычисляются формулы, строятся графики и таблицы. Тетрадь состоит из ячеек. Вся информация, которая есть в тетради, хранится в его ячейках. Как только Вы в пустом новом файле наберете хотя бы один символ, Mathematica создаст для него ячейку. Ячейка также является минимальной единицей, которую можно вычислить. Для выполнения содержимого ячейки надо поставить курсор в любое место ячейки и нажать Shift+Enter.

Для быстрого доступа к функциям существуют специальные типы окон, которые называются палитрами. Палитры содержат окна с кнопками, которые выполняют действия. Действия могут быть различными: от добавления греческой буквы, до раскрытия скобок в алгебраическом выражении. Ввод данных осуществляется в ячейки. Пакет поддерживает кириллицу и греческие буквы наравне с английским алфавитом. Вы можете смело называть переменные русскими буквами, также как и греческими. В то же время, идентификаторы различаются по регистру, т.е. переменная  $A$  отличается от переменной  $a$ .

Основой вычислений в Mathematica являются переменные. Для того чтобы объявить переменную, достаточно просто написать ее имя. Для того, чтобы что-либо положить в переменную, используется операция присвоения  $=$ . Например,  $x = 1$ . В системе Mathematica реализованы сотни специальных функций, используемых в чистой и прикладной математике. Все функции должны начинаться с заглавной буквы, например,  $Sin[x]$ ,  $Cos[x]$ . Этот пакет может решать широкий класс обыкновенных дифференциальных уравнений и уравнений в частных производных. Рассмотрим пример

$$y' + y = \sin x,$$

код в Mathematica будет следующим

```
DSolve[y'[x] + y[x] == Sin[x], y[x], x]
```

Команда  $DSolve$  аналитически решает обыкновенные дифференциальные уравнения и системы таких уравнений. Результат выполнения команды

```
[x] -> E^-x C[1] + 1/2 (-Cos[x] + Sin[x])
```

Команда  $NDSolve$  решает начальную задачу для дифференциальных уравнений приближенно

```
s = NDSolve[{y'[x] == y[x] Cos[x + y[x]], y[0] == 1}, y, {x, 0, 30}]
```

Также реализована возможность изображения графиков. Это можно сделать с помощью следующих команд: *Plot*, *ParametricPlot*, *Plot3D* и так далее. Подробную информацию о всех командах и их свойствах можно получить в разделе *Help* в самом пакете. Раздел *Help* вызывается путем выделения команды и нажатием клавиши *F1*.

## 2.3 Реализация алгоритма решения задачи в пакете Mathematica

Зададим систему уравнений (2.1), начальные (2.2) и краевое (2.3) условия. Уравнения системы и граничных условий представляют собой уже готовые разложения в ряды, следовательно, их можно записать в обозначениях (1.12)

Зададим начальные условия: момент времени  $t_0$  и вектор  $a = a^0$

```
t0 = 0;
```

```
a0 = {0, 0, Cos[θ], Sin[θ]};
```

Зададим краевые условия  $h(x(T), T) = x_2(T)$

```
h[x_, t_] := x[[2]];
```

Зададим критерий качества  $F(x(T), T) = -x_1(T)$

```
F0[x_, t_] := -x[[1]];
```

Зададим функцию  $f(t, x, u, \epsilon)$  – правую часть системы (2.2).

```
f[t_, x_, α_, s_, ε_] := {x[[3]], x[[4]], -ε ρ V s (Cx[α] x[[3]] + Cy[α] x[[4]]),  
-1 + ε ρ V s (Cy[α] x[[3]] - Cx[α] x[[4]])}.
```

Найдем нулевое приближение функции  $f$ . Для этого разложим ее в ряд по  $\epsilon$  и найдем коэффициент при нулевой степени

```
f0[t_, x_] := Module[{ε}, SeriesCoefficient[f[t, x, α, s, ε], {ε, 0, 0}]]
```

```
f0[t, x]
```

```
{x[[3]], x[[4]], 0, -1}
```

Переменная  $xx$  необходима для символьного представления переменной для решения ДУ в следующем шаге.

```
xx = Table[xi[t], {i, 1, 4}];
```

Шаг 1. Найдем общее решение системы нулевого приближения, то есть векторную функцию  $\varphi$ . Запишем систему нулевого приближения

```
system0 = Table[xi'[t] == f0[t, xx][[i]], {i, 1, 4}]
```

```
{x1'[t] == x3[t], x2'[t] == x4[t], x3'[t] == 0, x4'[t] == -1}
```

Решим систему

```
φ[t_] := Evaluate[xx /. DSolve[system0, xx, t][[1]]]
```

```
φ[t]
```

```
{C[1] + t C[2], -t^2/2 + C[3] + t C[4], C[2], -t + C[4]}
```

Получили решение системы нулевого приближения

Шаг 2. Находим  $n$  независимых первых интегралов системы нулевого приближения, то есть векторную функцию  $g$

```
c = Table[C[i], {i, 1, 4}];
```

```
gEquations = Thread[xx == φ[t]]
```

```
{x1[t] == C[1] + t C[2], x2[t] == -t^2/2 + C[3] + t C[4], x3[t] == C[2], x4[t] == -t + C[4]}
```

```
g[t_] := Evaluate[c /. Solve[gEquations, c][[1]]]
```

```
g[t]
```

$$\{x_1[t] - tx_3[t], x_3[t], \frac{1}{2}(-t^2 + 2x_2[t] - 2tx_4[t]), t + x_4[t]\}$$

Шаг 3. Находим частное решение  $x^0(t)$  системы нулевого приближения и определим векторную функцию  $\phi_0(t)$

$$\begin{aligned} \phi0Equations &= \text{Thread}[a0 == \phi[t0]] \\ \{0 == C[1], 0 == C[3], \text{Cos}[\theta] == C[2], \text{Sin}[\theta] == C[4]\} \\ \phi0[t_] &:= \text{Evaluate}[\phi[t] /. \text{Solve}[\phi0Equations, c][[1]]] \\ \phi0[t] & \end{aligned}$$

$$\left\{ t \text{Cos}[\theta], -\frac{t^2}{2} + t \text{Sin}[\theta], \text{Cos}[\theta], -t + \text{Sin}[\theta] \right\}$$

Шаг 4. Находим матрицы Якоби  $\Phi$  и  $G$

$$\begin{aligned} \Phi[t_] &:= \text{Evaluate}[D[\phi[t], \{c, 1\}]]; \\ \Phi[t] & // \text{MatrixForm} \end{aligned}$$

$$\begin{pmatrix} 1 & t & 0 & 0 \\ 0 & 0 & 1 & t \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\begin{aligned} G[t_] &:= \text{Evaluate}[D[g[t], \{xx, 1\}]]; \\ G[t] & // \text{MatrixForm} \end{aligned}$$

$$\begin{pmatrix} 1 & 0 & -t & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & -t \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Шаг 5. Находим момент времени  $T^0$  как наименьший положительный корень уравнения  $h^0(x^0(T^0), T^0) = 0$

$$\begin{aligned} T0 &= (t /. \text{Solve}[h[\phi0[t], t] == 0, t])[2] \\ 2 \text{Sin}[\theta] & \end{aligned}$$

Шаг 6. Вычисляем вектор  $x^0(T^0)$

$$\phi0[T0]$$

$$\{2 \text{Cos}[\theta] \text{Sin}[\theta], 0, \text{Cos}[\theta], -\text{Sin}[\theta]\}$$

Шаг 7. Вычисляем нулевое приближение  $J^0$  к оптимальному значению критерия качества по формуле  $J^0 = F^0(x^0(T^0), T^0)$

$$\begin{aligned} J0 &= F0[\phi0[T0], T0] \\ -2 \text{Cos}[\theta] \text{Sin}[\theta] & \end{aligned}$$

Шаг 8. Находим частные производные  $F^0$  и  $h^0$

$$\begin{aligned} \text{xVec} &= \text{Table}[xi, \{i, 1, 4\}] \\ \{x_1, x_2, x_3, x_4\} & \end{aligned}$$

$$\begin{aligned} F0t0 &= D[F0[xVec, t], t] \\ 0 & \end{aligned}$$

$$\begin{aligned} F0x0 &= D[F0[xVec, t], \{xVec, 1\}] \\ \{-1, 0, 0, 0\} & \end{aligned}$$

$$\begin{aligned} h0t0 &= D[h[xVec, t], t] \\ 0 & \end{aligned}$$

$$\begin{aligned} h0x0 &= D[h[xVec, t], \{xVec, 1\}] \\ \{0, 1, 0, 0\} & \end{aligned}$$

Шаг 9. Вычисляем выражение для  $\lambda^0$ , используя последнюю из формул (1.14) и результаты предыдущих вычислений

$$\lambda0 = \frac{F0t0 + F0x0.f0[T0, \phi0[T0]]}{h0t0 + h0x0.f0[T0, \phi0[T0]]}$$



$\text{Cot}[\theta]$

Шаг 10. Находим вектор  $s = (s_1, s_2, \dots, s_n)$ , используя формулу (1.23) и  $\lambda^0$   
 $\text{ss} = \text{Transpose}[\Phi[\text{T0}]].(\lambda^0 \mathbf{h0x0} - \mathbf{F0x0})$

$\{1, 2\text{Sin}[\theta], \text{Cot}[\theta], 2\text{Cos}[\theta]\}$

Шаг 11. Находим векторную функцию  $p^0(t)$ , используя равенство (1.22)

$\text{p0}[t\_]:= \text{Transpose}[G[t]].\text{ss}$

$\text{p0}[t]$

$\{1, \text{Cot}[\theta], -t + 2\text{Sin}[\theta], 2\text{Cos}[\theta] - t\text{Cot}[\theta]\}$

Шаг 12. Найдем первое приближение функции  $f$ . Для этого разложим ее в ряд по  $\epsilon$  и найдем коэффициент при первой степени

$\text{f1}[t\_ , x\_ , \alpha\_ , s\_ ] := \text{Module}[\{\epsilon\}, \text{SeriesCoefficient}[f[t, x, \alpha, s, \epsilon], \{\epsilon, 0, 1\}]]$

$\text{f1}[t, x, \alpha, s]$

$\{0, 0, -sV\rho\text{Cx}[\alpha]x[[3]] - sV\rho\text{Cy}[\alpha]x[[4]], sV\rho\text{Cy}[\alpha]x[[3]] - sV\rho\text{Cx}[\alpha]x[[4]]\}$

Шаг 13. Найдем скалярное произведение из (1.24)

$\text{hamilton} = \text{p0}[t].\text{f1}[t, \phi 0[t], \alpha, s]$

$(2\text{Cos}[\theta] - t\text{Cot}[\theta])(stV\rho\text{Cx}[\alpha] + sV\rho\text{Cos}[\theta]\text{Cy}[\alpha] - sV\rho\text{Cx}[\alpha]\text{Sin}[\theta]) + (-t + 2\text{Sin}[\theta])$

$(-sV\rho\text{Cos}[\theta]\text{Cx}[\alpha] + stV\rho\text{Cy}[\alpha] - sV\rho\text{Cy}[\alpha]\text{Sin}[\theta])$

Упростим полученное выражение

$\text{Factor}[\text{FullSimplify}[\text{hamilton}]]$

$-sV\rho(-2 + t\text{Csc}[\theta])(-2\text{Cos}[\theta]\text{Cx}[\alpha] + t\text{Cot}[\theta]\text{Cx}[\alpha] + t\text{Cy}[\alpha] + \text{Cos}[2\theta]\text{Csc}[\theta]\text{Cy}[\alpha])\text{Sin}[\theta]$

Разделим на множитель, не зависящий от управления (максимум функции не поменяется)

$\text{hamiltonFactored} = \text{FullSimplify}[\text{hamilton}/(V\rho\text{Sin}[\theta](-2 + t\text{Csc}[\theta]))]$

$-s((-2\text{Cos}[\theta] + t\text{Cot}[\theta])\text{Cx}[\alpha] + (t + \text{Cos}[2\theta]\text{Csc}[\theta])\text{Cy}[\alpha])$

$\text{hamiltonSimple} = \text{FullSimplify}[\text{hamiltonFactored}/((2\text{Cos}[\theta] - t\text{Cot}[\theta]))]$

$s \left( \text{Cx}[\alpha] + \frac{(t + \text{Cos}[2\theta]\text{Csc}[\theta])\text{Cy}[\alpha]\text{Sec}[\theta]}{-2 + t\text{Csc}[\theta]} \right)$

Шаг 14. Находим управляющую функцию  $u = (t, \lambda_1^0, \lambda_2^0, \dots, \lambda_r^0)$ , при которой выражение (1.27) достигает максимального значения.

Найдем управление  $u_0$ , которое доставляет минимум уравнению

$$s \left( \text{Cx}[\alpha] + \frac{(t + \text{Cos}[2\theta]\text{Csc}[\theta])\text{Cy}[\alpha]\text{Sec}[\theta]}{-2 + t\text{Csc}[\theta]} \right)$$

Возьмем производную по  $\alpha$ . Она должна равняться нулю, чтобы управление было оптимальным

$\text{prime} = D[\text{hamiltonSimple}, \alpha]$

$$s \left( \text{Cx}'[\alpha] + \frac{(t + \text{Cos}[2\theta]\text{Csc}[\theta])\text{Sec}[\theta]\text{Cy}'[\alpha]}{-2 + t\text{Csc}[\theta]} \right)$$

Найдем отношение  $\frac{\text{Cx}'(\alpha)}{\text{Cy}'(\alpha)}$

$\text{cxPrimeOverCyPrime} = (\text{Cx}'[\alpha]/.\text{Solve}[\text{prime} == 0, \text{Cx}'[\alpha]][[1]])/\text{Cy}'[\alpha]$   
 $-\frac{(t + \text{Cos}[2\theta]\text{Csc}[\theta])\text{Sec}[\theta]}{-2 + t\text{Csc}[\theta]}$

Найдем вторую производную по  $\alpha$ . Она должна быть больше 0, чтобы это была точка максимума

$\text{prime2} = D[\text{hamiltonSimple}/s, \{\alpha, 2\}]$

$$Cx''[\alpha] + \frac{(t + \cos[2\theta] \csc[\theta]) \sec[\theta] Cy''[\alpha]}{-2 + t \csc[\theta]} > 0$$

Заметим, что коэффициент при  $C_y'''(\alpha)$  равен отношению  $\frac{C_x'(\alpha)}{C_y'(\alpha)}$ , взятому с противоположным знаком. Произведем замену

**CoefficientList[prime2, Cy''[\alpha]][[2]] == -cxPrimeOverCyPrime**

**True**

**True**

**prime21 = prime2/.CoefficientList[prime2, Cy''[\alpha]][[2]] → - $\frac{Cx'[\alpha]}{Cy'[\alpha]}$**

$$Cx''[\alpha] - \frac{Cx'[\alpha] Cy''[\alpha]}{Cy'[\alpha]}$$

**prime21 ≥ 0**

$$Cx''[\alpha] - \frac{Cx'[\alpha] Cy''[\alpha]}{Cy'[\alpha]} \geq 0$$

Функции  $C_x$ ,  $C_y$  есть некоторые аэродинамические характеристики крыла. Для придания большей конкретности дальнейшим вычислениям зададим эти функции в виде

$$\begin{aligned} Cx[\alpha_] &:= 1 - \cos[2\alpha_0] \cos[2\alpha]; \\ Cy[\alpha_] &:= k \sin[2\alpha_0] \sin[2\alpha]; \end{aligned}$$

где  $\alpha_0$ ,  $K$ —постоянные. Величина  $K$  равна максимальному качеству аппарата  $\max\left(\frac{C_x}{C_y}\right)$ , а  $\alpha_0$ —угол атаки, при котором достигается максимальное качество.

Найдем значения  $\alpha$  и  $\alpha_0$ , при которых вторая производная неотрицательна

**Reduce[{prime21 ≥ 0, α > 0, α < Pi/2, α0 > 0, α0 < Pi/2}, {α, α0}]**

$$(0 < \alpha < \frac{\pi}{4} \&\& 0 < \alpha_0 \leq \frac{\pi}{4}) \parallel (\frac{\pi}{4} < \alpha < \frac{\pi}{2} \&\& \frac{\pi}{4} \leq \alpha_0 < \frac{\pi}{2})$$

Найдем  $\alpha$  из отношения производных в предположении  $0 \leq \alpha_0 \leq \pi/4$

$$\frac{Cx'[\alpha]/Cy'[\alpha]}{\cot[2\alpha_0] \tan[2\alpha]}$$

$k$

**tanEquation = Cx'[\alpha]/Cy'[\alpha] == cxPrimeOverCyPrime;**

**tan2α = Tan[2α]/.Solve[tanEquation, Tan[2α]][[1]];**

**α00[t\_]:=Evaluate[1/2ArcTan[tan2α]]**

**α00[t]**

$$-\frac{1}{2} \text{ArcTan} \left[ \frac{k(t + \cos[2\theta] \csc[\theta]) \sec[\theta] \tan[2\alpha_0]}{-2 + t \csc[\theta]} \right] \quad (2.4)$$

Функция  $S(t)$  выбирается в соответствии со знаком выражения

$$a[t_] := Cx[\alpha_{00}[t]] - Cy[\alpha_{00}[t]] \frac{Cx'[\alpha_{00}[t]]}{Cy'[\alpha_{00}[t]]};$$

**s00[t\_]:=Piecewise[{ {s1, a[t] ≥ 0} }, s2];**

Введем значение модуля скорости

$$V[x_] := \text{Sqrt}[x[[3]]^2 + x[[4]]^2];$$

Таким образом, мы построили нулевое приближение к оптимальному управлению.

Шаг 15. Полученное управление является ограниченной функцией и, следовательно, принадлежит множеству  $U$ .

Шаг 16. Далее будем находить поправку к нулевому приближению  $x^1(t)$ , используя формулу (1.20)

**underIntegral** =  $G[\tau].f1[\tau, \phi0[\tau], \alpha00[\tau], s00[\tau]]/.V \rightarrow V[\phi0[\tau]]$ ;

Шаг 17. Вычислим вектор  $x^1(t)$ , используя результат предыдущего пункта. Но построенное пакетом выражение не несет в себе требующейся аналитической информации. Поэтому введем числовые значения параметров и определим функцию  $x^1(t)$  и вычислим вектор  $x_0(T^0)$  и  $x_1(T^0)$

$\theta = \text{Pi}/4$ ;  $\alpha0 = \text{Pi}/8$ ;  $k = 1.5$ ;  $s1 = 1$ ;  $s2 = 2$ ;  $\rho = 1$ ;  $\epsilon = .01$ ;

$x01 = \{0, 0, 0, 0\}$ ;

Шаг 18. Поправка к нулевому приближению

$\phi11[t\_]:= \text{Evaluate}[N[\Phi[t].G[t0].x01 + \Phi[t].NIntegrate[\text{underIntegral}, \{\tau, t0, t\}]]]$ ;

Первое приближение

$\phi1[t\_]:= \phi0[t] + \epsilon \phi11[t]$

Шаг 19-21. Эти шаги пропускаем, так как  $r = 0$ .

Шаг 22. Вычислим поправку  $T^1$  к моменту  $T^0$  окончания процесса, используя третью из формул группы (1.15)

$T11 = T11/. \text{Solve}[D[h[xVec, t], \{xVec, 1\}].f0[T0, \phi0[T0]]T11 + D[h[xVec, t], \{xVec, 1\}].$

$\phi11[T0] == 0, T11][[1]]$

0.668026

Первое приближение момента окончания

$T1 = T0 + \epsilon T11$

1.42089

Шаг 23. Найдем величину  $J^1$  - поправку к функционалу критерия качества из пятого равенства группы формул (1.15)

$J11 = (F0t0 + F0x0.f0[T0, \phi0[T0]])T1 + F0x0.\phi1[T0]$

-2.00097

Новое значение функционала качества

$J1 = J0 + \epsilon J11$

-1.02001

## 2.4 Анализ полученных результатов

С помощью пакета Mathematica мы рассмотрели задачу о полете на максимальную дальность. Нужно было отыскать оптимальное управление. Было получено управление, близкое к оптимальному в смысле минимизации целевого функционала. При заданных параметрах системы возможно уточнить оптимизируемую величину и время полета. Но для слабоуправляемых систем величина поправки незначительна и имеет одинаковый порядок с параметром  $\epsilon^2$ , определяющим величину управляющего воздействия.

Ниже приводится график зависимости высоты полета  $x_2$  от дальности  $x_1$ , а также графики зависимости этих параметров от времени при управлении (2.4), полученном в предыдущем пункте. Согласно теории это управление близко к оптимальному.

График зависимости расстояния от времени

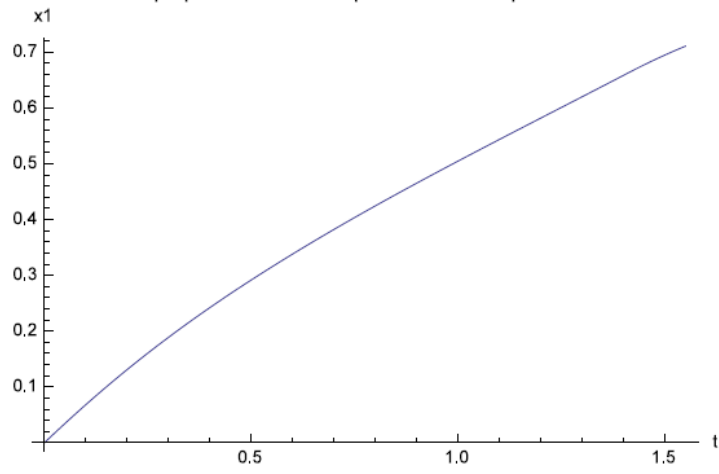


График зависимости высоты от времени

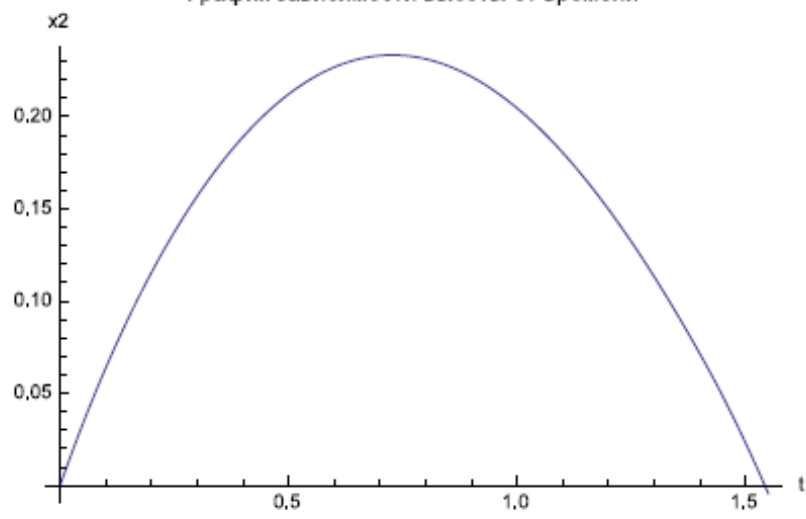
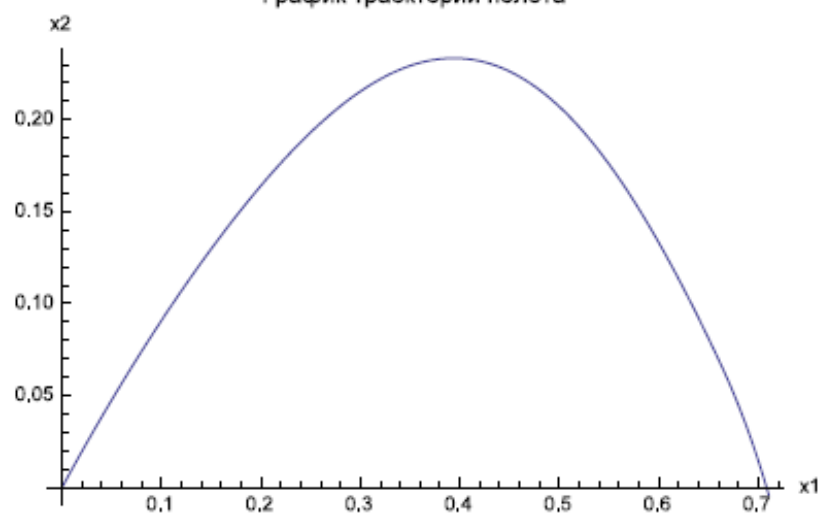


График траектории полета



## Контрольные вопросы

1. Является ли рассматриваемая здесь система слабоуправляемой, слабовозмущённой?
2. Как перейти к безразмерным переменным?
3. При каком ограничении на параметры системы она становится слабоуправляемой?
4. Как находится общее и частное решения системы нулевого приближения?
5. Сформулируйте принцип максимума Понтрягина.
6. Что такое аэродинамическая характеристика крыла (системы)?
7. Почему найденное допустимое управление является оптимальным?

## § 3. Метод малого параметра в задаче управления движением трехосного твердого тела

### 3.1 Постановка задачи

Ряд задач, представляющих значительный прикладной интерес, приводит к исследованию управляемых движений твердого тела относительно центра масс. Эти движения описываются системой дифференциальных уравнений Эйлера:

$$\begin{aligned} J_1 \dot{\omega}_1 + (J_3 - J_2) \omega_2 \omega_3 &= u_1 + \varepsilon f_1(\omega_1, \omega_2, \omega_3), \\ J_2 \dot{\omega}_2 + (J_1 - J_3) \omega_1 \omega_3 &= u_2 + \varepsilon f_2(\omega_1, \omega_2, \omega_3), \end{aligned} \quad (3.1)$$

$$\begin{aligned} J_3 \dot{\omega}_3 + (J_2 - J_1) \omega_1 \omega_2 &= u_3 + \varepsilon f_3(\omega_1, \omega_2, \omega_3), \\ \omega_1(0) &= \omega_1^0, \omega_2(0) = \omega_2^0, \omega_3(0) = \omega_3^0. \end{aligned} \quad (3.2)$$

Как видно, второе и третье уравнение получаются из первого циклической перестановкой индексов. Здесь  $J_1, J_2, J_3$  - главные центральные моменты инерции,  $\omega_1, \omega_2, \omega_3$  - проекции угловой скорости  $\omega$  на главные центральные оси инерции,  $\omega^0 = (\omega_1^0, \omega_2^0, \omega_3^0)$  - вектор начального состояния,  $u_1, u_2, u_3$  - управляющие функции, удовлетворяющие условию

$$||u||^2 = u_1^2 + u_2^2 + u_3^2 \leq 1. \quad (3.3)$$

Твердое тело подвергается действию малых возмущающих моментов, зависящих от скорости вращения, они представлены функциями  $f_i(\omega_1, \omega_2, \omega_3)$ ,  $i = 1, 2, 3$ .

Так как по предположению твердое тело является трехосным, то все его главные центральные моменты инерции отличны от нуля и, не ограничивая общности рассуждений, можно считать, что  $J_1 > J_2 > J_3$ .

Для возмущенной управляемой системы ( $\varepsilon \neq 0$ ) рассматривается задача оптимального по быстродействию торможения вращений. Требуется перевести фазовую точку системы из начального состояния в начало координат за минимальное время  $T$ :

$$\omega_i(T) = 0, i = 1, 2, 3. \quad (3.4)$$

Полагая  $\varepsilon = 0$  в правых частях системы (3.1) мы получаем управляемую систему нулевого приближения. Следовательно, система (3.1) в целом не является слабоуправляемой. Тем не менее для решения поставленной задачи в данном случае может быть применим метод малого параметра.

Сделав в (3.1) замену переменных  $J_i \omega_i = x_i$ ,  $i = 1, 2, 3$ . Получим систему уравнений:

$$\begin{aligned} \dot{x}_i &= u_i + a_i x_k x_l + \varepsilon \phi_i(x_1, x_2, x_3), \\ x_i(0) &= x_i^0, \\ \text{sign } a_i &= (-1)^{i+1}, \\ a_1 + a_2 + a_3 &= 0, 0 \leq \varepsilon < 1, i, k, l \in \{1, 2, 3\}, i \neq k, k \neq l, i \neq l. \end{aligned} \quad (3.5)$$

### 3.2 Случай невозмущенной системы

Положим в (3.5)  $\varepsilon = 0$ . Введем обозначения:

$$x = (x_1, x_2, x_3), u = (u_1, u_2, u_3), (x, u) = \sum x_i u_i, l^2 = (x, x) = \sum x_i^2, \|x\| = \sqrt{(x, x)}.$$

Умножим первое из уравнений системы (3.5) на  $x_1$ , второе на  $x_2$ , третье на  $x_3$  и сложим полученные выражения:

$$x_1 \dot{x}_1 + x_2 \dot{x}_2 + x_3 \dot{x}_3 = \sum_{i=1}^3 u_i x_i,$$

откуда следует

$$\dot{l} = (u, \eta), \eta = x l^{-1}, l(0) = \|x_0\| = l_0. \quad (3.6)$$

Из (3.3) и (3.6) с учетом соотношения  $\|\eta\| = 1$  следует  $|\dot{l}| = |(u, \eta)| \leq \|u\| \|\eta\| = \|u\| \leq 1$ . При возрастании времени в процессе успокоения твердого тела расстояние  $l(t)$  от фазовой точки  $x(t) = (x_1(t), x_2(t), x_3(t))$  до начала координат стремится к нулю, функция  $l(t)$  убывает и, следовательно, ее производная  $\dot{l}$  должна быть отрицательной при  $0 < t \leq T$ .

При этом модуль скорости  $\dot{l}(t)$  для достижения оптимального быстрогодействия должен быть наибольшим. Учитывая сделанное замечание, а также ограничение  $|\dot{l}(t)| \leq 1$ , находим  $\dot{l}(t) = -1$ . Интегрируя это равенство с учетом начальных условий находим  $l(t) = t - l_0$ . В нулевом приближении время  $T^0$  успокоения твердого тела находится из равенства  $l(T^0) = T^0 - l_0 = 0$ , следовательно,  $T^0 = l_0$ .

Оптимальное управление в нулевом приближении находим из равенства  $\dot{l} = (x, u) = -1$ , откуда находим  $u_i(t) = -x_i(t) l^{-1}(t)$ ,  $i = 1, 2, 3$ ,  $0 \leq t \leq T^0$ .

Решение задачи оптимального управления в нулевом приближении будет завершено после нахождения оптимальной фазовой траектории, т.е. решения системы дифференциальных уравнений:

$$\begin{aligned}
\dot{x}_i &= a_i x_k x_l - x_i l^{-1}(t), \\
x_i(0) &= x_i^0, \\
\text{sign } a_i &= (-1)^{i+1}, \\
a_1 + a_2 + a_3 &= 0, 0 \leq \varepsilon < 1, i, k, l \in 1, 2, 3, i \neq k, k \neq l, i \neq l.
\end{aligned} \tag{3.7}$$

Проведем традиционные в этом случае преобразования.

### 3.3 Эллиптические функции I-ого рода, их применение

Рассмотрим интеграл

$$w(u) = \int_0^u \frac{dy}{\sqrt{(1-y^2)(1-k^2y^2)}}, \quad u \in [-1, 1].$$

Этот интеграл является несобственным интегралом I-ого рода, т.к. подинтегральная функция неограничена на любом интервале, содержащем точки  $y = \pm 1$ ,  $y = \pm |k|^{-1}$ . Так как производная

$$w'(u) = \frac{1}{\sqrt{(1-u^2)(1-k^2u^2)}} > 0, \tag{3.8}$$

то функция  $w(u)$  строго возрастает на области определения и поэтому имеет обратную, которую обозначим  $u = sn(w)$  и назовем эллиптическим синусом. Из равенств

$$\begin{aligned}
sn^2 w + cn^2 w &= 1, \\
k^2 sn^2 w + dn^2 w &= 1.
\end{aligned} \tag{3.9}$$

по определению получаем эллиптический косинус  $cn(w)$  и эллиптический  $k$ -косинус  $dn(w)$ . Из (3.8) и определения обратной функции следует:

$$(sn(w))' = \frac{1}{w'(y)} \Big|_{y=sn(w)} = \sqrt{(1-y^2)(1-k^2y^2)} \Big|_{y=sn(w)} = cn(w)dn(w).$$

Для получения производной от  $cn(w)$ , а так же от  $dn(w)$  нужно продифференцировать соотношения (3.9) что дает

$$\begin{aligned}
\frac{d}{dw} cn(w) &= -sn(w)dn(w), \\
\frac{d}{dw} dn(w) &= -k^2 sn(w)cn(w).
\end{aligned}$$

Применим определенные выше эллиптические функции для нахождения оптимальных фазовых траекторий в задаче успокоения твердого тела за минимальное время.

Сделаем в системе уравнений (3.7) замену переменных  $x_i = z_i l(t)$ ,  $i = 1, 2, 3$ , получим систему дифференциальных уравнений:

$$\begin{aligned} \dot{z}_i &= a_i z_k z_l l(t), \quad z_i(0) = x_{0i} l_0^{-1}, \\ i, k, l &\in 1, 2, 3, \quad i \neq k, i \neq l, k \neq l, \end{aligned}$$

первые интегралы которой имеет вид:

$$z_1^2 + \alpha^2 z_2^2 = c_1^2, \quad z_3^2 + \beta^2 z_2^2 = c_2^2. \quad (3.10)$$

Произвольные постоянные  $c_1$  и  $c_2$  с учетом начальных условий из (3.10) запишутся в виде:

$$c_1^2 = (x_{01}^2 + \alpha^2 x_{02}^2) l_0^{-2}, \quad c_2^2 = (x_{03}^2 + \beta^2 x_{02}^2) l_0^{-2}.$$

В контексте рассматриваемой задачи естественным является предположение, что вектор начальных условий отличен от 0. В частности достаточно предположить, что  $x_{20} \neq 0$ , откуда следует, что произвольные постоянные  $c_1, c_2$  так же отличны от нуля. Выразим  $z_1$  и  $z_2$  из (3.10) и, подставив полученное выражение во второе из уравнений системы (3.5) получим, уравнение для  $z_2$ :

$$\dot{z}_2 = a_2 \sqrt{c_1^2 - \alpha^2 z_2^2} \sqrt{c_2^2 - \beta^2 z_2^2} l(t)$$

В выражении справа вынесем из под корня  $c_1$  и  $c_2$  и сделаем замену переменной  $\alpha z_2 / c_1 = y$  и положим  $k^2 = (c_1 \beta)^2 / (c_2 \alpha)^2 < 1$ . Если последнее неравенство не выполняется, то нужно сделать замену  $\beta z_2 / c_2 = y$  и положить  $(k')^2 = (c_2 \alpha) / (c_1 \beta)$ . Так как  $(k')^2$  является обратной величиной для  $k^2$ , то в случае  $k^2 > 1$ , величина  $(k')^2 < 1$ , в результате такой замены получим

$$\dot{y} = c_2 \alpha a_2 \sqrt{(1 - y^2)(1 - k^2 y^2)} l(t),$$

где  $l(t) = l_0 - t$ . Разделяя переменные в этом дифференциальном уравнении находим:

$$\begin{aligned} \frac{dy}{\sqrt{(1 - y^2)(1 - k^2 y^2)}} &= \alpha c_2 a_2 (l_0 - t) dt \\ \int_{y_0}^y \frac{dy_1}{\sqrt{(1 - y_1^2)(1 - k^2 y_1^2)}} &= \alpha c_2 a_2 (l_0 t - \frac{t^2}{2}) \\ \int_0^y \frac{dy_1}{\sqrt{(1 - y_1^2)(1 - k^2 y_1^2)}} &= \alpha c_2 a_2 (l_0 t - \frac{t^2}{2}) + \int_0^{y_0} \frac{dy}{\sqrt{(1 - y^2)(1 - k^2 y^2)}} \end{aligned}$$

Предварительно введем обозначение:

$$\Phi(y) = \int_0^y \frac{dy_1}{\sqrt{(1 - y_1^2)(1 - k^2 y_1^2)}} = w(t).$$

Тогда имеем  $y = \Phi^{-1}(w(t))$ . Согласно введенному ранее определению  $y(t) = sn(w(t))$ . Далее, используя равенства (3.8), находим

$$z_1^2 = c_1^2 - \frac{c_1^2}{\alpha^2} \alpha^2 sn^2 w(t) = c_1^2 cn^2 w(t),$$



т.е.  $z_1(t) = c_1 cn(w(t))$ ;  $z_2^2(t) = c_2^2 - \beta^2 \frac{c_1^2}{\alpha^2} sn^2 w(t) = c_2^2(1 - \frac{\beta^2 c_1^2}{\alpha^2 c_2^2} sn^2 w(t)) = c_2^2(1 - k^2 sn^2 w(t)) = c_2^2 dn^2 w(t)$ ;  $z_3(t) = c_2 dn(w(t))$ . Окончательно выражения для координатных функций фазовых траекторий системы (3.1) - (3.2) приобретают вид:

$$\begin{aligned} x_1(t) &= c_1 l(t) cn(\alpha c_2 a_2 (l_0 t - \frac{t^2}{2}) + c_3), \\ x_2(t) &= \frac{c_1}{\alpha} l(t) sn(\alpha c_2 a_2 (l_0 t - \frac{t^2}{2}) + c_3), \\ x_3(t) &= c_2 l(t) dn(\alpha c_2 a_2 (l_0 t - \frac{t^2}{2}) + c_3), \end{aligned} \quad (3.11)$$

где постоянные  $c_1, c_2, c_3$  выражаются через начальные значения (3.2) по формулам:

$$c_1^2 = (x_{01}^2 + \alpha^2 x_{02}^2) l_0^{-2}, \quad c_2^2 = (x_{03}^2 + \beta^2 x_{02}^2) l_0^{-2}, \quad c_3 = \int_0^{y_0} \frac{dy}{\sqrt{(1-y^2)(1-k^2 y^2)}}.$$

### 3.4 Принцип Беллмана, его применение в случае $\varepsilon \neq 0$

Перейдем к рассмотрению случая  $\varepsilon \neq 0$ . Применим принцип Беллмана: Оптимальное управление обладает тем свойством, что для любого начального состояния и выбранного начального управления последующее оптимальное управление совпадает с исходным оптимальным управлением относительно состояния, получающегося в результате применения оптимального управления.

Пусть  $x$  - произвольная точка фазового пространства,  $T(x, \varepsilon)$  - функция Беллмана - минимальное значение времени торможения для процесса, начинающегося в фазовой точке  $x$ . Тогда согласно принципу Беллмана оптимальное управление находится из условия

$$\min_{|u| \leq 1} \left( \frac{\partial T}{\partial x}, u + \phi_0(x) + \varepsilon \phi(x) \right) = -1, \quad T(0, \varepsilon) = 0. \quad (3.12)$$

Здесь  $\frac{\partial T}{\partial x}$  - вектор частных производных  $T(x, \varepsilon)$  по  $x$ , векторные функции  $\phi_0, \phi : \mathbb{R}^3 \rightarrow \mathbb{R}^3$  определяются равенствами:

$$\begin{aligned} \phi_0(x) &= (a_1 x_2 x_3, a_2 x_1 x_3, a_3 x_1 x_2), \\ \phi(x) &= (\phi_1(x_1, x_2, x_3), \phi_2(x_1, x_2, x_3), \phi_3(x_1, x_2, x_3)). \end{aligned}$$

В левой части равенства (3.12) стоит скалярное произведение, минимизация по  $u$  которого приводит к оптимальному управлению вида:

$$u(x, \varepsilon) = -p / \|p\|, \quad p = \left( \frac{\partial T}{\partial x_1}, \frac{\partial T}{\partial x_2}, \frac{\partial T}{\partial x_3} \right) = \frac{\partial T}{\partial x}. \quad (3.13)$$

Подставим выражение для  $u(x, \varepsilon)$  из (3.13) в (3.12), получим:

$$\|p\| - (p, \phi_0) - \varepsilon (p, \phi) = -1, \quad T(0, \varepsilon) = 0, \quad (3.14)$$

решение которого будем искать в виде ряда по степеням  $\varepsilon$ :

$$T(x, \varepsilon) = T^0(x) + \varepsilon T^1(x) + \dots + \varepsilon^k T^k(x) + \dots$$

Введем обозначения:

$$p(x, \varepsilon) = \frac{\partial T(x, \varepsilon)}{\partial x}, \quad p^k(x) = \frac{\partial T^k}{\partial x}, \quad k = 0, 1, 2, \dots$$

тогда

$$p(x) = p^0(x) + \varepsilon p^1(x) + \dots + \varepsilon^k p^k(x) + \dots$$

Подставив выражения для  $p(x)$  в (3.14) и приравняв коэффициенты при одинаковых степенях  $\varepsilon$  в левой и правой частях равенства, получим систему уравнений для коэффициентов  $p^k$ : при  $\varepsilon = \varepsilon^0$ ,  $T^0(0) = 0$ . При  $\varepsilon^1$  имеем:

$$\begin{aligned} \|p^0\| - (p^0, \phi_0) &= 1, \\ (p^1, p^0 - \phi_0) &= (p^0, \phi(x)) = F^1(x), \quad T^1(0) = 0, \\ (p^k, p^0 - \phi_0) &= F^k(x), \quad T^k(0) = 0, \quad k = 2, 3, 4, \dots \end{aligned} \quad (3.15)$$

Легко показать, что  $T^0(x) = \|x\|$ ,  $p^0(x) = x\|x\|^{-1}$ .

Подставим найденное значение для  $p^0(x)$  во второе уравнение системы (3.15), получим линейное, неоднородное уравнение в частных производных с неизвестной функцией  $T^1(x)$ , удовлетворяющей условию  $T^1(0) = 0$ :

$$\frac{\partial T^1}{\partial x_1}(x_1\|x\|^{-1} - a_1x_2x_3) + \frac{\partial T^1}{\partial x_2}(x_2\|x\|^{-1} - a_2x_1x_3) + \frac{\partial T^1}{\partial x_3}(x_3\|x\|^{-1} - a_3x_1x_2) = F^1(x).$$

Уравнение характеристик для него задается соотношениями:

$$\frac{dx_1}{x_1\|x\|^{-1} - a_1x_2x_3} = \frac{dx_2}{x_2\|x\|^{-1} - a_2x_1x_3} = \frac{dx_3}{x_3\|x\|^{-1} - a_3x_1x_2} = \frac{dT^1}{-F^1} \quad (3.16)$$

Семейство характеристик строится однозначно, т.к. сумма квадратов коэффициентов при частных производных отлична от нуля. Обозначим  $\|x\| = l$ , тогда в результате несложных преобразований получим систему обыкновенных дифференциальных уравнений, равносильную (3.16)

$$\begin{aligned} \frac{dx_i}{dl} &= x_i l^{-1} - a_i x_k x_l, \quad i = 1, 2, 3, \quad k \neq i, \quad l \neq i, \quad l \neq k, \\ \frac{dT^1}{dl} &= -F^1(x_1, x_2, x_3). \end{aligned} \quad (3.17)$$

Рассмотрим первые три уравнения системы (3.17). Подставим найденные выражения (3.11) для  $x_i$ ,  $i = 1, 2, 3$ , в последнее уравнение системы (3.17), находим

$$T^1(x) = - \int_0^{\|x\|} F^1(x_1(l), x_2(l), x_3(l)) dl.$$

Уравнение для последующих коэффициентов  $T^k(x)$ ,  $k \geq 2$ , имеет вид:

$$(p^k, p^0 - \phi_0) = F^k(x), \quad k = 1, 2, 3, \dots \quad (3.18)$$

Функции  $F^k(x)$  в правой части уравнения в результате вычисления функций  $T^1(x), \dots, T^{k-1}(x)$  на предыдущих шагах. Аналогично случаю  $k = 1$  решение  $T^k(x)$  уравнения (3.18) перепишется в виде:

$$T^k(x) = - \int_0^{\|x\|} F^k(x_1(l), x_2(l), x_3(l)) dl.$$

Если векторная функция  $\phi(x)$  непрерывно дифференцируема по  $x$  в некоторой окрестности начала координат  $k$  раз, то

$$T(x, \varepsilon) = \|x\| + \sum_{j=1}^k \varepsilon^j T^j(x) + o(\varepsilon^{k+1}).$$

## Контрольные вопросы

1. Является ли рассматриваемая здесь система слабоуправляемой, слабовозмущённой?
2. Что называется первым интегралом системы?
3. Сколько независимых первых интегралов имеет невозмущённая система уравнений?
4. Дайте определение эллиптических функций первого рода.
5. Как находится решение системы нулевого приближения?
6. Сформулируйте принцип Беллмана.
7. Что представляет собой уравнение Беллмана в рассматриваемом случае?
8. Как применяется метод характеристик?
9. Каковы итоги численного эксперимента?
10. Почему найденное допустимое управление является оптимальным?

## § 4. Численный эксперимент

Рассмотрим систему вида

$$\begin{aligned} \dot{x}_i &= a_i x_k x_l + b_1 u_i + b_2 \varepsilon \phi_i(x_1, x_2, x_3), \\ x_i(0) &= x_i^0, \operatorname{sign} a_i = (-1)^{i+1}, \\ a_1 + a_2 + a_3 &= 0, 0 \leq \varepsilon < 1, \\ i, k, l &\in 1, 2, 3, i \neq k, k \neq l, \end{aligned} \tag{4.1}$$

где  $b_1, b_2$  - булевы переменные, которые принимают значения 0 или 1.

В качестве управления в системе (4.1) полагается  $u_i(t) = -x_i(t)l^{-1}(t)$ ,  $i = 1, 2, 3$ . Полученная система дифференциальных уравнений решается методом Рунге-Кутты 4-ого порядка. В таблице 4 представлены варианты работы программы с различными параметрами. Под  $T$  понимается промежуток времени, при котором

$$\max_{i=1,2,3} x_i(T) < 10^{-3}.$$

Функция  $\phi_i(x) = -\sum_{j=1}^3 \lambda_{ij}x_j$ , где  $\lambda_{ij}$  элементы матрицы

$$\Lambda = \begin{pmatrix} 2 & 2 & 1 \\ 2 & 3 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

На рисунках кружочком обозначено начало координат, квадратиком - начальная точка  $(x_1(0), x_2(0), x_3(0))$ .

Результаты работы программы

Пример №	$b_1$	$b_2$	$\varepsilon$	$x_1$	$x_2$	$x_3$	$a_1$	$a_2$	$a_3$	$T$	Рисунок
1	1	1	0.1	1	1	1	1	-2	1	1.43	Рис. 1
2	0	1	0.1	1	1	1	1	-2	1	105.49	Рис. 2
3	1	0	0.1	1	1	1	1	-2	1	1.728	Рис. 1
4	0	0	0.1	1	1	1	1	-2	1	—	Рис. 3
5	0	0	0.1	1	1	0	1	-2	1	—	Рис. 4
6	0	0	0.1	1	0	1	1	-2	1	—	Рис. 5
7	0	0	0.1	0	1	1	1	-2	1	—	Рис. 6
8	0	1	0.01	1	1	1	1	-2	1	846.63	Рис. 7
9	1	1	0.01	1	1	1	1	-2	1	1.69	Рис. 1

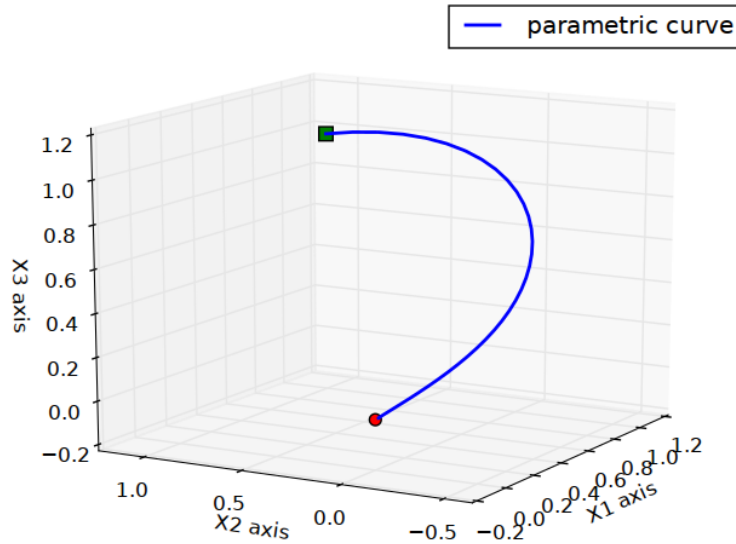


Рис. 1: Фазовая траектория системы ДУ (4.1) при наличии управления  $u$ .

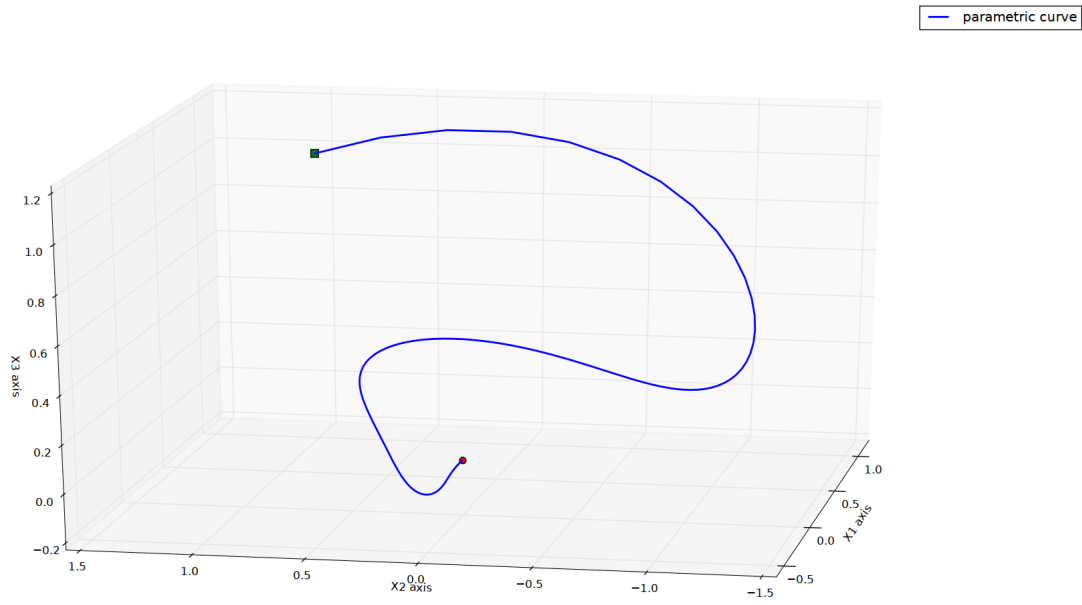


Рис. 2: Фазовая траектория системы ДУ (4.1) при наличии возмущающих моментов инерции  $\phi$  и без управления  $u$ . Значение параметра  $\varepsilon = 0.1$ .

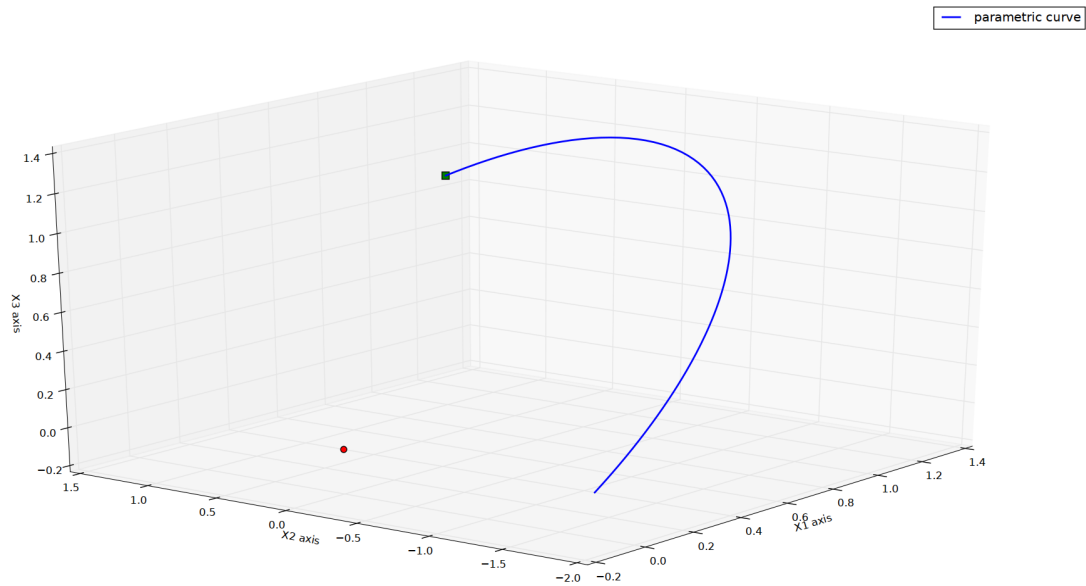


Рис. 3: Фазовая траектория системы ДУ (4.1) без управляющей функции  $u$  и возмущающих моментов инерции  $\phi$ . Вектор  $x = (1, 1, 1)$ .

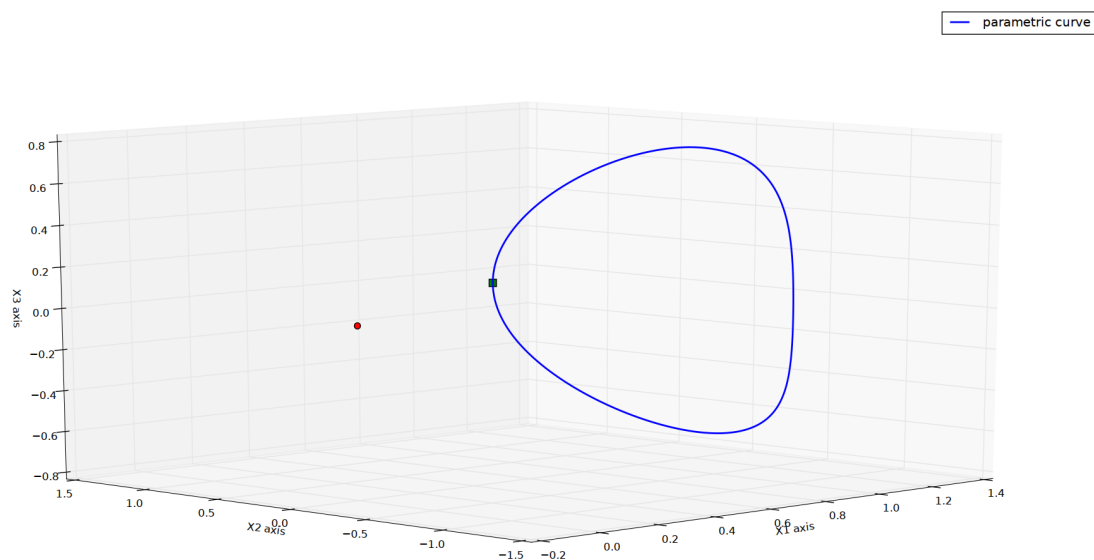


Рис. 4: Фазовая траектория системы ДУ (4.1) без управляющей функции  $u$  и возмущающих моментов инерции  $\phi$ . Вектор  $x = (1, 1, 0)$ .

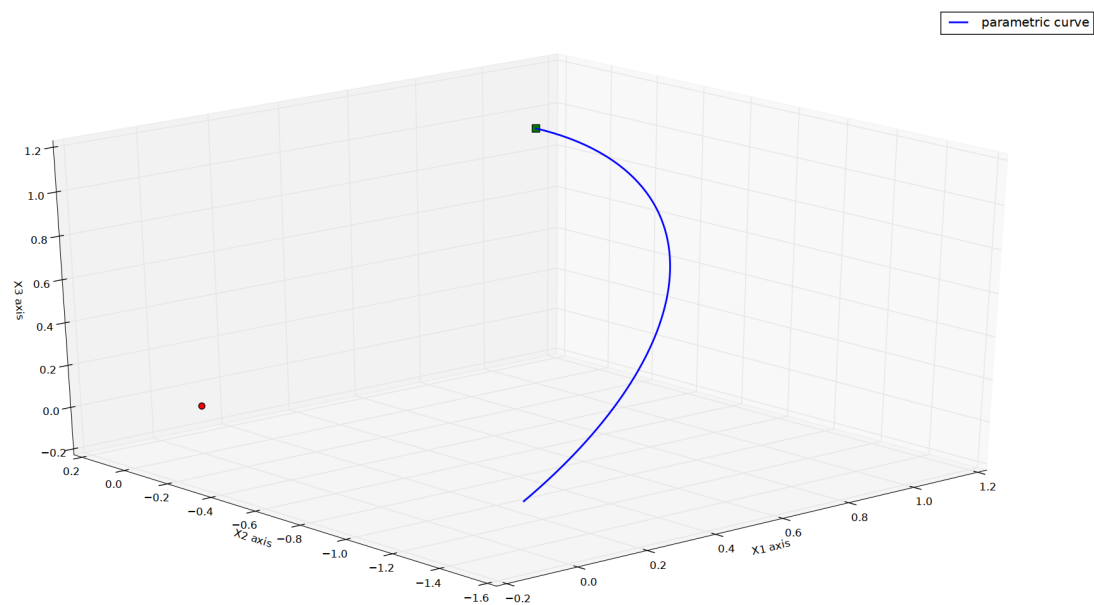


Рис. 5: Фазовая траектория системы ДУ (4.1) без управляющей функции  $u$  и возмущающих моментов инерции  $\phi$ . Вектор  $x = (1, 0, 1)$ .

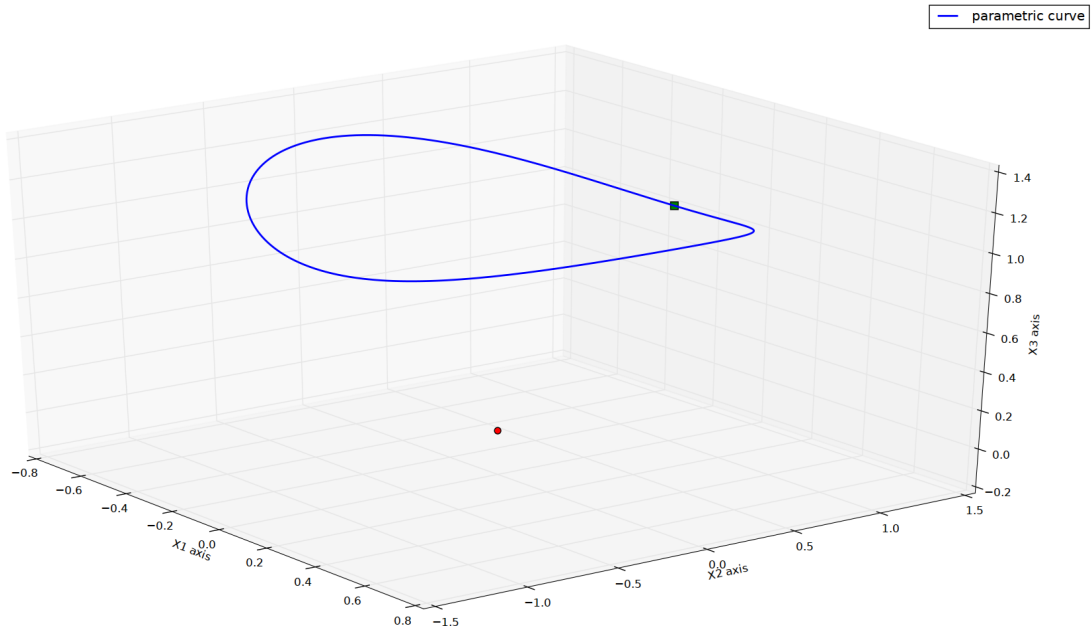


Рис. 6: Фазовая траектория системы ДУ (4.1) без управляющей функции  $u$  и возмущающих моментов инерции  $\phi$ . Вектор  $x = (0, 1, 1)$ .

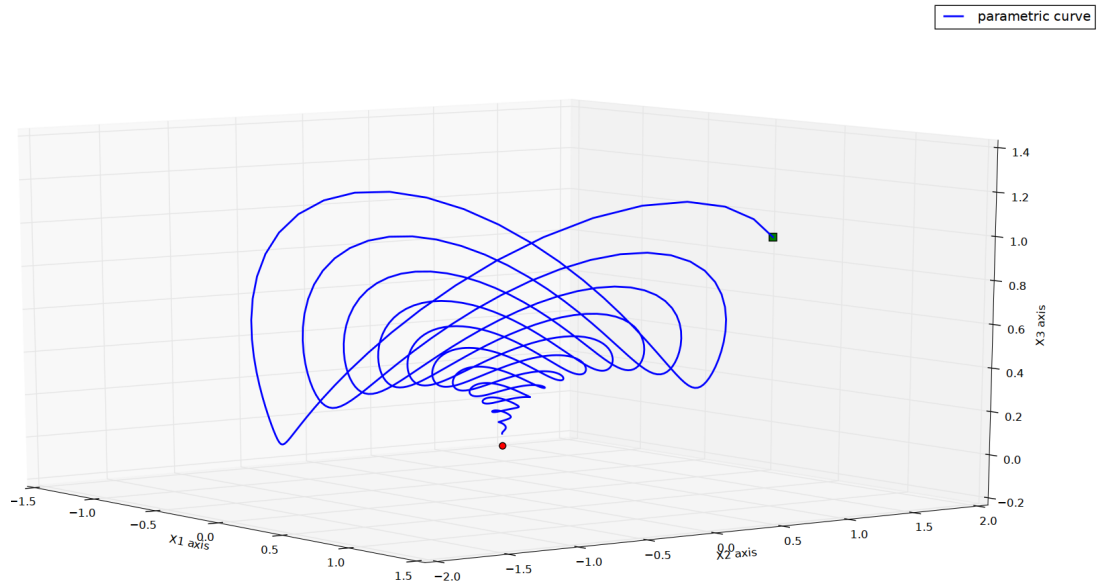


Рис. 7: Фазовая траектория системы ДУ (4.1) без управляющей функции  $u$ . Значение параметра  $\varepsilon = 0.01$ .

Приведем пример решения задачи (4.1) на языке программирования *Python*. Для написания программы использовалась версия 2.7.11 интерпретатора *Python*. Программа тестировалась под операционной системой ALT Linux.

В программе использовались следующие библиотеки: *math*, *argparse*, *matplotlib* 1.5.0, *mpl\_toolkits* 1.5.0.

Опции программы имеют вид:

```
usage: Program1.py [-h] [--disable-control] [--disable-disturbance]
                  [--points-count POINTS_COUNT] [--x1 X1] [--x2 X2] [--x3 X3]
                  [--a1 A1] [--a2 A2] [--a3 A3] [--T T]
                  [--file-with-matrix FILE_WITH_MATRIX]
```

optional arguments:

```
-h, --help            show this help message and exit
--disable-control      disable-control (default: False)
--disable-disturbance  disable-control (default: False)
--points-count POINTS_COUNT
                        start value of energy (default: 200)
--x1 X1               Initial x1 value (default: 1)
--x2 X2               Initial x2 value (default: 1)
--x3 X3               Initial x3 value (default: 1)
--a1 A1               Define a1 coefficient (default: 1)
--a2 A2               Define a2 coefficient (default: -2)
--a3 A3               Define a3 coefficient (default: 1)
--T T                 time interval (default: 10)
--file-with-matrix FILE_WITH_MATRIX
                        Path to file with Phi matrix(default: None)
```

Ниже представлен листинг кода программы.

```
#!/usr/bin/python

from math import exp, sin, cos
from math import pow as mpower
import argparse

import matplotlib.pyplot as plt
import matplotlib as mpl
from mpl_toolkits.mplot3d import Axes3D

matr = [
    [2, 2, 1],
    [2, 3, 1],
```



```
[1, 1, 1]
]
```

```
class MainSystem(object):
    """Main system of differential equations

    """
    def __init__(self, matr=matr, x01=1, x02=1, x03=1, eps=0.01,
                  b1=1, b2=1, a1=1, a2=-2, a3=1):
        self.x01 = x01
        self.x02 = x02
        self.x03 = x03

        self.matr = matr
        self.a1 = a1
        self.a2 = a2
        self.a3 = a3
        self.eps = eps
        self.b1 = b1
        self.b2 = b2

    def phi_fun(self, i, x1, x2, x3):
        i = i - 1
        elem_sum = 0
        x = [x1, x2, x3]
        for j in xrange(len(self.matr[i])):
            elem_sum += self.matr[i][j] * x[j]

        return (-1) * elem_sum

    def fun1(self, t, x1, x2, x3):
        l = (x1 ** 2 + x2 ** 2 + x3 ** 2) ** 0.5
        res = self.a1 * x2 * x3 - self.b1 * (float(x1) / l) + \
            self.b2 * self.eps * self.phi_fun(1, x1, x2, x3)
        return res

    def fun2(self, t, x1, x2, x3):
        l = (x1 ** 2 + x2 ** 2 + x3 ** 2) ** 0.5
        res = self.a2 * x1 * x3 - self.b1 * (float(x2) / l) + \
            self.b2 * self.eps * self.phi_fun(2, x1, x2, x3)
        return res

    def fun3(self, t, x1, x2, x3):
        l = (x1 ** 2 + x2 ** 2 + x3 ** 2) ** 0.5
        res = self.a3 * x1 * x2 - self.b1 * (float(x3) / l) + \
            self.b2 * self.eps * self.phi_fun(3, x1, x2, x3)
```

```

        return res

    def get_all_functions(self):
        functions = [self.fun1, self.fun2, self.fun3]
        return functions

    def get_init_values(self):
        x_init = [self.x01, self.x02, self.x03]
        return x_init

class Runge_Rule(object):
    """Runge-Kutta method

    """
    def __init__(self, variabale_count):
        self.variabale_count = variabale_count

    # integrate system of differential equations on one step
    def integrate_on_step(self, step, time, x_values, fun):
        K1 = []
        K2 = []
        K3 = []
        K4 = []

        x = x_values
        t = time

        for i in xrange(self.variabale_count):
            K1.append(fun[i](t, *x))

        t = time + step / 2.0
        x = map(lambda i: x_values[i] + step * K1[i] / 2.0,
                range(len(x_values)))
        for i in xrange(self.variabale_count):
            K2.append(fun[i](t, *x))

        t = time + step / 2.0
        x = map(lambda i: x_values[i] + step * K2[i] / 2.0,
                range(len(x_values)))
        for i in xrange(self.variabale_count):
            K3.append(fun[i](t, *x))

        t = time + step
        x = map(lambda i: x_values[i] + step * K3[i],
                range(len(x_values)))
        for i in xrange(self.variabale_count):

```

```

        K4.append(fun[i](t, *x))

    new_x = map(lambda i: x_values[i] + (step / 6.0) * \
                    (K1[i] + 2 * K2[i] + 2 * K3[i] + K4[i]),
                    range(len(x_values)))
    return new_x

def integrate(self, a, b, x_values_init, fun, nodes_count):
    results = []
    for i in xrange(self.variabale_count):
        results.append([])
    step = float(b - a) / nodes_count
    time = a

    # when x_values is changed then x_values_init is changed too
    x_values = x_values_init
    for i in xrange(len(x_values)):
        results[i].append(x_values[i])

    for i in xrange(nodes_count):
        x_values = self.integrate_on_step(step, time, x_values, fun)

        for i in xrange(len(x_values)):
            results[i].append(x_values[i])

        time += step

    return results

def main(disable_control, disable_disturbance, points_count,
        x01, x02, x03, a1, a2, a3, time_interval, file_with_matrix):
    method = Runge_Rule(3)
    if disable_control:
        b1 = 0
    else:
        b1 = 1
    if disable_disturbance:
        b2 = 0
    else:
        b2 = 1

    custom_matrix = matr
    if file_with_matrix is not None:
        custom_matrix = []
        with open(file_with_matrix, 'r') as fh:
            for line in fh:

```

```

        line = line.strip()
        if line != "":
            row = map(lambda elem: float(elem), line.split(" "))
            custom_matrix.append(row)

system = MainSystem(matr=custom_matrix, x01=x01, x02=x02,
                    x03=x03, b1=b1, b2=b2, a1=a1, a2=a2, a3=a3)

# get all right functions of system of differential equations
funcs = system.get_all_functions()
# get all initial values for system of differential equations
init_val = system.get_init_values()

N = points_count
T = time_interval
a = 0

# solve system of differential equations by Runge-Kutta method
x1, x2, x3 = method.integrate(0, T, init_val, funcs, N)

step = float(T - 0) / N
time = [a]
for i in xrange(N):
    time.append(a + (i + 1) * step)

# create 3D parametric curve
fig = plt.figure()
ax = fig.gca(projection='3d')
ax.scatter(*init_val, s=70, c='g', marker='s') # start point
ax.plot(x1, x2, x3, linewidth=2.0, label='parametric curve')
ax.scatter(0, 0, 0, s=50, c='r') # origin of coordinates

ax.set_xlabel('X1 axis')
ax.set_ylabel('X2 axis')
ax.set_zlabel('X3 axis')

ax.legend()
plt.show()

if __name__ == "__main__":
    parser = argparse.ArgumentParser()

    parser.add_argument("--disable-control", action="store_true",
                        default=False,
                        help="disable-control (default: %(default)s)")
    parser.add_argument("--disable-disturbance", action="store_true",

```

```

        default=False,
        help="disable-control (default: %(default)s)")
parser.add_argument("--points-count", type=int,
        default=200,
        help="start value of energy (default: %(default)s)")
parser.add_argument("--x1", type=float, default=1,
        help="Initial x1 value (default: %(default)s)")
parser.add_argument("--x2", type=float, default=1,
        help="Initial x2 value (default: %(default)s)")
parser.add_argument("--x3", type=float, default=1,
        help="Initial x3 value (default: %(default)s)")
parser.add_argument("--a1", type=float, default=1,
        help="Define a1 coefficient (default: %(default)s)")
parser.add_argument("--a2", type=float, default=-2,
        help="Define a2 coefficient (default: %(default)s)")
parser.add_argument("--a3", type=float, default=1,
        help="Define a3 coefficient (default: %(default)s)")
parser.add_argument("--T", type=float, default=10,
        help="time interval (default: %(default)s)")
parser.add_argument("--file-with-matrix", default=None,
        help=("Path to file with Phi matrix"
              "(default: %(default)s)"))

args = parser.parse_args()

main(args.disable_control, args.disable_disturbance,
      args.points_count, args.x1, args.x2,
      args.x3, args.a1, args.a2, args.a3,
      args.T, args.file_with_matrix)

```

## Вопросы к экзамену

1. Общая постановка задачи оптимального управления.
2. Принцип максимума Понтрягина для неавтономных систем.
3. Слабоуправляемые системы.
4. Вывод уравнений нулевого и первого приближений.
5. Анализ уравнений нулевого и первого приближений.
6. Исследование сопряжённой системы.
7. Оптимальное управление в первом приближении.
8. Алгоритм приближённого решения задачи оптимального управления для слабоуправляемых систем
9. Постановка задачи о полёте на максимальную дальность, приведение её к виду удобному для исследования.
10. Применение ранее полученного алгоритма к решению задачи о полёте на максимальную дальность.
11. Нахождение оптимального управления при заданных аэродинамических характеристиках системы.
12. Постановка задачи об успокоении твёрдого тела за минимальное время при наличии возмущения
13. Эллиптические функции первого рода (определение и свойства).
14. Оптимальное управление фазовая траектория и время успокоения для невозмущённой системы.
15. Принцип Беллмана, применение его к нахождению оптимального времени успокоения твёрдого тела при наличии возмущения..

## Список литературы

- [1] Математическая теория оптимальных процессов / А. С. Понтрягин [ и др. ]. – Москва : Наука, 1969. – 256 с.
- [2] Черноусько Ф. Л. Управление колебаниями / Ф. Л. Черноусько, Л. Д. Акуленко, Б. Н. Соколов. – Москва : Наука, 1980. – 383 с.
- [3] Ногин В. Д. Введение в оптимальное управление : учебно-методическое пособие / В. Д. Ногин. – Санкт-Петербург : Ютас, 2008. – 92 с.
- [4] Матвеев А. С. Оптимальные системы управления: Обыкновенные дифференциальные уравнения. Специальные задачи/ А. С. Матвеев, В. А. Якубович. – Санкт-Петербург : Издательство Санкт-Петербургского университета, 2003. – 540 с.
- [5] Бушуев А. Ю. Введение в оптимальное управление : электронное учебное издание / А. Ю. Бушуев. – Москва : Издательство Московского государственного технической университет им. Н. Э. Баумана, 2014. – 23 с.
- [6] Ахиезер Н. И. Элементы теории эллиптических функций / Н. И. Ахиезер. – Москва : Наука, 1970. – 303 с.

# Содержание

Введение . . . . .	3
1. Метод малого параметра в теории управления . . . . .	4
1.1. Постановка задачи оптимального управления . . . . .	4
1.2. Принцип максимума Понтрягина . . . . .	4
1.3. Задачи управления с малым параметром. Слабоуправляемые системы	6
1.4. Вывод уравнений нулевого и первого приближений . . . . .	7
1.5. Построение приближенного решения . . . . .	8
1.6. Оптимальное управление в первом приближении . . . . .	9
1.7. Алгоритм приближенного решения задачи оптимального управления для слабоуправляемых систем . . . . .	10
Контрольные вопросы . . . . .	12
2. Задача о полете на максимальную дальность . . . . .	12
2.1. Постановка задачи и преобразование системы к нормальному виду . .	12
2.2. Пакет Mathematica: полезная информация . . . . .	14
2.3. Реализация алгоритма решения задачи в пакете Mathematica . . . . .	15
2.4. Анализ полученных результатов . . . . .	19
Контрольные вопросы . . . . .	21
3. Метод малого параметра в задаче управления движением трехосного твердого тела . . . . .	21
3.1. Постановка задачи . . . . .	21
3.2. Случай невозмущенной системы . . . . .	22
3.3. Эллиптические функции I-ого рода, их применение . . . . .	23
3.4. Принцип Беллмана, его применение в случае $\varepsilon \neq 0$ . . . . .	25
Контрольные вопросы . . . . .	27
4. Численный эксперимент . . . . .	27
Вопросы к экзамену . . . . .	38
Список литературы . . . . .	39



*Учебное издание*

# **МЕТОД МАЛОГО ПАРАМЕТРА В ЗАДАЧАХ ОПТИМАЛЬНОГО УПРАВЛЕНИЯ**

Учебное пособие для вузов

Авторы:

**Кацаран** Татьяна Константиновна,  
**Кабанцова** Лариса Юрьевна