САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Кафедра МЕХАНИКИ УПРАВЛЯЕМОГО ДВИЖЕНИЯ

Левдик Вероника Владимировна

Выпускная квалификационная работа бакалавра

Оптимальное по смешанному критерию гашение малых колебаний космического аппарата

Направление 010400
Прикладная математика и информатика

Научный руководитель,
кандидат физ.-мат. наук,
доцент
Потоцкая И. Ю.

Санкт-Петербург
2018

Содержание

Введение

Данная работа посвящена вопросу об управлении движением механической системы и о способах оптимизации управления. В настоящее время актуальность задач оптимизации объясняется развитием технологий в современном мире и, соответственно, увеличением масштаба автоматизации. На практике возникает все больше и больше задач, требующих оптимизации движения, то есть достижения каких-либо целевых состояний системы при соблюдении различных наложенных ограничений. Подобные задачи возникают в областях науки, таких как математика, физика, химия, экономика, экология, биология и т. д. Одним из основных аспектов формализации задач является корректное описание математических моделей. Любое исследование зачастую начинается с построения упрощенных моделей, позволяющих описать и воспроизвести поведение системы в целом, либо, наоборот, каких-то её локализованных частей. Для каждой практической задачи существуют влияющие факторы, которые нужно учитывать при построении усложненных и более конкретизированных математических моделей на основе уже изученных.

Одной из основных развивающихся научных и промышленных отраслей сегодня является космическая динамика. В этой области автоматизированное управление значительно преобладает над ручным управлением ввиду сложности, а иногда невозможности, поддержания связи человека с космическим аппаратом. В связи с этим аналитическое изучение соответствующих математических моделей является главным инструментом для решения возникающих задач.

В представленной работе рассматривается задача гашения малых колебаний спутника, которые значительно снижают качество работы космического аппарата (далее — КА). Управление спутниками осуществляется с помощью реактивных двигателей, которые работают на дорогостоящем топливе, что влечет за собой вопрос о минимизации затрат. Новизна исследования состоит в том, что производится построение оптимального управления, которое обуславливает быстродействие и минимизацию расхода топлива одновременно. Нужно отметить, что решение указанной задачи

по смешанному критерию не дает и не может давать результатов относительно каждого критерия, рассматриваемого в отдельности, лучших, чем если бы эти критерии рассматривались автономно. Суть работы состоит в том, чтобы построить наиболее удовлетворяющее реальным практическим запросам управление, которое позволит улучшить качество работы космического аппарата.

Цель работы

Одним из аспектов, отвечающий за качество функционирования КА на орбите, является его положение в состоянии относительного равновесия. На практике движение КА может сопровождаться возмущениями различного вида, которые могут пагубно влиять на качество функционирования аппарата и выводить его из состояния стационарного вращения. В настоящей работе ставится задача построения оптимального управления механической системой с целью гашения малых колебаний. Движение системы смоделировано с помощью задачи Лагранжа, которая описывается системой линейных обыкновенных дифференциальных уравнений с постоянными коэффициентами. Подразумевается, что управление осуществляется с помощью реактивных двигателей. Критерием оптимальности является одновременное уменьшение количества расходуемого топлива и продолжительности действия управления.

Цель работы:

- вывести аналитический алгоритм для описания функции оптимального управления, обеспечивающего гашение малых колебаний КА;
- осуществить программную реализацию найденного алгоритма и получить графики, демострирующие корректную работу метода;
- на основе численной реализации произвести анализ результатов, построить графики, отображающие зависимость частей функционала качества от характерного параметра.

Обзор литературы

При написании данной работы были использованы научная и учебно-методическая литература конца XX – начала XXI века. Основные аспекты теории оптимального управления приведены в многочисленных источниках. В частности, в работе Мейера А. [1], систематически изложены методы математического моделирования динамических систем и способы их управления, основы теории оптимальных систем, методы анализа устойчивости. Также в источнике [1] демонстрируются классические начала вариационного исчисления, принцип Максимума Понтрягина и метод динамического программирования Беллмана. Одним из фундаментальных примеров научной литературы, посвященной теории управления является книга американских ученых Атанса М. и Фалба П. [2], в которой изложены методы проектирования систем, оптимальных по отношению к различным критериям качества. Другими примерами источников по указанной тематике являются [3]–[7].

Во второй половине XX века ученые пришли к выводу, что изучение задач с критерием оптимальности по расходу топлива вызывает большие трудности в теоретическом и практическом аспектах, в связи с чем акцент рассмотрения подобных задач перешёл на задачи с квадратичными функционалами. Изучение задач с негладким функционалом возобновилось, например, в работе Бабаджанянца Л. К., Потоцкой И. Ю. [8]. В указанном источнике описывается метод оптимизации по расходу топлива для задач управления линейной системой с кусочно-постоянным видом функции управления.

За основу настоящей работы взята статья [9], где описывается метод построения оптимального управления по смешанному критерию. С помощью указанного метода в данной работе составлен алгоритм для аналитического построения оптимального управления в задаче Лагранжа на примере вращательного движения спутника вокруг неподвижной точки в однородном поле тяжести. Теоретический материал для создания математической модели взят из работ [10]–[13].

Программная реализация полученного метода осуществлена с помо-

щью языка программирования Python, теоретические основы и особенности которого широко описаны в работе Лутца М. [14]. В программе использованы численные методы Брента Р. [15].

§1. Постановка задачи оптимального управления КА

1. Вращательное движение в однородном поле тяжести

Пусть твердое тело (космическая станция) совершает вращательное движение в гравитационном поле. Механика данного процесса подробно изложена в [12]. Приведём основные выкладки, неоходимые для описания модели.

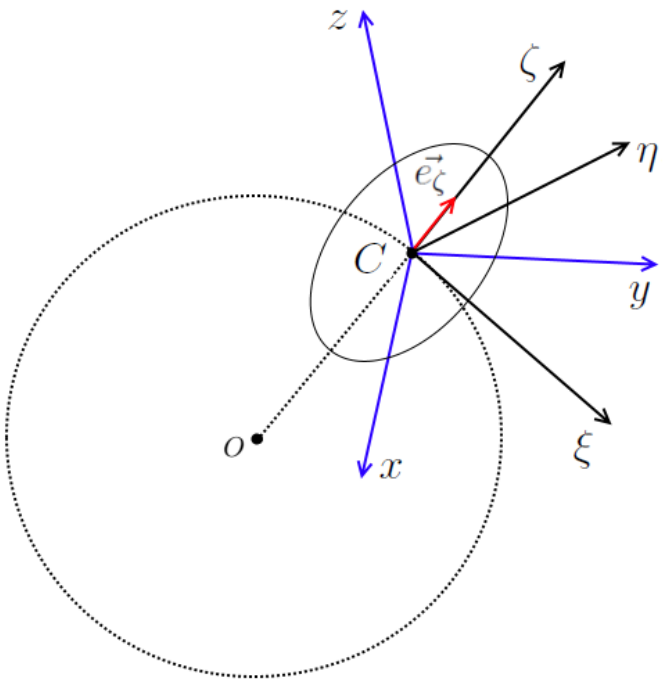

Рис. 1: Движение спутника

Движение происходит вокруг неподвижной точки O под действием силы тяжести (Рис. 1). $\vec{F}_т = m\vec{g}$, $m = const$ — масса тела, $\vec{g}$ — гравитационная постоянная, $\vec{r_c}$ — радиус-вектор центра масс C относительно указанной неподвижной точки. Рассмотрим две системы координат с началом в точке C: двигающуюся поступательно вместе с центром масс систему $(\xi, \eta, \zeta,)$ и жестко связанную с телом (связную) систему (x, y, z), оси которой совпадают с главными центральными осями инерции тела. $\vec{e_\zeta}$ — орт оси $O\zeta$, а его направляющие косинусы в связной системе координат — (α, β, γ). Тензор инрции J в главных осях инерции диагонален: $J = \begin{pmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{pmatrix}$. Момент импульса $\vec{L} = J\vec{\omega}$, где $\vec{\omega} = (\omega_x, \omega_y, \omega_z)$ — угловая скорость вращения твердого тела около точки C в связной системе координат, $C = (x_c, y_c, z_c)$. В указанных обозначениях запишем *динамические уравнения Эйлера* и *урав-*

нения Пуассона, описанные в [12]:

$$\begin{cases} J_x\dot{\omega}_x + (J_z - J_y)\omega_z\omega_y = mg(\beta z_c - \gamma y_c), \\ J_y\dot{\omega}_y + (J_x - J_z)\omega_x\omega_z = mg(\gamma x_c - \alpha z_c), \\ J_z\dot{\omega}_z + (J_y - J_x)\omega_x\omega_y = mg(\alpha y_c - \beta x_c). \end{cases} \tag{1}$$

$$\begin{cases} \dot{\alpha} = \beta\omega_z - \gamma\omega_y, \\ \dot{\beta} = \gamma\omega_x - \alpha\omega_z, \\ \dot{\gamma} = \alpha\omega_y - \beta\omega_x. \end{cases} \tag{2}$$

Известно 3 интеграла системы (1), (2) и доказано, что если найти четвертый интеграл, то задача может быть решена в конечном виде.

Теорема Ковалевской. *Четвертый алгебраический интеграл системы (1), (2) существует только в следующих случаях:*

- *Случай Эйлера.* $x_c = 0$, $y_c = 0$, $z_c = 0$;
- *Случай Лагранжа.* $J_x = J_z$, $x_c = 0$, $z_c = 0$;
- *Случай Ковалевской.* $J_x = J_y = 2J_z$, $z_c = 0$.

2. Описание математической модели

Для приведения возмущенного движения в состояние стационарного вращения применяется управление с помощью реактивных двигателей. В [10] описано построение математической модели, которое вкратце приведено ниже.

Рассмотрим случай Лагранжа и положим $J_x = J_z = J$. Доказано, что имеется устойчивость по Ляпунову при $J_y > J$, т. е. это условие необходимо для обеспечения устойчивости стационарного вращения. Уравнения Эйлера вращения спутника около центра масс имеют вид (1). Полагая, что гравитационные моменты значительно малы по сравнению с управляющи-

ми моментами, перепишем уравнения Эйлера в следующем виде:

$$\begin{cases} \dot{\omega}_x = \mu\,\omega_z + u_1, \\ \dot{\omega}_z = -\mu\,\omega_x + u_2, \\ \dot{\omega}_y = u_3. \end{cases} \tag{3}$$

Здесь $\mu = \dfrac{\omega_y(J_y - J)}{J} = const > 0$, управления u_1 и u_2 равны отношению соответствующего реактивного момента к моменту инерции J, u_3 — отношение реактивного момента относительно оси O_y к J_y. Полагается, что $u_3 = 0$. Обозначив $x_1 = \omega_x$, $x_2 = \omega_z$, получим:

$$\begin{cases} \dot{x}_1 = \mu x_2 + u_1, \\ \dot{x}_2 = -\mu x_1 + u_2. \end{cases} \tag{4}$$

3. Постановка задачи гашения колебаний

Согласно предыдущему пункту, движение КА описывается системой из двух линейных дифференциальных уравнений первого порядка:

$$\dot{x} = Ax + U, \tag{5}$$

$U(t) \in \mathbf{R}^2$, $x(t) \in \mathbf{R}^2$, $U = (u_1,\, u_2)^{\mathrm{T}}$ — вектор-функция управления, $x = (x_1,\, x_2)^{\mathrm{T}}$ — вектор фазовых переменных, $A = \begin{pmatrix} 0 & \mu \\ -\mu & 0 \end{pmatrix}$. Можно заметить, что собственными числами матрицы A являются чисто мнимые значения $\pm i\mu$. Это говорит о том, что без воздействия управления система совершает периодические, малые (невозрастающие) колебания частоты μ, гашение которых является целью работы управления. Кроме того, пусть заданы начальные условия:

$$x(0) = x_0. \tag{6}$$

По окончании работы двигателей вращение спутника должно стабилизироваться, т.е. фазовые переменные должны обнулиться, следовательно, задача гашения колебаний системы равносильна граничному условию

$$x(T) = 0. \tag{7}$$

Здесь T — *незаданный* момент окончания действия двигателей.

Управление представляется в виде кусочно-постоянной функции следующим образом (Рис. 2):

$$u_k = h_k \sum_{i=1}^{2r_k} (-1)^{i+1} H(t - t_i^k) + h_k \sum_{i=1}^{2q_k} (-1)^i H(t - \widetilde{t}_i^k), \quad k = 1, 2, \qquad (8)$$

где $H(t)$ — функция Хевисайда единичного скачка. Каждый компонент управления состоит из чередующихся r_k положительных и q_k отрицательных ступеней высоты h_k. Указанные параметры считаются заданными.

Здесь t_i^k, $\widetilde{t}_i^k$ — моменты переключения верхних и нижних ступеней соответственно. Полагается, что каждый компонент управления начитается с верхней ступени и заканчивается нижней и их количество одинаково. Пусть t_0 - момент начала действия управления, l — номер компонента управления, такой что $t_1^l = t_0$, m — такой, что $\widetilde{t}_{2q_m}^m = T$, $m \neq l$.

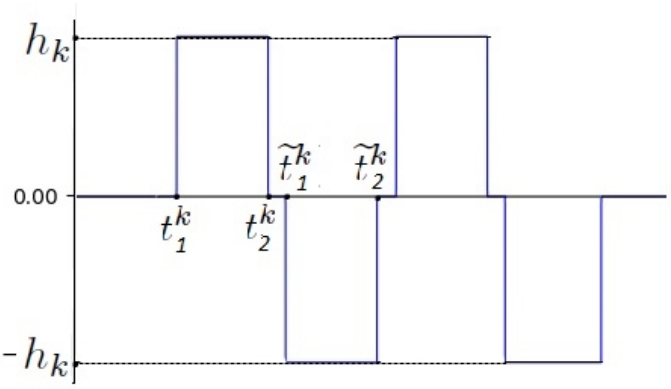

Рис. 2: Структура управления

Допустимым считается управление вида (8), при котором для системы (5), (6) выполняется условие (7).

Критерий качества рассматривается в виде функционала

$$J = \alpha \int_{t_0}^{T} dt + (1 - \alpha) \int_{t_0}^{T} |u_k(t)|\, dt. \qquad (9)$$

Первое слагаемое в (9) характеризует быстродействие, второе отвечает за расход топлива. Параметр $\alpha \in [0, 1]$.

В указанных обозначениях задача сводится к следующему. При заданных параметрах x_0, μ, r_k, q_k, h_k, α, l, m необходимо найти точки переключения t_i^k, $\widetilde{t}_i^k$ допустимого управления, доставляющие минимум функционалу (9).

§2. Алгоритм построения оптимального управления

ШАГ 1. Произведем линейную замену $x = By$, где B — неособая, постоянная, комплексная, (2×2)-матрица, $y = (y_1, y_2)^{\mathrm{T}}$ — вектор двух комплексных переменных. Подбираем B так, чтобы $B^{-1}AB = Y$, причем $Y = \begin{pmatrix} i\mu & 0 \\ 0 & -i\mu \end{pmatrix}$. Тогда система (5), (6) примет вид

$$\dot{y} = Y\,y + V, \quad y(0) = y_0, \tag{10}$$

$$C = B^{-1} = \begin{pmatrix} 1 & -i \\ 1 & i \end{pmatrix} = \left\{c_{ij}\right\}_{i=1,2}^{j=1,2},$$
$$V = CU = \begin{pmatrix} u_1 - iu_2 \\ u_1 + iu_2 \end{pmatrix},$$

с начальным условием

$$y_0 = Cx_0 = \begin{pmatrix} y_{10} \\ y_{20} \end{pmatrix} = \begin{pmatrix} x_{10} - ix_{20} \\ x_{10} + ix_{20} \end{pmatrix} \tag{11}$$

и граничным условием

$$y(T) = 0. \tag{12}$$

Таким образом, полученная система (10) состоит из двух автономных дифференциальных уравнений первого порядка. Запишем для них решение в форме Коши:

$$\begin{cases} y_1 = y_{10}\,e^{i\mu t} + e^{i\mu t}\displaystyle\int_0^t v_1(\tau)\,e^{-i\mu\tau}d\tau, \\ y_2 = y_{20}\,e^{-i\mu t} + e^{-i\mu t}\displaystyle\int_0^t v_2(\tau)\,e^{i\mu\tau}d\tau, \end{cases}$$

$$\begin{cases} y_1 = (y_{10} + \displaystyle\int_0^t v_1(\tau)(\cos\mu\tau - i\sin\mu\tau)d\tau)(\cos\mu t + i\sin\mu t), \\ y_2 = (y_{20} + \displaystyle\int_0^t v_2(\tau)(\cos\mu\tau + i\sin\mu\tau)d\tau)(\cos\mu t - i\sin\mu t), \end{cases} \tag{13}$$

ШАГ 2. Представим уравнения (13) в зависимости от точек переключения, подставив функцию управления (8) и сосчитав интеграл:

$$\begin{cases} y_1 = (y_{10} - \dfrac{i}{\mu}\displaystyle\sum_{k=1}^{2} c_{1k}h_k[F_1^k - iF_2^k])\cos\mu t + \\ \quad + (iy_{10} + \dfrac{1}{\mu}\displaystyle\sum_{k=1}^{2} c_{1k}h_k[F_1^k - iF_2^k])\sin\mu t + \dfrac{i}{\mu}\displaystyle\sum_{k=1}^{2} c_{1k}u_k, \\ y_2 = (y_{20} + \dfrac{i}{\mu}\displaystyle\sum_{k=1}^{2} c_{1k}h_k[F_1^k + iF_2^k])\cos\mu t + \\ \quad + (-iy_{20} + \dfrac{1}{\mu}\displaystyle\sum_{k=1}^{2} c_{1k}h_k[F_1^k + iF_2^k])\sin\mu t - \dfrac{i}{\mu}\displaystyle\sum_{k=1}^{2} c_{1k}u_k, \end{cases} \tag{14}$$

$$F_1^k = \sum_{i=1}^{2r_k}(-1)^{i+1}H(t-t_i^k)\cos\mu t_i^k + \sum_{i=1}^{2q_k}(-1)^{i}H(t-\widetilde{t}_i^k)\cos\mu\widetilde{t}_i^k,$$
$$F_2^k = \sum_{i=1}^{2r_k}(-1)^{i+1}H(t-t_i^k)\sin\mu t_i^k + \sum_{i=1}^{2q_k}(-1)^{i}H(t-\widetilde{t}_i^k)\sin\mu\widetilde{t}_i^k.$$

В соответствии с задачей, система (14) должна удовлетворять граничному условию (12). Положим $t = T$. Функция Хевисайда во всех слагаемых примет значение, равное единице, т. к. $T \geq t_i^k$, $T \geq \widetilde{t}_i^k$, $\forall i, k$. Слагаемые, содержащие компоненты управления обнулятся, т. к. к моменту T действие управлений закончится. Далее, приравняв к нулю коэффициенты при вещественных и мнимых частях уравнений (14) — условие равенства нулю комплексного числа, — получим две пары линейно зависимых условий. Выберем одну пару и тогда граничные условия будут равносильны следующим уравнениям:

$$\begin{cases} K_1 = y'_{10} + \dfrac{1}{\mu}\displaystyle\sum_{k=1}^{2} h_k[c^*_{1k}F_1^k - c'_{1k}F_2^k] = 0, \\ K_2 = y^*_{10} - \dfrac{1}{\mu}\displaystyle\sum_{k=1}^{2} h_k[c'_{1k}F_1^k + c^*_{1k}F_2^k] = 0, \end{cases} \tag{15}$$

где переменные y_{10}, c_{1k} разделены на вещественные и мнимые части: $y_{10} = y'_{10} + iy^*_{10}$, $c_{1k} = c'_{1k} + ic^*_{1k}$.

Аналогично, с учетом структуры управления (8) перепишем функционал качества (9) в зависимости от точек переключения:

$$J = \alpha(T - t_0) + (1-\alpha)\sum_{k=1}^{2} h_k\Big[\sum_{i=1}^{2r_k}(-1)^i\, t_i^k + \sum_{i=1}^{2q_k}(-1)^i\, \widetilde{t}_i^k\Big]. \qquad (16)$$

Таким образом, задача свелась к минимизации функционала (16) при выполнении условий (15).

ШАГ 3. Перейдем к безусловной минимизации, вводя множители Лагранжа λ_1, λ_2.

$$R = J + \lambda_1 K_1 + \lambda_2 K_2, \qquad (17)$$

$R = R\,(t_0,\, T,\, t_i^k,\, \widetilde{t}_i^k)$. *Необходимым условием* экстремума функционала R является равенство нулю всех частных производных. Рассмотрим два уравнения:

$$\begin{cases} \dfrac{\partial R}{\partial t_i^k} = 0 \Rightarrow a_k \cos \mu t_i^k + b_k \sin \mu t_i^k = -1 + \alpha, \\ \dfrac{\partial R}{\partial \widetilde{t}_i^k} = 0 \Rightarrow a_k \cos \mu \widetilde{t}_i^k + b_k \sin \mu \widetilde{t}_i^k = 1 - \alpha, \end{cases} \qquad (18)$$

$$a_k = \lambda_1 c'_{1k} + \lambda_2 c^*_{1k}, \quad b_k = \lambda_1 c^*_{1k} - \lambda_2 c'_{1k}. \qquad (19)$$

Из (18) получаем зависимость точек переключения для $j = \overline{1,\, r_k}$, $k = 1, 2$:

$$\begin{aligned} t_{2j-1}^k &= t_1^k + \frac{2\pi(j-1)}{\mu}, & t_{2j}^k &= t_2^k + \frac{2\pi(j-1)}{\mu}, \\ \widetilde{t}_{2j-1}^k &= t_1^k + \frac{2\pi(j-1)}{\mu} + \frac{\pi}{\mu}, & \widetilde{t}_{2j}^k &= t_2^k + \frac{2\pi(j-1)}{\mu} + \frac{\pi}{\mu}. \end{aligned} \qquad (20)$$

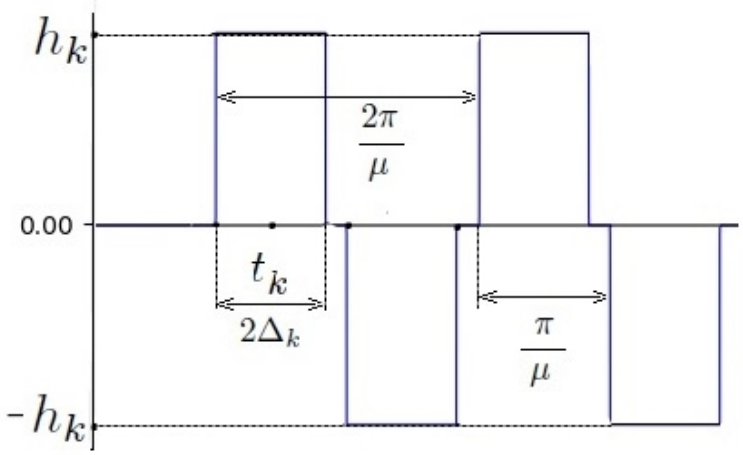

Рис. 3: Периодичность точек переключения

Из полученных соотношений видно, что ступени периодичны и имеют одинаковую ширину (Рис. 3). Расстояние между двумя соседними точками включения верхних ступеней равно $\dfrac{2\pi}{\mu}$, тогда как расстояние между двумя соседними точками включения верхней и нижней ступеней равно $\dfrac{\pi}{\mu}$ для каждого компонента управления. Аналогично выполняются соотношения между точками выключения ступеней. В соответствии с этим, мы можем ограничиться рассмотрением лишь точек включения и выключения первой ступени каждого компонента управления, т. к. все остальные выражаются через них, согласно формуле (20). Отдельному рассмотрению подлежат первая ступень l-го компонента управления и последняя ступень m-го компонента, потому что их длины могут быть меньше величины длин остальных ступенек соответствующего компонента управления.

ШАГ 4. Введем новые обозначения. Пусть t_k, t^*, t_T — средние моменты первой ступени k-го, первой ступени l-го и последней ступени m-го компонентов управления соответственно, $2\Delta_k$, $2\Delta_0$, $2\Delta_T$ — значения ширин соответствующих ступеней (Рис. 3). Тогда:

$$\begin{aligned} &t_1^k = t_k - \Delta_k, && t_2^k = t_k + \Delta_k, \quad k = 1, 2, \\ &t_0 = t^* - \Delta_0, && t_2^l = t^* + \Delta_0, \\ &\widetilde{t}^m_{q_m - 1} = t_T - \Delta_T, && T = t_T + \Delta_T. \end{aligned} \tag{21}$$

В соответствии с этими обозначениями, перепишем граничные условия (15) и функционал качества (16).

$$\begin{aligned} F_1^l &= 2\sin\mu t^* \sin\mu\Delta_0 + 2(r_l + q_l - 1)\sin\mu t_l \sin\mu\Delta_l, \\ F_2^l &= -2\cos\mu t^* \sin\mu\Delta_0 - 2(r_l + q_l - 1)\cos\mu t_l \sin\mu\Delta_l, \\ F_1^m &= -2\sin\mu t_T \sin\mu\Delta_T + 2(r_m + q_m - 1)\sin\mu t_m \sin\mu\Delta_m, \\ F_2^m &= 2\cos\mu t_T \sin\mu\Delta_T - 2(r_m + q_m - 1)\cos\mu t_m \sin\mu\Delta_m. \end{aligned}$$

$\Rightarrow$

$$\begin{cases} L_1 = y'_{10} + \dfrac{2}{\mu}\Big[\sum\limits_{k=1}^{2} S_k(c^*_{1k}\sin\mu t_k + c'_{1k}\cos\mu t_k)\sin\mu\Delta_k + \\ \qquad + h_l(c^*_{1l}\sin\mu t^* + c'_{1l}\cos\mu t^*)\sin\mu\Delta_0 - \\ \qquad - h_m(c^*_{1m}\sin\mu t_T + c'_{1m}\cos\mu t_T)\sin\mu\Delta_T\Big], \\ L_2 = y^*_{10} - \dfrac{2}{\mu}\Big[\sum\limits_{k=1}^{2} S_k(c'_{1k}\sin\mu t_k - c^*_{1k}\cos\mu t_k)\sin\mu\Delta_k + \\ \qquad + h_l(c'_{1l}\sin\mu t^* - c^*_{1l}\cos\mu t^*)\sin\mu\Delta_0 - \\ \qquad - h_m(c'_{1m}\sin\mu t_T - c^*_{1m}\cos\mu t_T)\sin\mu\Delta_T\Big]. \end{cases} \tag{22}$$

$$M = \alpha(T - t_0) + 2(1-\alpha)\Big[\sum_{k=1}^{2} S_k\Delta_k + h_l\Delta_0 + h_m\Delta_T\Big]. \tag{23}$$

$$S_k = (r_k + q_k - 1)h_k.$$

ШАГ 5. Рассмотрим первое слагаемое функционала (23).

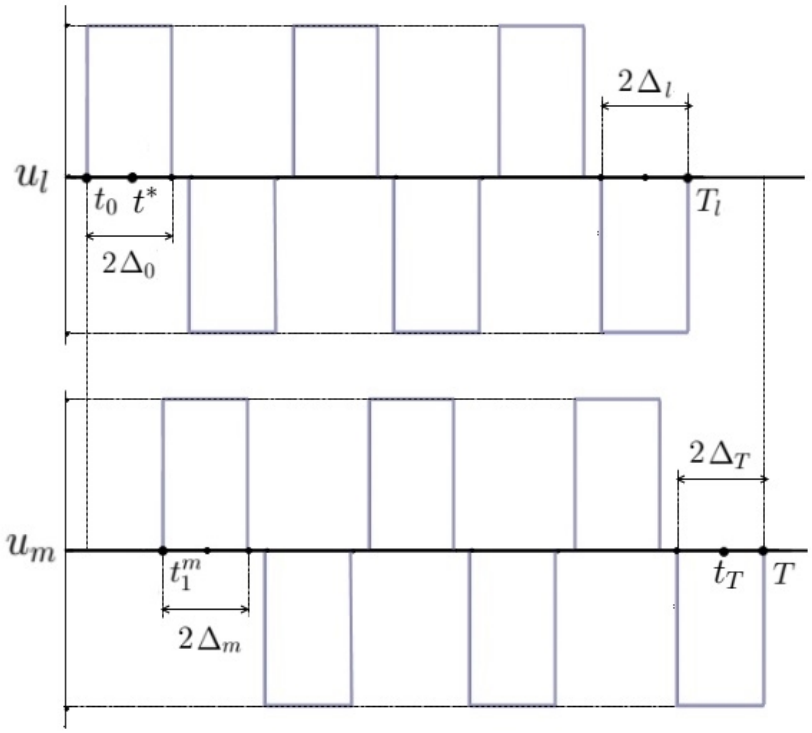

Рис. 4: Промежуток действия управления

Согласно выражениям (20), сохраняется периодичность средних моментов ступеней для всех компонентов управления, следовательно

$$t^* = t_l, \quad t_T = t_m + \frac{(r_m + q_m - 1)\pi}{\mu}.$$

Пусть T_l — момент окончания действия управления u_l, т. е. $T_l = \widetilde{t}^l_{2q_l}$, ΔT_l, ΔT_m — время действия соответствующих компонентов управления. Тогда (Рис. 4):

$$\Delta T_l = T_l - t_0, \quad \Delta T_m = T - t_1^m, \tag{24}$$

$$\Delta T_l = \Delta_l + \Delta_0 + \frac{(r_l + q_l - 1)\pi}{\mu}, \quad \Delta T_m = \Delta_m + \Delta_T + \frac{(r_m + q_m - 1)\pi}{\mu}. \tag{25}$$

Складывая уравнения (24), мы получим

$$\Delta T_l + \Delta T_m = (T - t_0) + (T_l - t_1^m),$$

$$T - t_0 = (\Delta T_l + \Delta T_m) - (T_l - t_1^m). \tag{26}$$

Минимизация первого слагаемого функционала (23) есть минимизация (26). В соответствии с периодичностью ступеней, полагается, что второе слагаемое (26) фиксировано, следовательно, при минимизации им можно пренебречь.

$$\Delta T_l + \Delta T_m = \Delta_l + \Delta_m + \Delta_0 + \Delta_T + \frac{(r_l + q_l - 1)\pi}{\mu} + \frac{(r_m + q_m - 1)\pi}{\mu}. \tag{27}$$

Т. к. константы не влияют на нахождение точек экстремума, то мы можем их опустить. Тогда, в соответствии с (27), функционал M примет следующий вид:

$$M = \alpha(\Delta_l + \Delta_m + \Delta_0 + \Delta_T) + 2(1 - \alpha)\left[\sum_{k=1}^{2} S_k\Delta_k + h_l\Delta_0 + h_m\Delta_T\right]. \tag{28}$$

ШАГ 6. Таким образом, задача состоит в том, чтобы минимизировать функционал (28) при выполнении условий (22). Перейдем к безусловной минимизации функционала $G = M + \lambda_1 L_1 + \lambda_2 L_2$.

$$G = M + \lambda_1 y'_{10} + \lambda_2 y^*_{10} + \frac{2}{\mu}\Bigg[\sum_{k=1}^{2} S_k(b_k \sin\mu t_k - a_k \cos\mu t_k)\sin\mu\Delta_k + \\ + h_l(b_l \sin\mu t^* - a_l \cos\mu t^*)\sin\mu\Delta_0 - h_m(b_m \sin\mu t_T - a_m \cos\mu t_T)\sin\mu\Delta_T\Bigg]. \tag{29}$$

Приравняем нулю все частные производные функционала G:

$$\begin{aligned}
&\frac{\partial G}{\partial t_k} = 0 \Rightarrow (b_k \cos\mu t_k - a_k \sin\mu t_k)\sin\mu\Delta_k = 0,\\
&\frac{\partial G}{\partial \Delta_k} = 0 \Rightarrow (b_k \sin\mu t_k + a_k \cos\mu t_k)\cos\mu\Delta_k = -\frac{\alpha}{2S_k} - 1 + \alpha,\\
&\frac{\partial G}{\partial t^*} = 0 \Rightarrow (b_l \cos\mu t^* - a_l \sin\mu t^*)\sin\mu\Delta_0 = 0,\\
&\frac{\partial G}{\partial \Delta_0} = 0 \Rightarrow (b_l \sin\mu t^* + a_l \cos\mu t^*)\cos\mu\Delta_0 = -\frac{\alpha}{2h_l} - 1 + \alpha,\\
&\frac{\partial G}{\partial t_T} = 0 \Rightarrow (b_m \cos\mu t_T - a_m \sin\mu t_T)\sin\mu\Delta_T = 0,\\
&\frac{\partial G}{\partial \Delta_T} = 0 \Rightarrow (b_m \sin\mu t_T + a_m \cos\mu t_T)\cos\mu\Delta_T = \frac{\alpha}{2h_m} + 1 - \alpha.
\end{aligned} \tag{30}$$

I. Рассмотрим первые два уравнения системы (30).

1). $\sin\mu\Delta_k = 0 \Rightarrow \Delta_k = 0$, что означает отстутствие управления (этот случай не рассматривается), либо $\Delta_k = \dfrac{\pi}{\mu}$, что тоже невозможно, т. к. должно выполняться соотношение $\Delta_k \le \dfrac{\pi}{2\mu}$ (из (20)). Следовательно, $\sin\mu\Delta_k \neq 0$.

2). $\sin\mu\Delta_k \neq 0 \Rightarrow b_k \cos\mu t_k - a_k \sin\mu t_k = 0$. Пусть $\cos\mu t_k = \alpha_k$, тогда получим:

$$\sin\mu t_k = \frac{\left(-\dfrac{\alpha}{2S_k} - 1 + \alpha\right)b_k}{\alpha_k(a_k^2 + b_k^2)}, \quad \cos\mu t_k = \frac{\left(-\dfrac{\alpha}{2S_k} - 1 + \alpha\right)a_k}{\alpha_k(a_k^2 + b_k^2)}. \tag{31}$$

II. Рассмотрим вторые два уравнения системы (30). Пусть $\cos\mu t^* = \alpha_0$,

тогда, аналогично пункту I, получим:

$$\sin\mu t^* = \frac{\left(-\dfrac{\alpha}{2h_l} - 1 + \alpha\right) b_l}{\alpha_0(a_l^2 + b_l^2)}, \quad \cos\mu t^* = \frac{\left(-\dfrac{\alpha}{2h_l} - 1 + \alpha\right) a_l}{\alpha_0(a_l^2 + b_l^2)}. \tag{32}$$

III. Рассмотрим пследние два уравнения системы (30). Пусть $\cos\mu t_T = \alpha_T$, тогда, аналогично пункту I, получим:

$$\sin\mu t_T = \frac{\left(\dfrac{\alpha}{2h_m} + 1 - \alpha\right) b_m}{\alpha_T(a_m^2 + b_m^2)}, \quad \cos\mu t_T = \frac{\left(\dfrac{\alpha}{2h_m} + 1 - \alpha\right) a_m}{\alpha_T(a_m^2 + b_m^2)}. \tag{33}$$

Далее рассмотрим уравнение $\lambda_2 L_1 - \lambda_1 L_2 = 0$ (из (22)) с целью выразить λ_2 через λ_1 и тем самым уменьшить количество неизвестных.

$$0 = \lambda_2 y'_{10} - \lambda_1 y^*_{10} + \frac{2}{\mu}\Bigg[\sum_{k=1}^{2} S_k(a_k \sin\mu t_k - b_k \cos\mu t_k)\sin\mu\Delta_k + \\ + h_l(a_l \sin\mu t^* - b_l \cos\mu t^*)\sin\mu\Delta_0 - h_m(a_m \sin\mu t_T - b_m \cos\mu t_T)\sin\mu\Delta_T\Bigg]. \tag{34}$$

Согласно (31)–(33), каждое слагаемое в квадратных скобках выражения (34) обнулится, следовательно

$$\lambda_2 = \frac{y^*_{10}}{y'_{10}}\lambda_1. \tag{35}$$

ШАГ 7. Выразим все оставшиеся переменные через λ_1. Положим $A_k = c'_{1k}y'_{10} + c^*_{1k}y^*_{10}$, $B_k = c^*_{1k}y'_{10} - c'_{1k}y^*_{10}$, $M_k = |y_{10}||c_{1k}|$, $|y_{10}| = \sqrt{y'^2_{10} + y^{*2}_{10}}$, $|c_{1k}| = \sqrt{c'^2_{1k} + c^{*2}_{1k}}$, $\Rightarrow\ M_k^2 = A_k^2 + B_k^2 = |y_{10}|^2|c_{1k}|^2$. Тогда, из (19), (35):

$$a_k = \frac{\lambda_1 A_k}{y'_{10}}, \quad b_k = \frac{\lambda_1 B_k}{y'_{10}}. \tag{36}$$

Из (30):

$$t_k = \frac{1}{\mu}\left(\arctan\left(\frac{B_k}{A_k}\right) + \pi n_k\right), \tag{37}$$

$$t^* = \frac{1}{\mu}\left(\arctan\left(\frac{B_l}{A_l}\right) + \pi n_l\right), \tag{38}$$

$$t_T = \frac{1}{\mu}\left(\arctan\left(\frac{B_m}{A_m}\right) + \pi n_m\right). \tag{39}$$

Как видно из Рис. 3 и формул (20), ширина одной ступени не превосходит $\frac{\pi}{\mu}$, т. е. величины Δ_k, Δ_0, Δ_T не превосходят $\frac{\pi}{2\mu}$. Для конкретики полагается, что каждый компонент управления начинает действовать на первом периоде, т. е. n_k, n_l выбираются таким образом, что $t_k \in (0; \frac{\pi}{2\mu}]$, $t^* \in (0; \frac{\pi}{2\mu}]$. Кроме того, $\sin \mu t_k > 0$, $\sin \mu t^* > 0$.

Определим интервал, которому принадлежит t_T. Из Рис. 4 и формул (20):

$$t_T = t_m + \frac{(r_m + q_m - 1)\pi}{\mu},$$

$$\text{при этом } 0 < t_m \leq \frac{\pi}{2\mu}, \Rightarrow$$

$$\frac{(r_m + q_m - 1)\pi}{\mu} < t_T \leq \frac{\pi}{2\mu} + \frac{(r_m + q_m - 1)\pi}{\mu} < \frac{(r_m + q_m)\pi}{\mu}.$$

Следовательно, n_m выбирается так, что $t_T \in (\frac{(r_m + q_m - 1)\pi}{\mu}; \frac{(r_m + q_m)\pi}{\mu})$. Согласно постановке задачи, $r_k = q_k \, \forall k$, тогда $r_m + q_m - 1$ — нечетное число, $r_m + q_m$ — четное число, т. е. $\sin \mu t_T < 0$.

Далее, из системы (30), пользуясь тригонометрическими формулами $\sin x = \frac{\tan x}{\pm\sqrt{1 + \tan^2 x}}$, $\cos x = \frac{1}{\pm\sqrt{1 + \tan^2 x}}$, выпишем следующие выражения:

$$\begin{aligned} &\sin \mu t_k = \frac{\sigma_k B_k}{M_k}, \quad \cos \mu t_k = \frac{\sigma_k A_k}{M_k}, \\ &\sin \mu \Delta_k = \frac{\sqrt{\lambda_1^2 M_k^2 - \left(-\frac{\alpha}{2S_k} - 1 + \alpha\right)^2 y_{10}'^2}}{|\lambda_1| M_k}. \end{aligned} \tag{40}$$

$$\begin{aligned} &\sin \mu t^* = \frac{\sigma_0 B_l}{M_l}, \quad \cos \mu t^* = \frac{\sigma_0 A_l}{M_l}, \\ &\sin \mu \Delta_0 = \frac{\sqrt{\lambda_1^2 M_l^2 - \left(-\frac{\alpha}{2h_l} - 1 + \alpha\right)^2 y_{10}'^2}}{|\lambda_1| M_l}. \end{aligned} \tag{41}$$

$$\sin\mu t_t = \frac{\sigma_T B_m}{M_m}, \quad \cos\mu t_T = \frac{\sigma_T A_m}{M_m},$$
$$\sin\mu\Delta_T = \frac{\sqrt{\lambda_1^2 M_m^2 - \left(\dfrac{\alpha}{2h_m} + 1 - \alpha\right)^2 y_{10}'^2}}{|\lambda_1| M_m}. \tag{42}$$

Значения σ_k, σ_0, σ_T определяют знаки соответствующих синусов:

$$\sigma_k = sign(B_k), \quad \sigma_0 = sign(B_l), \quad \sigma_T = -sign(B_m).$$

Таким образом, все переменные выражены через λ_1. Теперь рассмотрим уравнение $\lambda_1 L_1 + \lambda_2 L_2 = 0$ (из (22)).

$$0 = \lambda_1 y_{10}' + \lambda_2 y_{10}^* + \frac{2}{\mu}\Bigg[\sum_{k=1}^{2} S_k(b_k \sin\mu t_k + a_k \cos\mu t_k)\sin\mu\Delta_k +$$
$$+ h_l(b_l \sin\mu t^* + a_l \cos\mu t^*)\sin\mu\Delta_0 - h_m(b_m \sin\mu t_T + a_m \cos\mu t_T)\sin\mu\Delta_T\Bigg].$$

Подставляя в это уравнение формулы (40)–(42), получим уравнение для нахождения λ_1:

$$\lambda_1|y_{10}|^2 = -\frac{2}{\mu}\Bigg[\sum_{k=1}^{2} S_k\sigma_k\sqrt{\lambda_1^2 M_k^2 - \left(-\frac{\alpha}{2S_k} - 1 + \alpha\right)^2 y_{10}'^2} +$$
$$+ h_l\sigma_0\sqrt{\lambda_1^2 M_l^2 - \left(-\frac{\alpha}{2h_l} - 1 + \alpha\right)^2 y_{10}'^2} -$$
$$- h_m\sigma_T\sqrt{\lambda_1^2 M_m^2 - \left(\frac{\alpha}{2h_m} + 1 - \alpha\right)^2 y_{10}'^2}\Bigg]. \tag{43}$$

Используя точки переключения, найденные с помощью формул (37)–(43), в аналитическом виде можно построить оптимальное управление для любых значений параметров.

§3. Программная реализация

1. Особенности программной реализации

Решение уравнения (43) относительно λ_1 является корнем функции следующего вида:

$$f(x) = kx + \sum_{i=1}^{n} \beta_i \sqrt{a_i^2 x^2 - b_i^2}, \tag{44}$$

причём k, β_i, a_i, b_i, $i = \overline{1,n}$ — константы, $k \geq 0$, $a_i > 0$, $i = \overline{1,n}$. Область определения этой функции

$$a_i^2 x^2 - b_i^2 \geq 0, \quad i = \overline{1,n}.$$

$$\text{Положим } A = \max_i \left| \frac{b_i}{a_i} \right| \geq 0, \Rightarrow$$

$$x \in \left(-\infty; -A \right] \cup \left[A; +\infty \right). \tag{45}$$

Рассмотрим производную функции $f(x)$:

$$f'(x) = k + x \sum_{i=1}^{n} \frac{\beta_i a_i^2}{\sqrt{a_i^2 x^2 - b_i^2}}.$$

$$f'(x) \xrightarrow[x \to \pm A]{} \pm\infty$$

Это говорит о том, что в окрестности границ области определения (45) производная функции неограничена. Ввиду этого, во избежание большой погрешности вычислений, в ходе программной реализации были использованы численные методы, которые не основываются на значениях производной. Также, в процессе реализации оценены интервалы нахождения корней.

I. $x \in \left[A; +\infty \right)$.

Рассмотрим сумму из правой части выражения (44):

$$\sum_{i=1}^{n} \beta_i \sqrt{a_i^2 x^2 - b_i^2} = \sum_{\substack{i=1,\\ \beta_i \geq 0}}^{n} \beta_i \sqrt{a_i^2 x^2 - b_i^2} - \sum_{\substack{i=1,\\ \beta_i < 0}}^{n} |\beta_i| \sqrt{a_i^2 x^2 - b_i^2}, \tag{46}$$

$$\sqrt{a_i^2 x^2 - b_i^2} \leq \sqrt{a_i^2 x^2} = a_i x, \tag{47}$$

$$\sqrt{a_i^2 x^2 - b_i^2} = \sqrt{(a_i x - |b_i|)(a_i x + |b_i|)} \geq \sqrt{(a_i x - |b_i|)^2} = a_i x - |b_i|. \tag{48}$$

Тогда, из (46)–(48):

$$\sum_{i=1}^{n} \beta_i \sqrt{a_i^2 x^2 - b_i^2} \geq \sum_{\substack{i=1,\\ \beta_i \geq 0}}^{n} \beta_i (a_i x - |b_i|) - \sum_{\substack{i=1,\\ \beta_i < 0}}^{n} |\beta_i| a_i x,$$

$$\sum_{i=1}^{n} \beta_i \sqrt{a_i^2 x^2 - b_i^2} \geq x \sum_{i=1}^{n} \beta_i a_i - \sum_{\substack{i=1,\\ \beta_i \geq 0}}^{n} \beta_i |b_i|.$$

Учитывая последнее неравенство, имеем:

$$f(x) = kx + \sum_{i=1}^{n} \beta_i \sqrt{a_i^2 x^2 - b_i^2} \geq kx + x \sum_{i=1}^{n} \beta_i a_i - \sum_{\substack{i=1,\\ \beta_i \geq 0}}^{n} \beta_i |b_i|,$$

$$f(x) \geq (k + \sum_{i=1}^{n} \beta_i a_i) x - \sum_{\substack{i=1,\\ \beta_i \geq 0}}^{n} \beta_i |b_i|.$$

Введем обозначения: $k_{pos} = (k + \sum_{i=1}^{n} \beta_i a_i)$, $b_{min} = -\sum_{\substack{i=1,\\ \beta_i \geq 0}}^{n} \beta_i |b_i|$, $\Rightarrow$

$$f(x) \geq k_{pos} x + b_{min}. \tag{49}$$

Таким образом, мы получили, что функция $f(x)$ на интервале $x \in \left[A; +\infty\right)$ ограничена снизу прямой, которая описывается уравнением с известными коэффициентами.

Аналогично производится оценка сверху. Из (46)–(48):

$$\sum_{i=1}^{n} \beta_i \sqrt{a_i^2 x^2 - b_i^2} \le \sum_{\substack{i=1,\\ \beta_i \ge 0}}^{n} \beta_i a_i x - \sum_{\substack{i=1,\\ \beta_i < 0}}^{n} |\beta_i|(a_i x - |b_i|),$$

$$\sum_{i=1}^{n} \beta_i \sqrt{a_i^2 x^2 - b_i^2} \le x \sum_{i=1}^{n} \beta_i a_i + \sum_{\substack{i=1,\\ \beta_i < 0}}^{n} |\beta_i||b_i|.$$

Учитывая последнее неравенство, имеем:

$$f(x) = kx + \sum_{i=1}^{n} \beta_i \sqrt{a_i^2 x^2 - b_i^2} \le kx + x \sum_{i=1}^{n} \beta_i a_i + \sum_{\substack{i=1,\\ \beta_i < 0}}^{n} |\beta_i||b_i|,$$

$$f(x) \le (k + \sum_{i=1}^{n} \beta_i a_i)x + \sum_{\substack{i=1,\\ \beta_i < 0}}^{n} |\beta_i||b_i|.$$

Обозначим $b_{max} = \sum\limits_{\substack{i=1,\\ \beta_i < 0}}^{n} |\beta_i||b_i|$, тогда

$$f(x) \le k_{pos} x + b_{max}. \tag{50}$$

II. $x \in \left(-\infty; -A\right]$.

На этом промежутке $x \le A \le 0$, следовательно, модули раскрываются с отрицательным знаком. В таком случае формулы (47), (48) примут вид:

$$\sqrt{a_i^2 x^2 - b_i^2} \le \sqrt{a_i^2 x^2} = -a_i x, \tag{51}$$

$$\sqrt{a_i^2 x^2 - b_i^2} = \sqrt{(a_i|x| - |b_i|)(a_i|x| + |b_i|)} \ge \sqrt{(a_i|x| - |b_i|)^2} = -a_i x - |b_i|. \tag{52}$$

Тогда, из (46), (51), (52):

$$\sum_{i=1}^{n} \beta_i \sqrt{a_i^2 x^2 - b_i^2} \ge \sum_{\substack{i=1,\\ \beta_i \ge 0}}^{n} \beta_i(-a_i x - |b_i|) - \sum_{\substack{i=1,\\ \beta_i < 0}}^{n} |\beta_i|(-a_i x),$$

$$\sum_{i=1}^{n} \beta_i \sqrt{a_i^2 x^2 - b_i^2} \ge -x \sum_{i=1}^{n} \beta_i a_i - \sum_{\substack{i=1,\\ \beta_i \ge 0}}^{n} \beta_i |b_i|.$$

Учитывая последнее неравенство, имеем:

$$f(x) = kx + \sum_{i=1}^{n} \beta_i \sqrt{a_i^2 x^2 - b_i^2} \geq kx - x \sum_{i=1}^{n} \beta_i a_i - \sum_{\substack{i=1,\\ \beta_i \geq 0}}^{n} \beta_i |b_i|,$$

$$f(x) \geq (k - \sum_{i=1}^{n} \beta_i a_i)x - \sum_{\substack{i=1,\\ \beta_i \geq 0}}^{n} \beta_i |b_i|.$$

Обозначим $k_{neg} = k - \sum_{i=1}^{n} \beta_i a_i$, тогда

$$f(x) \geq k_{neg} x + b_{min}. \tag{53}$$

Аналогично производится оценка сверху. Из (46), (51), (52):

$$\sum_{i=1}^{n} \beta_i \sqrt{a_i^2 x^2 - b_i^2} \leq \sum_{\substack{i=1,\\ \beta_i \geq 0}}^{n} \beta_i (-a_i x) - \sum_{\substack{i=1,\\ \beta_i < 0}}^{n} |\beta_i| (-a_i x - |b_i|),$$

$$\sum_{i=1}^{n} \beta_i \sqrt{a_i^2 x^2 - b_i^2} \leq -x \sum_{i=1}^{n} \beta_i a_i + \sum_{\substack{i=1,\\ \beta_i < 0}}^{n} |\beta_i| |b_i|.$$

Учитывая последнее неравенство, имеем:

$$f(x) = kx + \sum_{i=1}^{n} \beta_i \sqrt{a_i^2 x^2 - b_i^2} \leq kx - x \sum_{i=1}^{n} \beta_i a_i + \sum_{\substack{i=1,\\ \beta_i < 0}}^{n} |\beta_i| |b_i|,$$

$$f(x) \leq (k - \sum_{i=1}^{n} \beta_i a_i)x + \sum_{\substack{i=1,\\ \beta_i < 0}}^{n} |\beta_i| |b_i|.$$

$$f(x) \leq k_{neg} x + b_{max}. \tag{54}$$

График, иллюстрирующий применение оценок (49), (50), (53), (54) для уравнения (43) при значениях параметров $\alpha = 0.5$, $x = (95, -10)^{\mathrm{T}}$, $r_k = q_k = 3$, $h_k = 6$, $k = \overline{1,2}$, $\mu = 1$ представлен на Рис. 5. Корень ищется на промежутке, одной из границ которого является точка пересечения соответствующей ограничивающей прямой с осью Ox, а другой — граница области определения (Рис. 5). Таким образом, на указанном промежутке

функция полностью определена, что обеспечивает корректную работу программы.

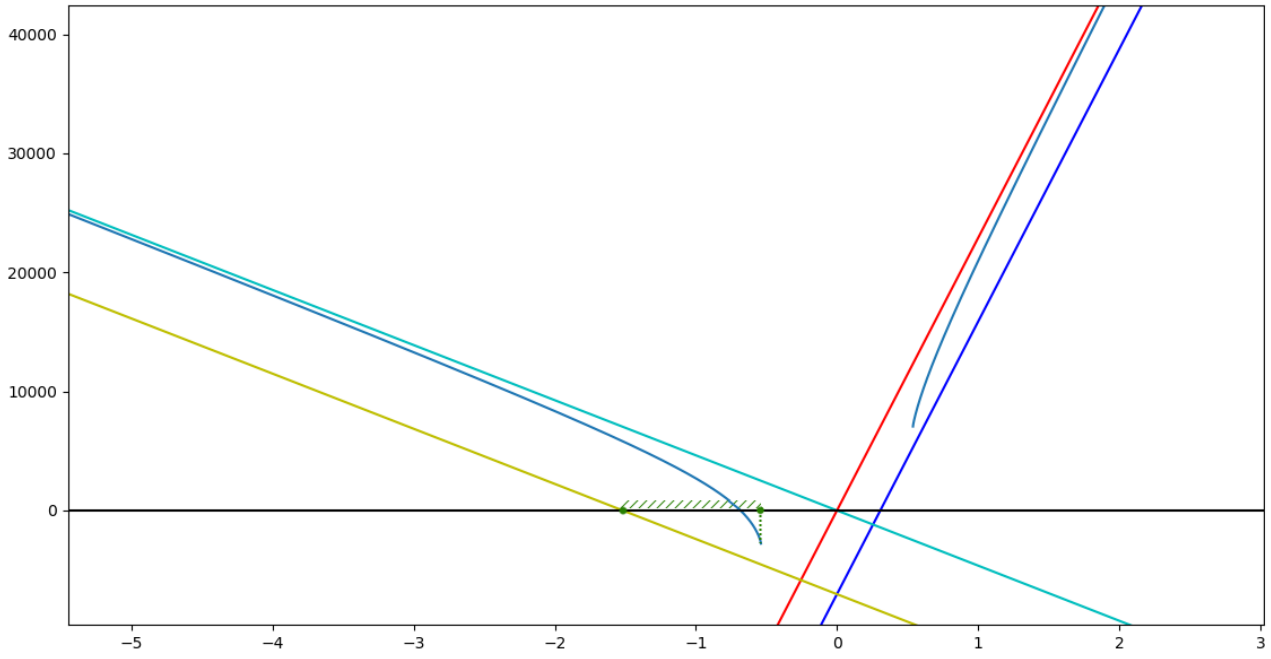

Рис. 5: Интервал нахождения корня

2. Результаты программной реализации

При некторых значениях параметров уравнение (43) может не иметь решения, т. е. в таких случаях не существует оптимального управления, доставляющего минимум функционалу (9). В процессе численной реализации методом перебора были найдены некоторые значения параметров x_0, μ, r_k, q_k, h_k, α, l, m, при которых существует решение. Графики, иллюстрирующие работу алгоритма при некоторых из этих значений приведены ниже (Рис. 7–13).

В общем случае решение исходной системы (5) без воздействия управления выглядит, как показано на Рис. 6. Левые два графика показывают изменение соответствующих фазовых переменных в зависимости от времени, правый график изображает фазовый портрет системы. Решение построено при векторе начальных данных $x_0 = (95, -10)^T$. Как видно из Рис. 6, система совершает периодические колебания, гашение которых и является целью воздействия управления.

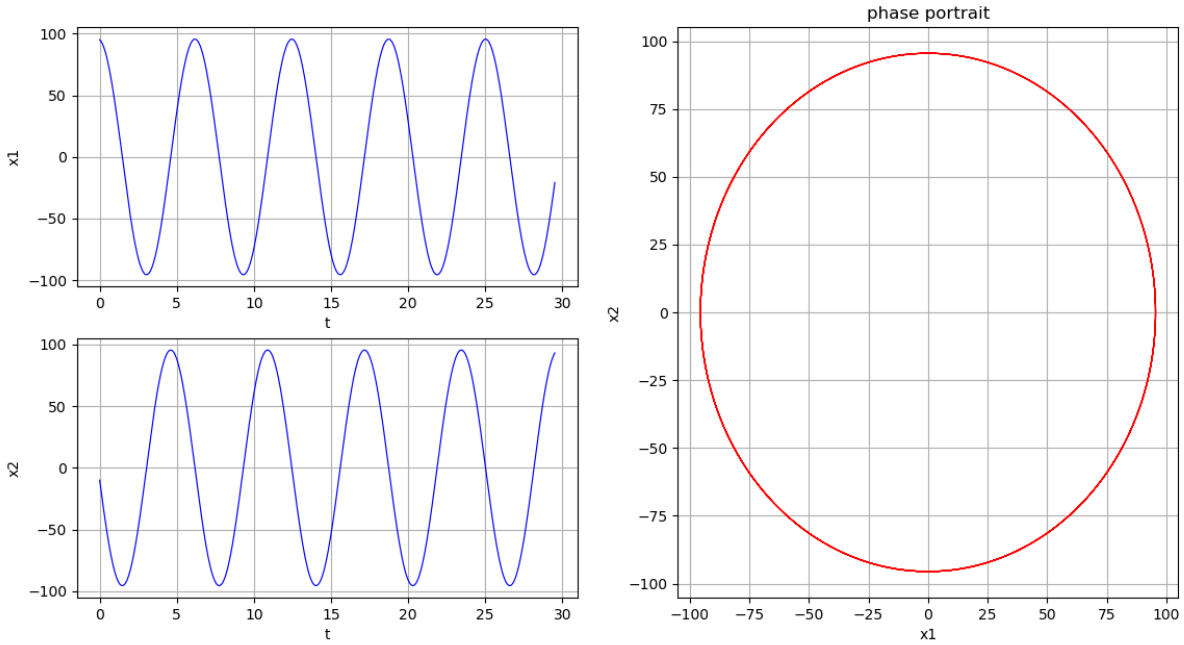

Рис. 6: Поведение системы без воздействия управления

Теперь рассмотрим аналогичные графики для системы, на которую воздействует управление, построенное согласно описанному в §2 алгоритму.

1). Приведем графики поведения системы при фиксированном значении следующих параметров: $\alpha = 0$ (что соответствует критерию по расходу топлива), $x_0 = (1, -1)^T$, $\mu = 1$, $l = 1$, $m = 2$ и различных значениях параметров r_k, q_k, h_k (Рис. 7–9).

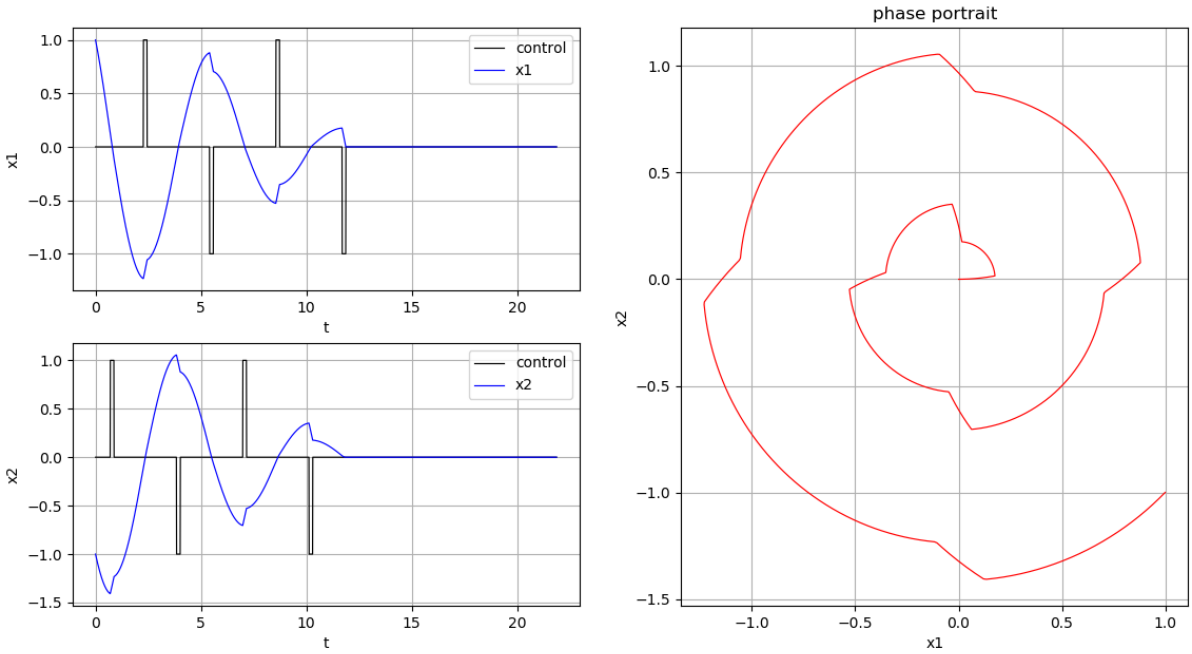

Рис. 7: $h_k = 1$, $r_k = q_k = 2$, $\forall k$

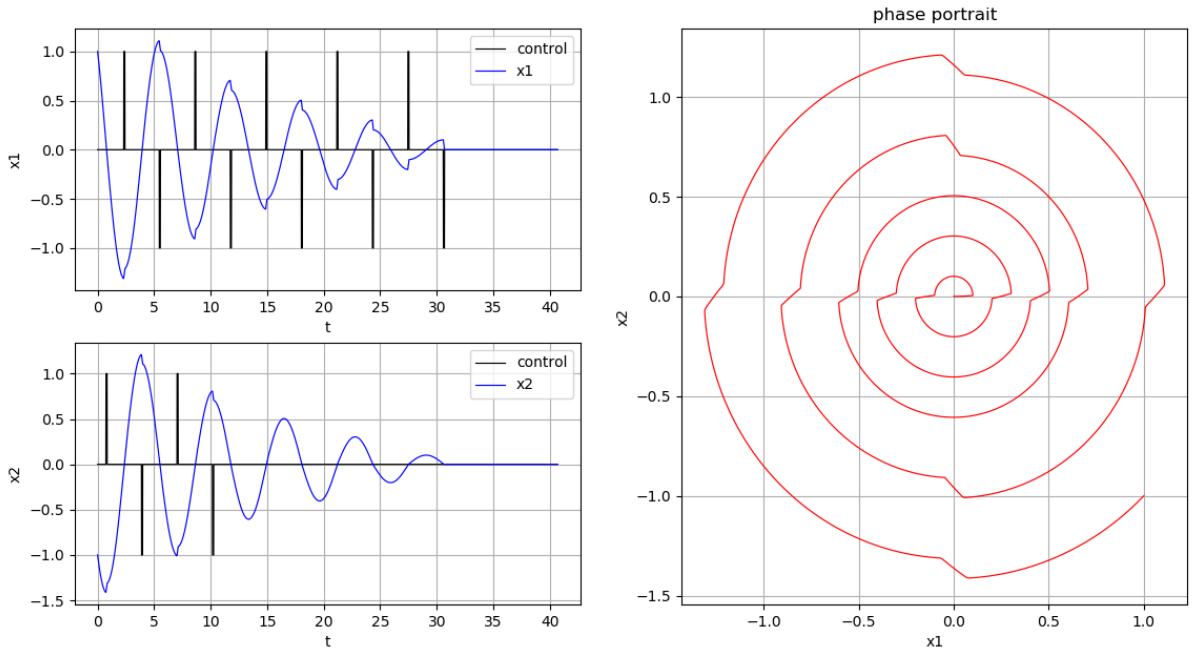

Рис. 8: $h_1 = h_2 = 1$, $r_1 = q_1 = 5$, $r_2 = q_2 = 2$

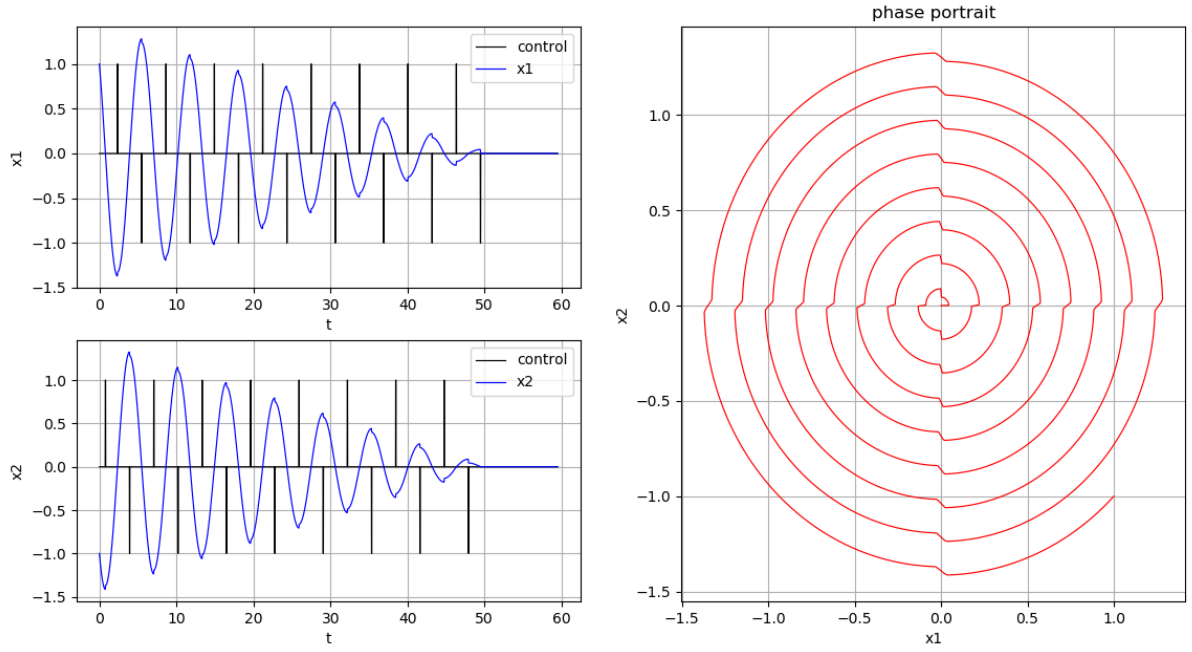

Рис. 9: $h_k = 1$, $r_k = q_k = 8$, $\forall k$

2). Приведем графики поведения системы при фиксированном значении следующих параметров: $\alpha = 1/2$, $x_0 = (95, -10)^T$, $\mu = 1$, $l = 1$, $m = 2$ и различных значениях параметров r_k, q_k, h_k (Рис. 10, 11).

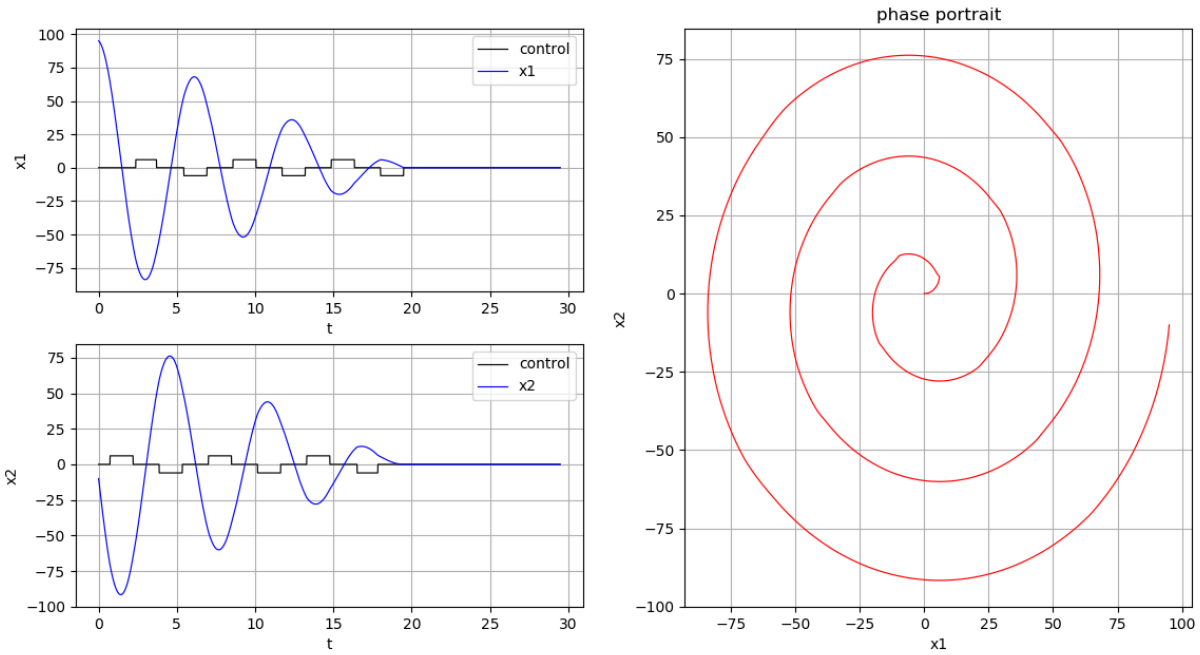

Рис. 10: $h_k = 6$, $r_k = q_k = 3$, $\forall k$

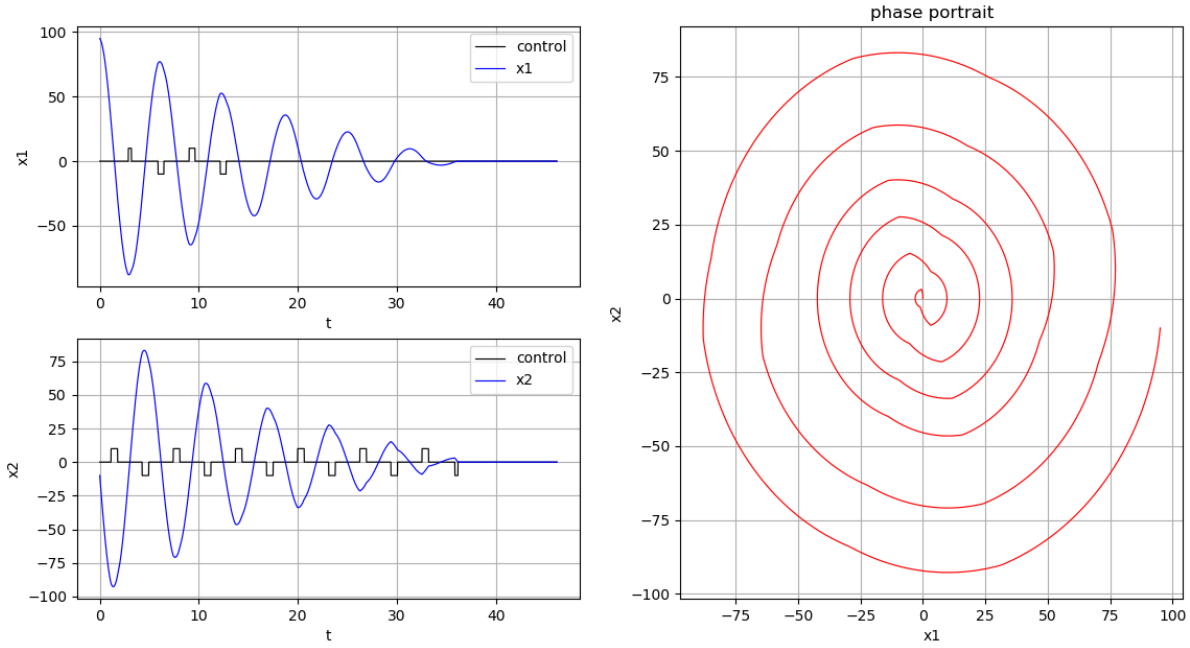

Рис. 11: $h_1 = h_2 = 10$, $r_1 = q_1 = 2$, $r_2 = q_2 = 6$

3). Приведем графики поведения системы при фиксированном значении следующих параметров: $\alpha = 1$ (что соответствует критерию по быстродействию), $x_0 = (95, -10)^T$, $\mu = 1$, $l = 1$, $m = 2$ и различных значениях параметров r_k, q_k, h_k (Рис. 12, 13).

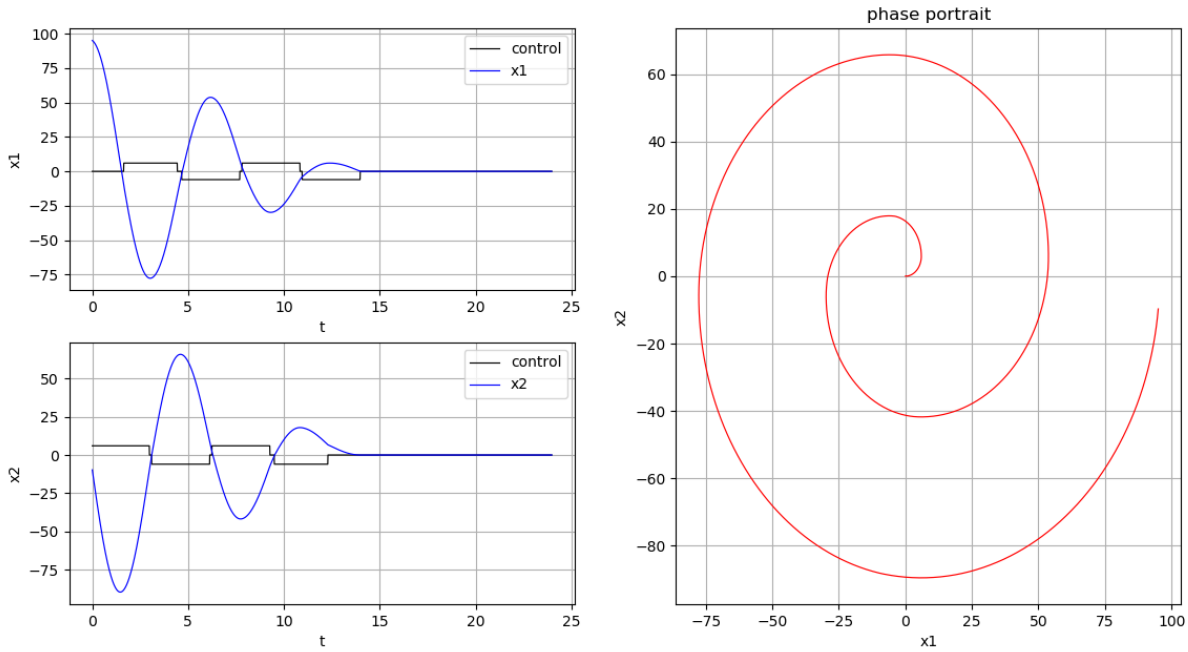

Рис. 12: $h_k = 6$, $r_k = q_k = 2$, $\forall k$

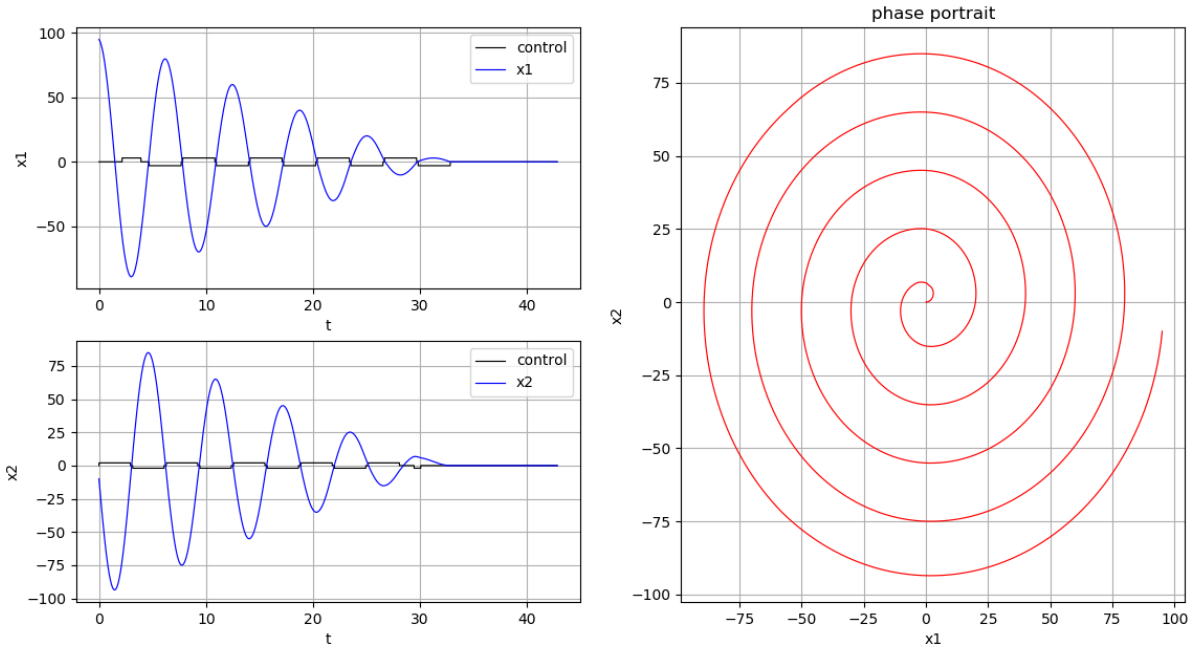

Рис. 13: $h_1 = 3$, $h_2 = 2$, $r_1 = q_1 = 5$, $r_2 = q_2 = 5$

На Рис. 11, 13 наиболее наглядно можно заметить отличие ширин первой ступени первого канала управления и последней ступени второго канала управления от остальных ширин ступеней, равных между собой и соответствующих данным каналам, что предусмотрено алгоритмом. Кроме того, на Рис. 7–9 видно, что в каждом из случаев по окончании действия

управления фазовые переменные обнуляются, а фазовые портреты соответствующих решений сходятся к нулю, что демонстрирует корректность работы алгоритма.

На Рис. 14, 15 показаны графики зависимости в отдельности каждой из частей смешанного функционала качества от параметра α. Для построения были приняты значения параметров: $x_0 = (95, -10)^T$, $\mu = 1$, $l = 1$, $m = 2$, $h_k = 6$, $r_k = q_k = 3$, $\forall k$.

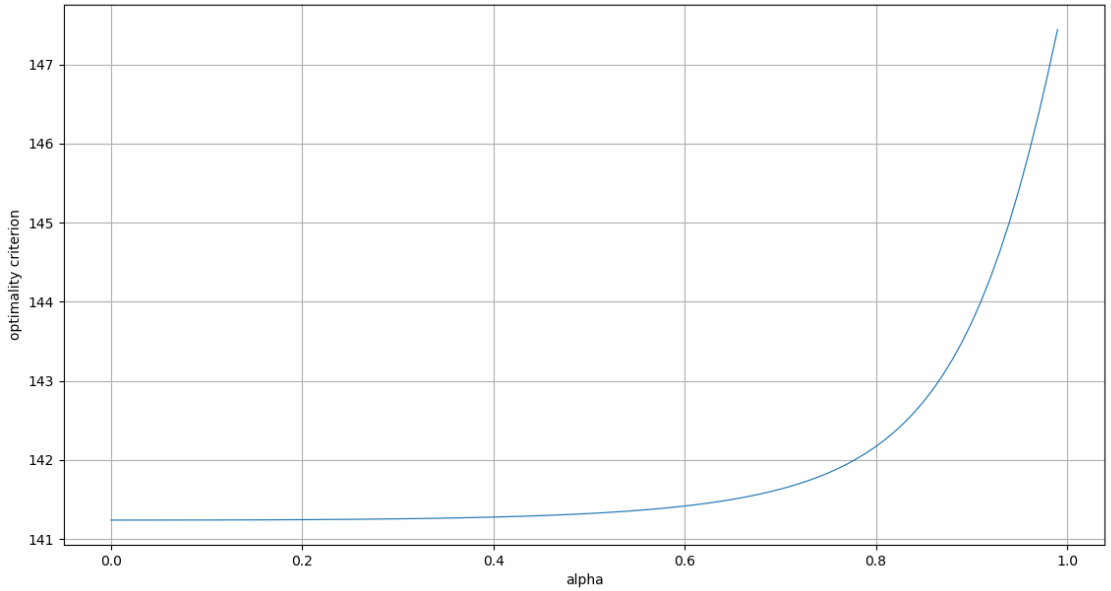

Рис. 14: Зависимость расхода топлива от параметра α

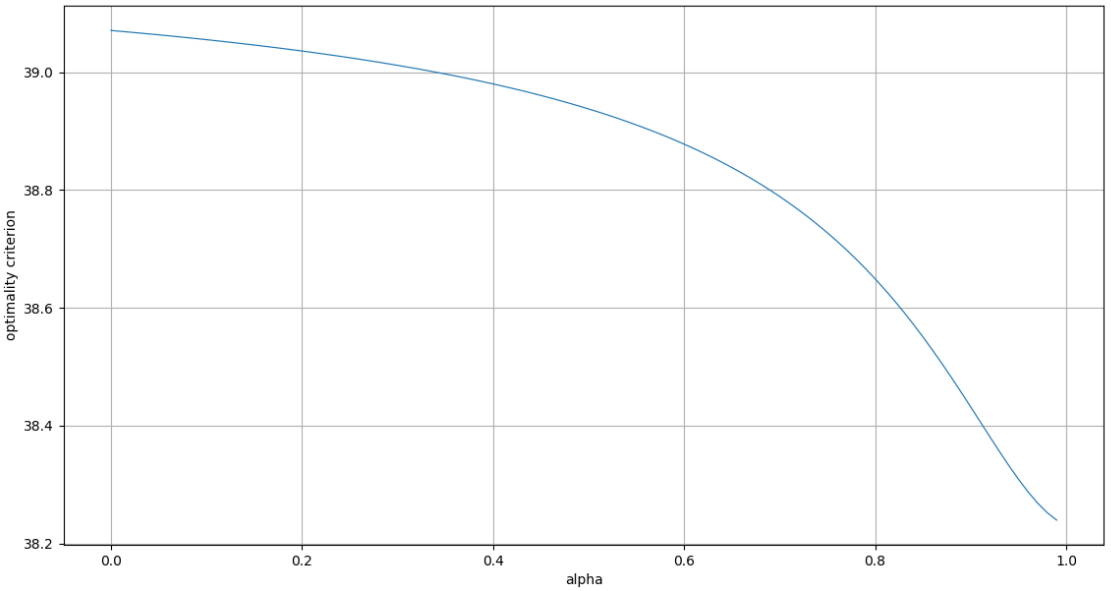

Рис. 15: Зависимость времени от параметра α

На графиках видно, что при увеличении α количество затраченного для стабилизации системы топлива увеличивается, тогда как продолжительность действия управления уменьшается. Логично, ведь при $\alpha = 0$ имеем критерий по расходу топлива, т. е. выполняется минимизация затрат ресурсов при неограниченном промежутке времени. И наоборот, при $\alpha = 1$ имеем критерий по быстродействию, т. е. необходимо стабилизировать систему в кратчайшие сроки при неограниченном объеме затраченного топлива.

При данных значениях параметров имеем: прирост количества топлива равен $|Q_{max} - Q_{min}| = 6.39559$, т. е. при максимальном возрастании α количество необходимого топлива возросло на $\approx 4,3\%$; прирост же затраченного времени равен $|T_{max} - T_{min}| = 0.85126$, т. е. при максимальном возрастании α величина затраченного времени снизилась на $\approx 2,2\%$. В соответствии с этим, можно сделать вывод, что **при изменении параметра α количество необходимого топлива изменяется быстреееe, чем время, затраченное на гашение колебаний.**

Выводы

- Построение оптимального управления для задачи (5)–(9) с заданными значениями параметров x_0, μ, r_k, q_k, h_k, α, l, m есть поиск точек переключения функции управления вида (8). Согласно построенному алгоритму, точки переключения ищутся с помощью формул (20), (21), (37)–(43). Используя их, в аналитическом виде можно построить оптимальное по смешанному критерию управление для исходной задачи при любых значениях параметров. В ходе построения алгоритма соблюдены граничные условия, а значит, фазовые переменные обратились в ноль к моменту времени T. Возвращаясь к постановке задачи, отметим, что фазовые переменные системы (5) есть проекции угловой скорости вращения КА около центра масс в связной системе координат на соответствующие оси. Полученный результат говорит о том, что после отработки реактивных двигателей космическая станция перешла в режим стационарного вращения, что и являлось целью воздействия управления.

- Найденный алгоритм реализован на языке программирования Python (см. Приложение). В ходе реализации для некоторых значений параметров были получнеы графики (Рис. 7–13), иллюстрирующие результат воздействия управления на исходную систему и подтверждающие корректность работы описанного метода. Согласно этим графикам, под воздействием управления фазовые переменные обнуляются, а соответствующий фазовый портрет системы сходится к нулю.

- На основе многочисленных экспериментов произведен анализ процентного роста/спада величины расходуемого топлива и количества затраченного времени при изменении параметра α (Рис. 14, 15): при изменении параметра α количество необходимого топлива изменяется быстрее, чем время, затраченное на гашение колебаний.

Список литературы

[1] Сю Д., Мейер А. Современная теория автоматического управления и ее применение. М.: Машиностроение, 1972. 544 с.

[2] Athans M., Falb P. L. Optimal Control: An Introduction to the Theory and Its Applications. New York: Dover Publications, 2007. 896 p.

[3] Lee E. B., Markus L. Foundations of optimal control theory. New York: Wiley, 1967. 576 p.

[4] Афанасьев В.Н., Колмановский В.Б., Носов В.Р. Математическая теория конструирования систем управления. М.: Высшая школа, 2003. 615 с.

[5] Понтрягин Л.С., Болтянский В.Г., Гамкрелидзе Р.В., Мищенко Е.Ф. Математическая теория оптимальных процессов. М.: Наука, 1983. 393 с.

[6] Бабаджанянц Л.К., Голубева Н.И., Новосёлов В.С. «Оптимальное демпфирование быстрых линейных колебаний стационарного ИСЗ с маховиком» // Пермь: Межвузовский сборник трудов «Проблемы механики управляемого движения», Вып. 3, 1973. С. 18–25.

[7] Babadzanjanz L. K. , Pototskaya I. Yu., Pupysheva Yu. Yu. "Expenditure Optimization in a Problem of Controlled Motion of Mechanical Systems" // International Conference on Numerical Analysis and Applied Mathematics 2015, AIP Conference Proceedings, 2016. P. 1–4.

[8] Бабаджанянц Л. К., Потоцкая И. Ю. Управление по критерию расхода в механических системах, Издательство СПБГУ, 2003. 137 с.

[9] Alesova I. M., Babadzanjanz L. K., Pototskaya I. Yu., Pupysheva Yu. Yu., Saakyan A. T. Control of Mechanical Systems by the "Mixed Time and

Expenditure" Criterion. // International Scientific Conference on mechanics "The Eighth Polyakhov's Reading", AIP Conference Proceedings, 2018. P. 1–4.

[10] Новосёлов В. С., Королёв В. С. Аналитическая механика управляемой системы. Уч. пособие. СПб.: ООП НИИ Химии СПбГУ, 2002. 246 с.

[11] Белецкий В. В. Движение спутника относительно центра масс. М.: Наука, 1965. 416 с.

[12] Бухгольц Н. Н. Основной курс теоретической механики, часть первая. М.: Наука, 1965. 468 с.

[13] Черноусько Ф. Л., Баничук Н. В. Вариационные задачи механики и управления. М.: Наука, 1973. 238 с.

[14] Лутц М. Изучаем Python. СПб.: Символ-Плюс, 2011. 1280 с.

[15] Brent R. P. An Algorithm with Guaranteed Convergence for Finding a Zero of a Function. // The Computer Journal, 1971. Vol. 14, No 4. P. 422—425.

Приложение

Ниже приведен программный код полученного алгоритма.

```
from numpy import arctan2, arcsin, cos, sin, pi, sign,
                    matrix, arange, array, exp, complex64
from numpy.linalg import eig
import numpy as np

def _find_first_midpoints_and_sigmas(mu, y0_1, C1, init_point):
    midpoint = list()
    sigma = list()

    for c_i in C1:
        t_i = mu ** (-1) * arctan2(
            y0_1.real * c_i.imag - y0_1.imag * c_i.real,
            y0_1.real * c_i.real + y0_1.imag * c_i.imag
        )
        m_i = 0
        step_i = sign(init_point - t_i) * pi / mu
        first_dot_interval = init_point, init_point + pi / mu

        while not first_dot_interval[0] <= t_i < first_dot_interval[1]:
            t_i += step_i
            m_i += 1
        midpoint.append(t_i)
        sigma.append((-1) ** m_i * sign(y0_1.real * c_i.imag -
                    y0_1.imag * c_i.real))
    return midpoint, sigma

def _built_root_intervals(f, domain_edge, k_pos, k_neg, b_min, b_max):
    if not k_pos:
        max_edge = None
    else:
        max_edge = -b_max / k_pos if f(domain_edge) > 0 else -b_min / k_pos
        if max_edge < domain_edge:
            max_edge = None

    if not k_neg:
        min_edge = None
    else:
```

```
        min_edge = -b_max / k_neg if f(-domain_edge) > 0 else -b_min / k_neg
        if min_edge > -domain_edge:
            min_edge = None

    root_intervals = list()
    if min_edge is not None:
        root_intervals.append((min_edge, -domain_edge))
    if max_edge is not None:
        root_intervals.append((domain_edge, max_edge))

    return root_intervals

def _solve_lambda1_equation(mu, sigma, sigma_lm, h, r, q,
                            C1, y0_1, l, m, alpha):
    _sq_y0_1_real = y0_1.real**2
    _S = [h_i * (r_i + q_i - 1) for h_i, r_i, q_i in zip(h, r, q)]
    _a = [h_i * s_i * (r_i + q_i - 1) for h_i, s_i, r_i, q_i
         in zip(h, sigma, r, q)]
    _b = [abs(y0_1) ** 2 * abs(c_i) ** 2 for c_i in C1]
    _b_lm = [_b[l], _b[m]]
    _c = [(alpha - 1 - alpha/(2 * _S_i))**2 * _sq_y0_1_real for _S_i in _S]
    _h_lm = [h[l], h[m]]
    _d_lm = [(alpha/(2 * h_i) + 1 - alpha)**2 * _sq_y0_1_real for h_i in _h_lm]
    delta_lm = [1, -1]

    def f(x):
        #  This function build by lambda1 equation,
        #root of this function will be a solution.
        # Equation hasn't modules under radicals,
        #it's added considering computational error.
        # Rupture area will be excluded from intervals of root finding.
        return abs(y0_1)**2 * x + 2 / mu *  (sum(
            _a_i *abs(x**2 * _b_i - _c_i)**(1/2) for _a_i, _b_i, _c_i
            in zip(_a, _b, _c)) + sum(
                _h_lm_i * delta_lm_i * s_lm_i *
                abs(x**2 * _b_lm_i - _d_lm_i)**(1/2)
                for _h_lm_i, delta_lm_i, s_lm_i, _b_lm_i, _d_lm_i
                in zip(_h_lm, delta_lm, sigma_lm, _b_lm, _d_lm)))

    # Let's estimate values of the function by straight lines:
```

```
# for x > 0: k_pos*x + b_min <= f(x) <= k_pos*x + b_max
# for x < 0: k_neg*x + b_min <= f(x) <= k_neg*x + b_max
# So we can find roots enclosing intervals.
_sum_from_radicals = 2 / mu * (sum(
    _a_i * _b_i**(1/2)
    for _a_i, _b_i in zip(_a, _b)
) + sum(
    _h_lm_i * delta_lm_i * s_lm_i * _b_lm_i**(1/2)
    for _h_lm_i, delta_lm_i, s_lm_i, _b_lm_i
    in zip(_h_lm, delta_lm, sigma_lm, _b_lm)))

k_pos = abs(y0_1)**2 + _sum_from_radicals
k_neg = abs(y0_1)**2 - _sum_from_radicals

b_max = 2 / mu * abs(y0_1.real) * (
    sum(
        _a_i * abs(alpha - 1 - alpha/(2 * _S_i)) for _a_i, _S_i, s_i
        in zip(_a, _S, sigma) if s_i < 0
    ) + sum(
        _h_lm_i * (alpha/(2 * _h_lm_i) + 1 - alpha)
        for _h_lm_i, delta_lm_i, s_lm_i
        in zip(_h_lm, delta_lm, sigma_lm) if delta_lm_i * s_lm_i < 0))

b_min = - 2 / mu * abs(y0_1.real) * (
    sum(
        _a_i * abs(alpha - 1 - alpha/(2 * _S_i)) for _a_i, _S_i, s_i
        in zip(_a, _S, sigma) if s_i > 0
    ) + sum(
        _h_lm_i * (alpha/(2 * _h_lm_i) + 1 - alpha)
        for _h_lm_i, delta_lm_i, s_lm_i
        in zip(_h_lm, delta_lm, sigma_lm) if delta_lm_i * s_lm_i > 0))

# rupture area is (-domain_edge, domain_edge)
domain_edge = max(max(
abs(y0_1.real)*abs(alpha - 1 - alpha/(2*_S_i))/_b_i**(1/2)
                    for _S_i, _b_i in zip(_S, _b) if abs(_b_i)),
                max(
abs(y0_1.real) * (alpha/(2 * _h_lm_i) + 1 - alpha) / _b_lm_i**(1/2)
                    for _h_lm_i, _b_lm_i in zip(_h_lm, _b_lm) if abs(_b_lm_i)
```

```
    root_intervals = _built_root_intervals(f, domain_edge, k_pos, k_neg,
                                           b_min, b_max)

    if not root_intervals:
        raise Exception("Can't find lambda1 equation solution")

    from scipy.optimize import brentq
    lambda1 = min(
        (brentq(f, *root_interval) for root_interval in root_intervals),
        key=lambda x: abs(f(x)))

    return lambda1

def create_control_intervals(mu, y0, C, h, r, q, init_point, l, m, alpha):
    y0_1 = y0[0, 0]
    C1 = C.tolist()[0]
    _S = [h_i * (r_i + q_i - 1) for h_i, r_i, q_i in zip(h, r, q)]

    midpoints,sigmas=_find_first_midpoints_and_sigmas(mu,y0_1,C1,init_point)
    sigma_lm = [sigmas[l], -sigmas[m]]
    lambda1 = _solve_lambda1_equation(mu, sigmas, sigma_lm, h, r, q,
                                      C1,y0_1,l,m,alpha)

    deltas = list()
    for i, c in enumerate(C1):
        deltas.append(1 / mu * arcsin(
            (lambda1**2 * abs(y0_1)**2 * abs(c)**2 -
            (-alpha/(2 * _S[i]) - 1 + alpha)**2 * y0_1.real**2)**(1/2) /
            (abs(lambda1) * abs(y0_1) * abs(c))))

    control_intervals = [
        [(m - d + i * pi / mu, m + d + i * pi / mu) for i in range(r_ + q_)]
        for m, d, r_, q_ in zip(midpoints, deltas, r, q)
    ]

    #finding very first and very last intervals
    delta_0 = 1/mu * arcsin(
        (lambda1**2 * abs(y0_1)**2 * abs(C1[l])**2 -
        (alpha/(2 * h[l]) + 1 - alpha)**2 * y0_1.real**2)**(1/2) /
        (abs(lambda1) * abs(y0_1) * abs(C1[l])))
```

```
    delta_T = 1/mu * arcsin(
        (lambda1**2 * abs(y0_1)**2 * abs(C1[m])**2 -
        (alpha/(2 * h[m]) + 1 - alpha)**2 * y0_1.real**2)**(1/2) /
        (abs(lambda1) * abs(y0_1) * abs(C1[m])))
    t_0 = midpoints[l]
    t_T =midpoints[m] + (r[m] + q[m] - 1)*pi/mu

    control_intervals[l][0] = t_0 - delta_0, t_0 + delta_0
    control_intervals[m][-1] = t_T - delta_T, t_T + delta_T
    return control_intervals

def create_control(mu, y0, C, h, r, q, init_point=0, control_intervals=None):
    if control_intervals is None:
        control_intervals = create_control_intervals(
            mu, y0, C, h, r, q, init_point)

    def u(t):
        res = list()
        for h_k, intervals_for_k in zip(h, control_intervals):
            for i, interval in enumerate(intervals_for_k):
                if interval[0] <= t <= interval[1]:
                    res.append((-1)**i * h_k )
                    break
            else:
                res.append(0)
        return matrix([res]).T

    return u

def count_fuel(h, control_intervals):
    return sum(h_k * sum(end - begin for begin, end in intervals_k)
               for h_k, intervals_k in zip(h, control_intervals))

def build_ode_solution(mu, y0, C, h, control_intervals):
    # 'Wasted_parts_expressions' for each of ode equation,
    # for each of control intervals component
    # collects values of spent stages integrals for number
    # of the unfinished stage
    # (Wasted_parts_expressions[ode_eq_num]
    # [c_interv_component_num][num_of_first_unfinished_stage]).
```

```
# Needs for removal repeated calculations.
wasted_parts_expressions = list()
for eq_num_in_ode in range(len(C)):
    list_for_ode_nums = list()
    for intervals_k in control_intervals:
        wasted_parts_before_this = [0j]
        for i, (t1, t2) in enumerate(intervals_k):
            _real = sin(mu * t2) - sin(mu * t1)
            _imag = (-1) ** eq_num_in_ode * (cos(mu * t2) - cos(mu * t1))
            stage_i = (-1) ** i * (_real + _imag * 1j) / mu

            previous_stages = wasted_parts_before_this[i]
            wasted_parts_before_this.append(previous_stages + stage_i)
        list_for_ode_nums.append(wasted_parts_before_this)
    wasted_parts_expressions.append(list_for_ode_nums)

# Returns value of integrals with control function in ode solution
def find_quad(t):
    result = list()
    for i, C_i in enumerate(C.tolist()):
        result_i = 0
        for k, (c_k, h_k, intervals_k) in enumerate(
        zip(C_i, h, control_intervals)):
            if t <= intervals_k[0][0]:
                result_i += 0
            elif t >= intervals_k[-1][1]:
                result_i += c_k * h_k * wasted_parts_expressions[i][k][-1]
            else:
                ind, next_interval = next((ind, (t1, t2))
                for ind, (t1, t2) in enumerate(intervals_k) if t2 > t)
                t1 = next_interval[0]

                wasted_parts = wasted_parts_expressions[i][k][ind]
                incomplete_part = 0
                if t > t1:
                    _real = sin(mu * t) - sin(mu * t1)
                    _imag = (-1) ** i * (cos(mu * t) - cos(mu * t1))
                    incomplete_part = (-1)**ind/mu * (_real + _imag * 1j)

                result_i += c_k * h_k * (wasted_parts + incomplete_part)
```

```
            result.append([result_i])
        return np.matrix(result)

    def ode_solution(t):
        return np.diag(exp(1j*mu * t * array([1, -1])))*(y0 + find_quad(t))

    return ode_solution

if __name__ == '__main__':
    a = 1
    A = matrix([[0, a], [-a, 0]])
    mu, V = abs(eig(A)[0][0].imag), eig(A)[1]
    C = matrix([[1, -1j], [1, 1j]])

    l, m, alpha = 0, 1, 1

    def x_init(t):
        def init_ode_solution(t):
            return np.diag(exp(1j * mu * t * array([1, -1]))) * y0
        return C.I * init_ode_solution(t)

    h = [6, 6]
    r = q = [2, 2]
    x0 = matrix([[95, -10]]).T
    y0 = C * x0

    control_intervals = create_control_intervals(
        mu, y0, C, h, r, q, 0, l, m, alpha
    )
    sign_12 = sign(x_init(control_intervals[0][0][1])[0, 0].real),
              sign(x_init(control_intervals[1][0][1])[1, 0].real)
    h_ = [-sign_12[0] * h[0], -sign_12[1] * h[1]]
    u = create_control(mu,y0,C,h_,r, q,control_intervals=control_intervals)
    ode_solution = build_ode_solution(mu, y0, C, h_, control_intervals)

    from pylab import *
    def plot_graph():
        figure(1)
        gcf().canvas.set_window_title(u'solution of the system')
```

```
T = max(control_interval_k[-1][1] for control_interval_k
        in control_intervals)
t = arange(0.0, T + 10, 0.005)
t_ = arange(0.0, T + 10, 0.01)

data_u = list(zip(*(u(t_i).T.tolist()[0] for t_i in t)))

x = lambda t: C.I * ode_solution(t)
data_x = [{'real': list(), 'imag': list()},
          {'real': list(), 'imag': list()}]
for i in t_:
    x_i = x(i)
    data_x[0]['real'].append(x_i[0, 0].real)
    data_x[0]['imag'].append(x_i[0, 0].imag)
    data_x[1]['real'].append(x_i[1, 0].real)
    data_x[1]['imag'].append(x_i[1, 0].imag)

subplot(221)
xlabel(u't')
ylabel(u'x1')
control_line, solution_line = plot(t, data_u[0], 'k',
         t_, data_x[0]['real'], 'b', lw=0.85)
legend((control_line, solution_line), (u'control', u'x1'))
grid()

subplot(223)
xlabel(u't')
ylabel(u'x2')
control_line, solution_line = plot(t, data_u[1], 'k',
         t_, data_x[1]['real'], 'b', lw=0.85)
legend((control_line, solution_line), (u'control', u'x2'))
grid()

subplot(122)
xlabel(u'x1')
ylabel(u'x2')
title(u'phase portrait')
plot(data_x[0]['real'], data_x[1]['real'], 'r', lw=0.85)
grid()
```

```
def plot_init_graph():
    figure(2)
    gcf().canvas.set_window_title(u'solution of the init system')
    T = max(control_interval_k[-1][1] for control_interval_k
        in control_intervals)
    t_ = arange(0.0, T + 10, 0.01)

    data_x_init = [{'real': list(), 'imag': list()},
                   {'real': list(), 'imag': list()}]
    for i in t_:
        x_i = x_init(i)
        data_x_init[0]['real'].append(x_i[0, 0].real)
        data_x_init[0]['imag'].append(x_i[0, 0].imag)
        data_x_init[1]['real'].append(x_i[1, 0].real)
        data_x_init[1]['imag'].append(x_i[1, 0].imag)

    subplot(221)
    xlabel(u't')
    ylabel(u'x1')
    plot(t_, data_x_init[0]['real'], 'b', lw=0.85)
    grid()

    subplot(223)
    xlabel(u't')
    ylabel(u'x2')
    plot(t_, data_x_init[1]['real'], 'b', lw=0.85)
    grid()

    subplot(122)
    xlabel(u'x1')
    ylabel(u'x2')
    title(u'phase portrait')
    plot(data_x_init[0]['real'], data_x_init[1]['real'], 'r', lw=0.85)
    grid()

def plot_criterion():
    figure(3)
    gcf().canvas.set_window_title(u'optymality criterion')
    fuel_ = list()
    time_ = list()
```

```
    for alpha_ in arange(0, 0.9, 0.001):
        print(alpha_)
        control_intervals = create_control_intervals(
            mu, y0, C, h, r, q, 0, l, m, alpha_
        )
        fuel_.append(count_fuel(h, control_intervals))
        control_intervals = create_control_intervals(
            mu, y0, C, h, r, q, 0, l, m, 1-alpha_
        )
        time_.append(max(control_interval_k[-1][1] for control_interval_k
                     in control_intervals) -
                     min(control_interval_k[0][0] for control_interval_k
                     in control_intervals))
    _alpha = arange(0, 0.9, 0.001)

    subplot(121)
    xlabel(u'alpha')
    ylabel(u'fuel')
    title(u'fuel of alpha')
    plot(_alpha, fuel_, lw=0.85)
    grid()

    subplot(122)
    xlabel(u'alpha')
    ylabel(u'time')
    title(u'time of alpha')
    plot(_alpha, time_, lw=0.85)
    grid()

plot_criterion()
plot_init_graph()
plot_graph()
show()
```