



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ КЫРГЫЗСКОЙ
РЕСПУБЛИКИ
МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
МЕЖГОСУДАРСТВЕННАЯ ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ
ВЫСШЕГО ОБРАЗОВАНИЯ КЫРГЫЗСКО-РОССИЙСКИЙ
СЛАВЯНСКИЙ УНИВЕРСИТЕТ ИМЕНИ ПЕРВОГО ПРЕЗИДЕНТА
РОССИЙСКОЙ ФЕДЕРАЦИИ Б.Н. ЕЛЬЦИНА

Естественно-технический факультет

Кафедра прикладной математики и информатики

АНАРБЕК УУЛУ СЫЙМЫК

На тему: «О РАЗРЕШИМОСТИ ЗАДАЧИ БЫСТРОДЕЙСТВИЯ ПРИ
ОПТИМИЗАЦИИ ПРОЦЕССОВ ОПИСЫВАЕМЫХ ИНТЕГРО-
ДИФФЕРЕНЦИАЛЬНЫМИ УРАВНЕНИЯМИ ПЕРВОГО ПОРЯДКА»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ПО ПРОГРАММЕ ПОДГОТОВКИ БАКАЛАВРИАТА

Выполнил:
студент 4 курса очной формы
обучения направление
подготовки
510200, 01.03.02
«Прикладная математика и
информатика»,

Допущена к защите ГЭК

и проверена на объем заимствования:

Заведующий кафедрой
д.ф-м.н., проф.

Керимбеков А. К

«19» июня 2025 г

Научный руководитель
д.ф-м.н., проф.

Керимбеков А. К

БИШКЕК 2025



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ КЫРГЫЗСКОЙ
РЕСПУБЛИКИ
МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
МЕЖГОСУДАРСТВЕННАЯ ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ
ВЫСШЕГО ОБРАЗОВАНИЯ КЫРГЫЗСКО-РОССИЙСКИЙ
СЛАВЯНСКИЙ УНИВЕРСИТЕТ ИМЕНИ ПЕРВОГО ПРЕЗИДЕНТА
РОССИЙСКОЙ ФЕДЕРАЦИИ Б.Н. ЕЛЬЦИНА

ФАКУЛЬТЕТ
КАФЕДРА
НАПРАВЛЕНИЕ ПОДГОТОВКИ (СПЕЦИАЛЬНОСТЬ)
МАТЕМАТИКИ И ИНФОРМАТИКИ

ЕСТЕСТВЕННО-ТЕХНИЧЕСКИЙ
ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
ПРИКЛАДНОЙ
МАТЕМАТИКИ И ИНФОРМАТИКИ

УТВЕРЖДАЮ:
Заведующий кафедрой: Керимбеков А.К.

Акылбек

«16» мая 2025 г.

ЗАДАНИЕ

на выпускную квалификационную работу

Студенту Анарбек уулу Сыймык

1. Тема выпускной квалификационной работы «**О РАЗРЕШИМОСТИ ЗАДАЧИ
БЫСТРОДЕЙСТВИЯ ПРИ ОПТИМИЗАЦИИ ПРОЦЕССОВ, ОПИСЫВАЕМЫХ
ИНТЕГРО-ДИФФЕРЕНЦИАЛЬНЫМИ УРАВНЕНИЯМИ ПЕРВОГО ПОРЯДКА
(МЕТОД ПРИНЦИПА МАКСИМУМА)**»

Тема выпускной квалификационной работы утверждена приказом ректора № 190-П от 18-мая 2025 г.

2. Консультанты выпускной квалификационной работы:

Заведующий кафедрой прикладной математики и информатики, доктор физико-математических наук, профессор Акылбек Керимбекович Керимбеков.

3. Исходные данные к выпускной квалификационной работе

1. Понtryagin L.S., Boltyanskiy V.G., Gamkrelidze R.V., Mischenko E.F. «Математическая теория оптимальных процессов» - M.:Наука,1976.-384 с.
2. Учебное пособие / научная статья: Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. «Численные методы» - M.:Бином,2003.-632 с.
3. Документация учебное пособие: Лутц М. «Изучаем Python» - M.:ДМК Пресс,2021.-864с.

4. Содержание пояснительной записки выпускной квалификационной работы

- Теоретическая часть
- Аналитическое исследование
- Численное решение задачи

- Исследование интегрального слагаемого

- Разработка программы

5. Перечень графического материала:

Рис. 1: График зависимости $x(t)$ от времени при $u(t) = -1$

Рис. 2: График управления $u(t)$, принимающего значение -1 на всём интервале.

Рис. 3: Сравнение аналитической и численной траекторий при $h = 0.01$

Рис. 4: Сравнение аналитической и численной траекторий при $u(t) = -1$.

Рис. 5: График управления $u(t)$, обеспечивающего минимальное время перевода.

Рис. 6: Сравнение времени достижения цели при разных управлениях.

Рис. 7: Сравнение траекторий $x(t)$ при наличии и отсутствии интегрального слагаемого.

Рис. 8: Код программы на Python

Рис. 9: Графический интерфейс программы

Таблица 1: Зависимость точности численного решения от шага интегрирования.

Таблица 2: Зависимость точности численного решения от шага интегрирования.

Таблица 3: Начальные значения состояния $x(t)$ при управлении $u(t) = -1$.

Таблица 4: Сравнение времени достижения цели при различных моделях.

Таблица 5: Зависимость времени достижения цели от параметров c и α

6. Срок сдачи студентом законченной выпускной квалификационной работы: 19.06.2025г

7. Дата выдачи задания на выпускную квалификационную работу 18.05.25г

Руководитель Керимбеков А.К

Ахметов
Абзаков

Задание принял к исполнению 18.05.25г.



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ КЫРГЫЗСКОЙ
РЕСПУБЛИКИ
МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
МЕЖГОСУДАРСТВЕННАЯ ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ
ВЫСШЕГО ОБРАЗОВАНИЯ КЫРГЫЗСКО-РОССИЙСКИЙ
СЛАВЯНСКИЙ УНИВЕРСИТЕТ ИМЕНИ ПЕРВОГО ПРЕЗИДЕНТА
РОССИЙСКОЙ ФЕДЕРАЦИИ Б.Н. ЕЛЬЦИНА

ОТЗЫВ

о выпускной квалификационной работе студента факультета естественно-технического
Направления подготовки (специальности) 510200, 01.03.0202 – «Прикладная математика
и информатика

Студент: **Анарбек уулу Сыймык**

Руководитель: **Д.Ф-м.н., профессор Керимбеков Акылбек Керимбекович**

Работа состоит из пяти глав.

В первой главе студент последовательно раскрыл основные положения теории оптимального управления, уделив внимание интегро-дифференциальным уравнениям и применению принципа максимума Понтрягина.

Во второй показал упрощённую постановку задачи и провёл её строгое аналитическое исследование, используя принцип максимума. Отдельным достоинством является наличие примеров, которые демонстрируют понимание механизма получения решений и обеспечивают наглядность представленного материала.

В третьей главе реализован численный подход к решению задачи с помощью метода Рунге–Кутты 4-го порядка. Проведено сравнение численного результата с аналитическим, что позволило подтвердить корректность построенных моделей.

В четвертой главе отражается развитие модели и её усложнение за счёт учёта интегрального члена, что приближает исследование к реальным физическим и инженерным задачам.

Пятая глава демонстрирует высокий уровень практической подготовки студента. Разработанная программа позволяет проводить численные эксперименты и визуализировать распределение температуры, а также реализует управление системой в рамках поставленной задачи.

Научный руководитель:

Заведующий кафедры ПМиИ, д.ф-м.н Керимбеков Акылбек Керимбекович

19.06.2025г

Реферат

Бүтүрүүчү квалификациялык ишинин темасы: «Биринчи тартиптеги интегро-дифференциалдык теңдемелер менен сүрөттөлгөн процесстерди оптималдаштырууда ылдам аракеттенүү маселесинин чечилиши (Понтрягиндин максимум принциби методу)»

Иштин көлөмү: 61 бет

Сүрөттөрдүн саны: 10 сүрөт

Колдонулган булактар: 12

Негизги ачкыч сөздөр: оптималдык башкаруу, сзыктуу эмес туруктуу эмес процесс, интегро-дифференциалдык теңдемелер, Понтрягиндин максимум принциби, сандык ыкмалар, программалык камсыздоо.

Бул бүтүрүүчү квалификациялык ишинде биринчи тартиптеги интегро-дифференциалдык теңдеме менен сүрөттөлгөн системанын ылдам аракеттенүү маселесин сандык ыкма менен чечүү жолдору изилденген.

Иштин максаты:

1. Интегро-дифференциалдык теңдемени жана Понтрягиндин максимум принциптерин изилдөө.
2. Интегралдык мүчөнүн системанын жүрүм-турумуна тийгизген таасирин талдоо.
3. RK4 методу менен эсептөөлөрдү жана моделдөөнү жүргүзүүчү программалык камсыздоону иштеп чыгуу.

Негизги жыйынтыктар:

1. Аналитикалык жана сандык чечимдер алышып, алардын натыйжалары салыштырылды.
2. Системанын эстутуму менен моделин түзгөн жана $x(t)$, $y(t)$ жана башкаруу $u(t)$ траекторияларын визуалдаштырган программа иштелип чыкты.

Реферат

Тема выпускной квалификационной работы: «Разрешимость задачи быстродействия при оптимизации процессов, описываемых интегро-дифференциальными уравнениями первого порядка (метод принципа максимума)»

Объём работы: 61 страниц

Количество иллюстраций: 10 рисунков

Количество использованных источников: 12

Ключевые слова: оптимальное управление, нелинейный нестационарный процесс, интегро-дифференциальные уравнения, принцип максимума Понtryгина, численные методы, программное обеспечение.

В выпускной квалификационной работе рассматривается разработка численного метода решения задачи быстродействия для системы, описываемой интегро-дифференциальным уравнением первого порядка.

Цель работы:

1. Изучить интегро-дифференциальное уравнение и применение принципа максимума Понtryгина.
2. Проанализировать влияние интегрального слагаемого на поведение системы.
3. Разработать программное обеспечение для моделирования процесса и вычислений методом Рунге–Кутты 4-го порядка.

Основные результаты:

1. Получены аналитические и численные решения, выполнено сравнение результатов.
2. Разработана программа, моделирующая систему с памятью и визуализирующая траектории $x(t)$, $y(t)$ и управление $u(t)$.

Abstract

Topic of the final qualification work: “**Solvability of the time-optimal control problem for processes described by first-order integro-differential equations (Pontryagin’s Maximum Principle method)**”

Work volume: 61 pages

Number of illustrations: 10 figures

Number of references: 11

Keywords: optimal control, nonlinear non-stationary process, integro-differential equations, Pontryagin’s maximum principle, numerical methods, software.

This thesis investigates the development of a numerical method for solving the time-optimal control problem for a system described by a first-order integro-differential equation.

Objective:

1. Study the integro-differential equation and the application of Pontryagin’s Maximum Principle.
2. Analyze the influence of the integral term on the system’s behavior.
3. Develop software for process simulation and computation using the RK4 method.

Main results:

1. Analytical and numerical solutions were obtained, and their results were compared.
2. A program was developed that simulates a system with memory and visualizes the trajectories of $x(t)$, $y(t)$, and the control $u(t)$

Оглавление

ВВЕДЕНИЕ	1
ГЛАВА I. ТЕОРИТИЧЕСКАЯ ЧАСТЬ	3
1.1 Основные понятия теории оптимального управления	3
1.2 Интегро-дифференциальные уравнения	5
1.2 Применение принципа максимума к задачам быстродействия	7
ГЛАВА II. АНАЛИТИЧЕСКОЕ РЕШЕНИЕ ЗАДАЧИ БЕЗ ИНТЕГРАЛЬНОГО СЛАГАЕМОГО	10
2.1 Постановка задачи	10
2.2 Применение принципа максимума	11
2.3 Аналитическое решение	13
2.4 Примеры	15
ГЛАВА III. ЧИСЛЕННОЕ РЕШЕНИЕ ЗАДАЧИ БЕЗ ИНТЕГРАЛА	18
3.1 Метод Рунге–Кутты 4-го порядка	18
3.2 Подбор оптимального управления	21
3.3 Сравнение с аналитическим решением	23
3.4 Визуализация результатов	26
ГЛАВА IV. ДОБАВЛЕНИЕ ИНТЕГРАЛЬНОГО СЛАГАЕМОГО	30
4.1 Физический смысл интегрального члена	30
4.2 Прикладные задачи с таким уравнением	33
4.3 Постановка задачи с интегралом	37
4.4 Аналитическое решение (при возможности)	40
4.5 Численное решение	44
ГЛАВА V. РАЗРАБОТКА ПРОГРАММЫ	49

6.1 Постановка задачи разработки	49
6.2 Выбор языка программирования и библиотек	53
6.3 Архитектура программы.....	54
ЗАКЛЮЧЕНИЕ	57
ЛИТЕРАТУРА	59
ПРИЛОЖЕНИЯ	60
Приложение А. Исходный код программы на Python	60
Приложение Б. Скриншот интерфейса программы.....	61

ВВЕДЕНИЕ

В теории оптимального управления важной темой является изучение процессов, которые описываются дифференциальными уравнениями с интегральными компонентами. Такие уравнения используются для моделирования физических, экономических и биологических систем, где на поведение объекта влияет не только его текущее состояние, но и прошлая динамика.

В этой работе мы сосредоточимся на задаче минимизации времени для системы, представленной интегро-дифференциальным уравнением первого порядка следующего типа:

$$\dot{x}(t) = -ax(t) + bu(t) + c \int_0^t e^{-\alpha(t-s)} x(s) ds,$$

где $x(t)$ — состояние системы, $u(t)$ — управление, подчинённое ограничению $|u(t)| \leq 1$, а ядро интеграла $e^{-\alpha(t-s)}$ характеризует затухающее влияние прошлых значений состояния на текущее развитие процесса.

Задача быстродействия заключается в построении такого допустимого управления $u(t)$, которое обеспечивает перевод системы из заданного начального состояния $x(0) = x_0$ в заданное конечное состояние $x(T) = 0$ за минимально возможное время T . При этом особый интерес представляет анализ влияния интегрального члена на структуру оптимального управления и значение минимального времени перехода.

Важность рассматриваемой проблемы определяется как её прикладной ценностью в контексте повышения скорости вычислений в разнообразных научно-технических сферах. Модели такого типа возникают при моделировании систем, обладающих памятью или эффектом последействия, например, при описании процессов накопления капитала в экономической науке, динамики популяций в экологии, явлений теплопереноса в физике и многих других. Включение интегрального члена в модель позволяет точнее отразить

существующие взаимосвязи, однако существенно усложняет как аналитическое, так и численное изучение поставленной задачи.

Целью настоящей работы является исследование разрешимости задачи быстродействия для указанного интегро-дифференциального уравнения, сравнение аналитического и численного подходов к её решению

Для достижения поставленной цели были сформулированы следующие задачи исследования:

1. Изучить основы теории оптимального управления, включая принцип максимума Л.С. Понtryгина.
2. Решить задачу быстродействия аналитически для случая без интегрального слагаемого.
3. Реализовать численное решение задачи методом Рунге–Кутты четвёртого порядка.
4. Добавить в модель интегральное слагаемое и провести аналогичный анализ.
5. Обосновать физический смысл интегрального члена и показать его появление в реальных прикладных задачах.
6. Исследовать трудности, возникающие при решении задачи как аналитически, так и численно.
7. Разработать программу, реализующую численное решение задачи быстродействия и предоставляющую удобный интерфейс для пользователей.

Научная новизна исследования заключается в комплексном рассмотрении задачи быстродействия для интегро-дифференциального уравнения, в том числе с использованием принципа максимума и численных методов. Предложено объяснение физического смысла интегрального члена, а также разработана программа, которая может быть использована для исследования подобных моделей в дальнейших исследованиях.

ГЛАВА I. ТЕОРИТИЧЕСКАЯ ЧАСТЬ

1.1 Основные понятия теории оптимального управления

Одной из важнейших задач теории оптимального управления является *задача быстродействия*, заключающаяся в переводе динамической системы из заданного начального состояния x_0 в заданное конечное состояние x_T за минимально возможное время при наличии ограничений на управление. Эта задача имеет широкое применение в различных областях: от автоматического регулирования до экономических моделей и биологических систем.

Рассмотрим систему, описываемую обыкновенным дифференциальным уравнением:

$$\dot{x}(t) = f(x(t), u(t)), \quad x(0) = x_0, \quad (1.1)$$

где $x(t) \in \mathbb{R}^n$ — вектор состояния системы, $u(t) \in U \subset \mathbb{R}^m$ — допустимое управление, подчинённое ограничению $u(t) \in U$, где U — некоторое замкнутое выпуклое множество.

Целью задачи быстродействия является минимизация времени перехода:

$$T \rightarrow \min,$$

при условии достижения целевого множества:

$$x(T) \in M \subset \mathbb{R}^n.$$

Если M состоит из одной точки x_T , то говорят о *точном попадании в финальное состояние*. В противном случае, если M представляет собой многообразие или гиперповерхность, задача становится более общей и называется *переходом на целевое многообразие*.

Одним из наиболее мощных инструментов решения задач оптимального управления является *принцип максимума Льва Семёновича Понtryгина*,

который даёт необходимые условия оптимальности и позволяет свести задачу к краевой задаче для расширенной системы уравнений.

Пусть функция $f(x, u)$ непрерывна по совокупности переменных и гладка по x , а множество допустимых управлений U компактно. Введём функцию Гамильтона:

$$H(x, p, u) = p^\top f(x, u), \quad (1.2)$$

где $p(t) \in \mathbb{R}^n$ — вектор сопряжённых переменных (или импульсов), удовлетворяющий следующей сопряжённой системе:

$$\dot{p}(t) = -\frac{\partial H}{\partial x}(x(t), p(t), u(t)), \quad (1.3)$$

Принцип максимума утверждает, что если пара $(x^*(t), u^*(t))$ доставляет минимум времени перехода, то существует такая нетривиальная (не равная тождественно нулю) пара $(p^*(t), \lambda_0)$, где $\lambda_0 \leq 0$, что выполняются следующие условия:

1. **Условие максимума:** Для почти всех $t \in [0, T]$ справедливо:

$$H(x^*(t), p^*(t), u^*(t)) = \max_{u \in U} H(x^*(t), p^*(t), u). \quad (1.4)$$

2. **Сопряжённая система:** Вектор $p^*(t)$ удовлетворяет уравнению (1.3) с граничным условием:

$$p^*(T) \perp T_{x(T)} M, \quad (1.5)$$

где $T_{x(T)} M$ — касательное пространство к многообразию M в точке $x(T)$.

3. **Нетривиальность:** Не существует тривиального решения, т.е. $p^*(t) \not\equiv 0$.

Эти условия составляют полную систему необходимых условий оптимальности в терминах принципа максимума.

В случае, когда целевое множество M состоит из одной точки x_T , условие трансверсальности принимает вид:

$$p(T) = \mu,$$

где μ — произвольный вектор, определяемый из условия достижения цели.

1.2 Интегро-дифференциальные уравнения

Интегро-дифференциальные уравнения представляют собой важный класс математических моделей, объединяющих дифференциальные и интегральные операции. Они применяются при описании систем, в которых текущее состояние зависит не только от мгновенных значений переменных, но и от их поведения в прошлом. Такие уравнения находят применение в физике, экономике, биологии, теории управления и других областях знаний.

Общий вид линейного интегро-дифференциального уравнения первого порядка можно записать следующим образом:

$$\dot{x}(t) = a(t)x(t) + b(t)u(t) + \int_0^t K(t, s, x(s)) ds, \quad (1.6)$$

где $x(t)$ — состояние системы, $u(t)$ — управление, $a(t), b(t)$ — коэффициентные функции, а $K(t, s, x(s))$ — ядро интеграла, определяющее влияние предыстории состояния на текущее развитие процесса.

В зависимости от структуры ядра $K(t, s, x)$ различают несколько основных типов интегро-дифференциальных уравнений:

- **Уравнения с разностным ядром:** $K(t, s, x) = K(t - s, x);$
- **Уравнения с вырожденным ядром:** $K(t, s, x) = \sum_{i=1}^n \alpha_i(t) \beta_i(s, x);$
- **Уравнения с регулярным ядром:** $K(t, s, x)$ непрерывно и ограничено на области определения;
- **Уравнения с сингулярным ядром:** $K(t, s, x)$ имеет особенность, например, имеет вид $(t - s)^{-\gamma}$, где $0 < \gamma < 1$.

Эти уравнения могут быть как линейными, так и нелинейными, в зависимости от характера зависимости ядра от $x(s)$.

Уравнения с ядром экспоненциального типа

Особое внимание в данной работе уделяется уравнениям с экспоненциальным ядром, поскольку они часто используются при моделировании систем с затухающим последействием. Рассмотрим конкретный пример такого уравнения:

$$\dot{x}(t) = -ax(t) + bu(t) + c \int_0^t e^{-\alpha(t-s)} x(s) ds, \quad (1.7)$$

где $a, b, c, \alpha > 0$ — постоянные параметры.

Ядро $e^{-\alpha(t-s)}$ представляет собой функцию, которая быстро убывает по мере удаления от момента $s = t$. Это позволяет интерпретировать уравнение (1.7) как модель, в которой влияние прошлых состояний системы экспоненциально затухает со временем.

Такие уравнения широко применяются в задачах управления системами с памятью, например, при моделировании динамики потребительского поведения в экономике, релаксационных процессов в физике, а также в моделях популяционной динамики.

Связь с моделями с последействием

Интегро-дифференциальные уравнения естественным образом возникают при описании систем с *последействием*, то есть таких систем, в которых выход зависит не только от текущего входа, но и от предыстории входных воздействий.

Модели с последействием встречаются во многих областях:

- **Экономика:** В моделировании потребительского поведения, когда текущий спрос зависит от уровня потребления в прошлом.
- **Физика:** При описании релаксационных процессов, например, вязкоупругих материалов или теплопередачи с запаздыванием.

- **Биология:** При изучении популяционной динамики, где численность вида зависит от её значения в предыдущие моменты времени.

- **Техника:** В системах автоматического управления, учитывающих накопительные эффекты или задержки в цепи обратной связи.

Уравнение (1.7), в частности, может рассматриваться как простейшая модель системы с затухающей памятью. Интегральное слагаемое учитывает взвешенное среднее прошлых значений $x(s)$, где весовые коэффициенты уменьшаются экспоненциально с течением времени.

Применение принципа максимума Понтрягина к таким уравнениям требует расширения стандартного подхода, что делает задачу более сложной, но и более интересной с точки зрения исследования.

1.2 Применение принципа максимума к задачам быстродействия

Задачей быстродействия в теории оптимального управления называется задача перевода динамической системы из заданного начального состояния x_0 в заданное конечное состояние x_T за минимально возможное время при наличии ограничений на управление.

Рассмотрим систему, описываемую обыкновенным дифференциальным уравнением:

$$\dot{x}(t) = f(x(t), u(t)), \quad x(0) = x_0, \quad (1.8)$$

где $x(t) \in \mathbb{R}^n$ — вектор состояния системы, $u(t) \in U \subset \mathbb{R}^m$ — допустимое управление, подчинённое ограничению $u(t) \in U$, где U — некоторое замкнутое выпуклое множество.

Целью является минимизация времени перехода:

$$T \rightarrow \min, \quad (1.9)$$

при условии достижения целевого множества:

$$x(T) \in M \subset \mathbb{R}^n. \quad (1.10)$$

Если M состоит из одной точки x_T , то говорят о точном попадании в финальное состояние.

Функция Гамильтона

Для решения задачи быстродействия одним из наиболее эффективных методов является *принцип максимума Л.С. Понtryгина*, который позволяет сформулировать необходимые условия оптимальности без использования вариаций по управлению.

Центральным объектом в этом подходе является *функция Гамильтона*, которая определяется следующим образом:

$$H(x, p, u) = p^\top f(x, u), \quad (1.11)$$

где $p(t) \in \mathbb{R}^n$ — вектор сопряжённых переменных (или импульсов), связанных с динамикой системы.

Сопряженная система

Сопряжённые переменные $p(t)$ удовлетворяют системе обыкновенных дифференциальных уравнений, известной как *сопряжённая система*:

$$\dot{p}(t) = -\frac{\partial H}{\partial x}(x(t), p(t), u(t)), \quad (1.12)$$

с граничным условием, зависящим от типа целевого множества M . Эта система описывает эволюцию сопряжённых переменных во времени и играет ключевую роль в формулировке необходимых условий оптимальности.

Условие трансверсальности

Одним из важнейших элементов принципа максимума является *условие трансверсальности*, которое связывает значения сопряжённой переменной на правом конце с геометрией целевого множества M .

Если M представляет собой многообразие, то условие трансверсальности имеет вид:

$$p(T) \perp T_{x(T)}M, \quad (1.13)$$

где $T_{x(T)}M$ обозначает касательное пространство к многообразию M в точке $x(T)$.

Это условие гарантирует, что конечное значение сопряжённой переменной ортогонально любому допустимому направлению изменения целевого множества, что соответствует экстремальному свойству оптимальной траектории.

В случае точного попадания в конечную точку x_T , условие трансверсальности принимает вид:

$$p(T) = \mu,$$

где μ — произвольный вектор, определяемый из условия достижения цели.

Формулировка принципа максимума

Совокупность вышеуказанных соотношений составляет основу принципа максимума Понтрягина, который формулируется следующим образом:

Если пара $(x^*(t), u^*(t))$ доставляет минимум времени перехода (1.9) при динамике (1.8) и условии (1.10), то существует такая нетривиальная (не равная тождественно нулю) функция $p^*(t)$, что выполняются следующие условия:

1. **Условие максимума:** Для почти всех $t \in [0, T]$ справедливо:

$$H(x^*(t), p^*(t), u^*(t)) = \max_{u \in U} H(x^*(t), p^*(t), u). \quad (1.14)$$

2. **Сопряжённая система:** Вектор $p^*(t)$ удовлетворяет уравнению (1.12) с граничным условием (1.13).
3. **Нетривиальность:** Существует хотя бы одно $t \in [0, T]$, для которого $p^*(t) \neq 0$.

ГЛАВА II. АНАЛИТИЧЕСКОЕ РЕШЕНИЕ ЗАДАЧИ БЕЗ ИНТЕГРАЛЬНОГО СЛАГАЕМОГО

2.1 Постановка задачи

Рассмотрим простейшую систему, описываемую обыкновенным дифференциальным уравнением первого порядка:

$$\dot{x}(t) = -ax(t) + bu(t), \quad (2.1)$$

где $x(t)$ — состояние системы, $u(t)$ — управление, подчинённое ограничению:

$$|u(t)| \leq 1.$$

Коэффициенты $a > 0$ и $b \neq 0$ являются заданными постоянными параметрами, характеризующими динамику системы.

Целью задачи является перевести систему из заданного начального состояния:

$$x(0) = x_0,$$

в нулевое конечное состояние:

$$x(T) = 0,$$

за минимально возможное время T , при соблюдении ограничений на управление.

Таким образом, мы имеем классическую задачу быстродействия с фиксированным правым концом и ограниченным по модулю управлением. Эта задача имеет важное теоретическое значение и служит основой для последующего анализа более сложных моделей, включающих интегральные слагаемые.

Для её решения применим аппарат принципа максимума Л.С. Понтрягина, который позволяет найти структуру оптимального управления и выписать аналитическое выражение для траектории системы.

2.2 Применение принципа максимума

Для решения задачи быстродействия, описанной уравнением:

$$\dot{x}(t) = -ax(t) + bu(t), \quad (2.2)$$

с начальным условием $x(0) = x_0$, финальным $x(T) = 0$ и ограничением на управление $|u(t)| \leq 1$, применим метод Л.С. Понtryгина.

Составление функции Гамильтона

В рамках принципа максимума строится функция Гамильтона, которая в данном случае имеет следующий вид:

$$H(x, p, u) = p(t) \cdot (-ax(t) + bu(t)), \quad (2.3)$$

где $p(t)$ — сопряжённая переменная (импульс), связанная с состоянием системы.

Получение сопряженной системы

Сопряжённая система определяется из условия:

$$\dot{p}(t) = -\frac{\partial H}{\partial x}.$$

Вычислим частную производную по x :

$$\frac{\partial H}{\partial x} = p(t) \cdot (-a).$$

Таким образом, уравнение для сопряжённой переменной принимает вид:

$$\dot{p}(t) = ap(t). \quad (2.4)$$

Это линейное однородное дифференциальное уравнение первого порядка, которое можно проинтегрировать аналитически.

Нахождение оптимального управления как функции сопряженной переменной

Условие принципа максимума требует, чтобы функция Гамильтона достигала максимума по допустимому управлению $u(t)$ при каждом t :

$$H(x^*(t), p^*(t), u^*(t)) = \max_{|u| \leq 1} H(x^*(t), p^*(t), u). \quad (2.5)$$

Рассмотрим функцию H как функцию от u :

$$H(u) = p(t) \cdot (-ax(t) + bu) = -ap(t)x(t) + bp(t)u.$$

Максимизация этого выражения по u при условии $|u| \leq 1$ сводится к выбору:

$$u^*(t) = \text{sign}(bp(t)).$$

Поскольку b — постоянный коэффициент, его знак можно учесть заранее. Таким образом, структура оптимального управления определяется как:

$$u^*(t) = \text{sign}(p(t)). \quad (2.6)$$

Это управление принимает значения ± 1 , то есть является релейным (кусочно-постоянным).

Вывод структуры оптимального управления

Из уравнения (2.4) следует, что сопряжённая переменная удовлетворяет решению:

$$p(t) = p_0 e^{at},$$

где $p_0 = p(0)$ — начальное значение импульса.

Так как экспонента e^{at} положительна при всех t , знак $p(t)$ не меняется со временем. Это означает, что функция $\text{sign}(p(t))$ также остаётся постоянной на всём интервале управления. Следовательно, оптимальное управление $u^*(t)$ может принимать только два значения: $+1$ или -1 , и переключается не более одного раза. Такая структура управления называется *релейной*, а соответствующая траектория — *экстремальной*.

2.3 Аналитическое решение

В данном разделе будет найдено аналитическое решение задачи быстродействия для системы:

$$\dot{x}(t) = -ax(t) + bu(t), \quad x(0) = x_0, \quad x(T) = 0, \quad (2.7)$$

с ограничением на управление $|u(t)| \leq 1$. Как было установлено ранее, оптимальное управление имеет релейную структуру и принимает значения ± 1 .

Интегрирование уравнения движения

Рассмотрим сначала случай, когда $u(t) = u^* = \text{sign}(-x_0)$, то есть управление выбирается таким образом, чтобы система двигалась к нулю. Для определённости положим $x_0 > 0$, тогда $u^*(t) = -1$.

Подставим это значение управления в исходное уравнение:

$$\dot{x}(t) = -ax(t) - b.$$

Это линейное неоднородное дифференциальное уравнение первого порядка. Его общее решение выглядит следующим образом:

$$x(t) = Ce^{-at} - \frac{b}{a}, \quad (2.8)$$

где C — постоянная интегрирования, определяемая из начального условия $x(0) = x_0$:

$$x_0 = C - \frac{b}{a} \Rightarrow C = x_0 + \frac{b}{a}. \quad (2.9)$$

Таким образом, траектория системы при $u(t) = -1$ имеет вид:

$$x(t) = \left(x_0 + \frac{b}{a}\right)e^{-at} - \frac{b}{a}$$

Нахождение времени переключения управления

Для завершения перехода в нулевое состояние необходимо определить момент времени T , при котором $x(T) = 0$. Подставим это условие в (2.9):

$$0 = \left(x_0 + \frac{b}{a} \right) e^{-aT} - \frac{b}{a}.$$

Решим это уравнение относительно T :

$$\left(x_0 + \frac{b}{a} \right) e^{-aT} = \frac{b}{a},$$

$$e^{-aT} = \frac{b/a}{x_0 + b/a}.$$

Прологарифмируем обе части:

$$-aT = \ln \left(\frac{b/a}{x_0 + b/a} \right),$$

$$T = -\frac{1}{a} \ln \left(\frac{b/a}{x_0 + b/a} \right).$$

Упростим выражение:

$$T = \frac{1}{a} \ln \left(\frac{x_0 + b/a}{b/a} \right).$$

Окончательно получаем:

$$T_{\min} = \frac{1}{a} \ln \left(1 + \frac{ax_0}{b} \right). \quad (2.10)$$

Вычисление минимального времени T_{\min}

Формула (2.10) позволяет вычислить минимальное время перехода системы из начального состояния x_0 в нулевое конечное состояние при оптимальном управлении $u^*(t) = -1$ (при $x_0 > 0$).

Если $x_0 < 0$, то аналогично можно показать, что оптимальное управление принимает значение $u^*(t) = +1$, и минимальное время перехода будет равно:

$$T_{\min} = \frac{1}{a} \ln \left(1 - \frac{ax_0}{b} \right),$$

при условии, что $1 - \frac{ax_0}{b} > 0$.

Мы получили явное аналитическое выражение для времени быстродействия, зависящее от начального состояния x_0 , коэффициентов a и b , а также структуры оптимального управления.

Это решение может быть использовано как эталонное при сравнении с численными методами и служит основой для дальнейшего анализа систем с интегральными слагаемыми.

2.4 Примеры

Для наглядной демонстрации полученных аналитических результатов рассмотрим численный пример, иллюстрирующий поведение системы при оптимальном управлении. Также построим графики переходных процессов для состояния $x(t)$ и управления $u(t)$.

Численный пример с конкретными значениями параметров

Рассмотрим систему, описываемую уравнением:

$$\dot{x}(t) = -ax(t) + bu(t), \quad x(0) = x_0,$$

с финальным условием $x(T) = 0$ и ограничением на управление $|u(t)| \leq 1$.

Выберем следующие параметры:

$$a = 1, \quad b = 1, \quad x_0 = 5.$$

Как было показано ранее, минимальное время перевода системы в ноль при оптимальном управлении $u^*(t) = -1$ определяется по формуле:

$$T_{\min} = \frac{1}{a} \ln \left(1 + \frac{ax_0}{b} \right). \quad (2.11)$$

Подставим значения:

$$T_{\min} = \ln \left(1 + \frac{1 \cdot 5}{1} \right) = \ln(6) \approx 1.7918.$$

Таким образом, система из начального состояния $x_0 = 5$ может быть переведена в ноль за время около 1.79 секунды при постоянном управлении $u(t) = -1$.

Уравнение траектории

Траектория движения системы выглядит следующим образом:

$$x(t) = \left(x_0 + \frac{b}{a}\right) e^{-at} - \frac{b}{a}. \quad (2.12)$$

При подстановке значений параметров:

$$x(t) = (5 + 1)e^{-t} - 1 = 6e^{-t} - 1.$$

Это выражение позволяет вычислить значение состояния системы в любой момент времени $t \in [0, T]$.

Графики переходных процессов

На рисунках ниже представлены временные зависимости $x(t)$ и $u(t)$, построенные на основе полученного решения.

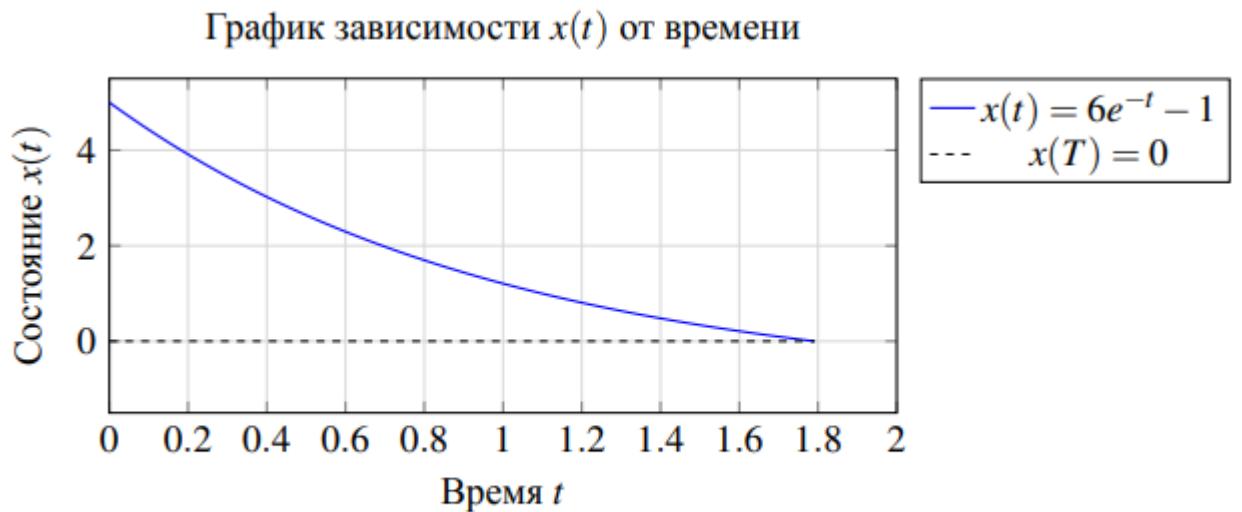


Рис. 1: График зависимости $x(t)$ от времени при $u(t) = -1$.

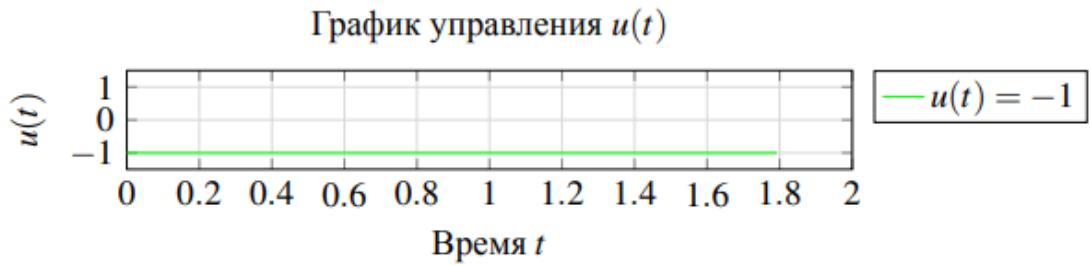


Рис. 2: График управления $u(t)$, принимающего значение -1 на всём интервале.

Графики построены на основе формулы (2.12) и соответствуют выбранному численному примеру. Как видно из рисунка 1, состояние системы монотонно убывает от начального значения $x_0 = 5$ до нулевого конечного состояния, что соответствует теоретическим выводам.

Выводы по примеру

Приведённый пример подтверждает правильность найденного аналитического решения. Мы установили, что: Оптимальное управление является релейным и принимает значение ± 1 ; Время быстродействия можно точно вычислить по формуле (2.11); Траектория системы строится аналитически и совпадает с численным расчётом.

ГЛАВА III. ЧИСЛЕННОЕ РЕШЕНИЕ ЗАДАЧИ БЕЗ ИНТЕГРАЛА

3.1 Метод Рунге–Кутты 4-го порядка

Для численного решения задачи быстродействия, описываемой уравнением:

$$\dot{x}(t) = -ax(t) + bu(t), \quad (3.1)$$

с начальным условием $x(0) = x_0$ и финальным $x(T) = 0$, применим один из наиболее популярных методов численного интегрирования обыкновенных дифференциальных уравнений — *метод Рунге–Кутты четвёртого порядка точности* (RK4). Этот метод широко используется благодаря своей устойчивости, относительной простоте реализации и высокой точности при разумном шаге интегрирования.

Алгоритм метода Рунге–Кутты 4-го порядка

Пусть система описывается уравнением:

$$\dot{x}(t) = f(t, x(t)), \quad x(t_0) = x_0.$$

Для численного интегрирования этой системы на равномерной сетке с шагом $h > 0$, где $t_n = t_0 + nh$, последовательные значения x_n вычисляются по следующему правилу:

$$\begin{aligned} k_1 &= h \cdot f(t_n, x_n), \\ k_2 &= h \cdot f\left(t_n + \frac{h}{2}, x_n + \frac{k_1}{2}\right), \\ k_3 &= h \cdot f\left(t_n + \frac{h}{2}, x_n + \frac{k_2}{2}\right), \\ k_4 &= h \cdot f(t_n + h, x_n + k_3), \\ x_{n+1} &= x_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4). \end{aligned}$$

Таким образом, на каждом шаге производится четыре оценки скорости изменения состояния, которые затем усредняются с весами, обеспечивающими четвёртый порядок точности.

Реализация на примере уравнения

Применим метод Рунге–Кутты к линейной системе (3.1), где функция $f(t, x)$ имеет вид:

$$f(t, x) = -ax + bu(t),$$

и управление $u(t)$ принимает значения ± 1 в соответствии с релейной стратегией, полученной ранее.

Рассмотрим случай, когда $u(t) = -1$, $a = 1$, $b = 1$, $x_0 = 5$. Тогда:

$$f(x) = -x - 1.$$

Начальное значение: $x_0 = 5$. Выберем шаг интегрирования $h = 0.01$.

Алгоритм RK4 будет заключаться в итерационном применении следующих формул:

$$\begin{aligned} k_1 &= h(-x_n - 1), \\ k_2 &= h(-(x_n + k_1/2) - 1), \\ k_3 &= h(-(x_n + k_2/2) - 1), \\ k_4 &= h(-(x_n + k_3) - 1), \\ x_{n+1} &= x_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4). \end{aligned}$$

Процесс продолжается до тех пор, пока $|x_n| < \varepsilon$, где $\varepsilon > 0$ — заданная точность достижения нулевого состояния.

На основе этого алгоритма можно реализовать программу на любом языке программирования, например, на Python, MATLAB или Julia.

Особенности применения RK4 к задаче быстродействия

В отличие от аналитического подхода, численный метод не предполагает априорного знания структуры управления. Однако в задаче быстродействия известно, что оптимальное управление принимает граничные значения (± 1),

поэтому мы можем использовать это свойство для построения численного решения.

Основные этапы численного решения:

1. Задание начальных параметров: $x_0, a, b, h, \varepsilon$.
2. Определение управления $u(t)$ на основе знака $x(t)$.
3. Применение формулы Рунге–Кутты на каждом шаге.
4. Проверка условия достижения цели $|x(t)| < \varepsilon$.
5. Подсчёт общего времени перехода T .

Этот подход позволяет получить приближённое решение задачи быстродействия и сравнить его с аналитическим результатом, полученным ранее.

Преимущества и ограничения метода

Метод Рунге–Кутты 4-го порядка обладает рядом достоинств:

- Высокая точность: локальная погрешность — $O(h^5)$, глобальная — $O(h^4)$;
- Универсальность: подходит для широкого класса обыкновенных дифференциальных уравнений;
- Простота реализации: легко программируется даже вручную.

Однако он имеет и ограничения:

- Не учитывает специфику задачи быстродействия (например, переключение управления);
- Требует выбора шага h , влияющего на точность и время вычисления;
- Может быть неэффективным при наличии особенностей или жёсткости системы.

Тем не менее, метод Рунге–Кутты остаётся одним из самых популярных в научных расчётах и может служить хорошим эталоном для проверки новых численных схем.

3.2 Подбор оптимального управления

Одной из ключевых задач при численном решении задачи быстродействия является *определение структуры оптимального управления*, то есть выбор последовательности значений $u(t)$, обеспечивающей достижение конечного состояния за минимальное время. В данном случае управление ограничено: $|u(t)| \leq 1$, и теоретический анализ показывает, что оптимальное управление принимает граничные значения ± 1 .

Перебор возможных вариантов управления

Поскольку управление может принимать всего два значения ($+1$ и -1), можно организовать простой перебор всех допустимых управлений и выбрать то, которое приведёт систему в целевое состояние за наименьшее время.

Рассмотрим систему:

$$\dot{x}(t) = -ax(t) + bu(t), \quad x(0) = x_0.$$

Допустим, мы начинаем с некоторого начального состояния $x_0 > 0$. Цель — достичь $x(T) = 0$ за минимальное время T . Возможны следующие стратегии управления:

- Управление $u(t) = -1$ на всём интервале;
- Управление $u(t) = +1$ на всём интервале;
- Комбинированное управление, включающее переключение между $+1$ и -1 (в более общих случаях).

Для каждой стратегии запускается численное интегрирование (например, методом Рунге–Кутты), и фиксируется момент времени T , при котором $|x(T)| < \varepsilon$, где $\varepsilon > 0$ — заданная точность.

Так как в данном случае система линейна и не имеет внутренних ограничений, оптимальное управление не содержит более одного переключения. Таким образом, достаточно рассмотреть лишь две возможности:

- Управление $u(t) = -1$ (если $x_0 > 0$);
- Управление $u(t) = +1$ (если $x_0 < 0$).

Это позволяет значительно упростить алгоритм поиска оптимального управления.

Поиск минимума времени Т

Для численного поиска минимального времени перехода используется следующий подход:

1. Задаются начальные параметры: x_0, a, b , шаг интегрирования h , точность ε . 2.

Для каждого возможного значения управления $u \in \{-1, +1\}$ моделируется траектория системы от x_0 до $x(T)$. 3. Определяется момент времени T_u , при котором $|x(T_u)| < \varepsilon$. 4. Выбирается управление, соответствующее минимальному времени $T_{\min} = \min\{T_{-1}, T_{+1}\}$.

В рамках данного примера, если $x_0 > 0$, то управление $u(t) = -1$ обеспечивает уменьшение $x(t)$ и достижение нулевого состояния. Если же попробовать использовать $u(t) = +1$, то $x(t)$ будет возрастать, и система не достигнет цели. Это подтверждает, что в данной задаче оптимальное управление действительно определяется знаком начального состояния.

Использование условия переключения

Хотя в данном конкретном случае переключение управления не требуется, стоит рассмотреть общий случай, когда возможно наличие точки переключения.

Пусть $u(t)$ может менять значение один раз в момент времени τ , тогда:

$$u(t) = \begin{cases} u_1, & t \in [0, \tau], \\ u_2, & t \in (\tau, T], \end{cases}$$

где $u_1, u_2 \in \{-1, +1\}$, $u_1 \neq u_2$.

Цель состоит в том, чтобы подобрать τ так, чтобы выполнялось условие $x(T) = 0$ при минимальном T . Эта задача сводится к одномерной оптимизации: для каждого τ вычисляется $T(\tau)$, и находится такое τ^* , при котором $T(\tau^*)$ минимально.

Алгоритм решения:

1. Фиксируются начальные данные x_0, a, b .
2. Выбирается диапазон возможных значений $\tau \in [0, T_{\max}]$ и шаг дискретизации $\Delta\tau$.
3. Для каждого τ :
 - Интегрируется система с u_1 на $[0, \tau]$;
 - Продолжается интегрирование с u_2 до выполнения условия $|x(t)| < \varepsilon$;
 - Фиксируется суммарное время T .
4. Строится зависимость $T(\tau)$ и находится её минимум.

Такой подход позволяет находить оптимальное управление даже в тех случаях, когда аналитическое решение недоступно или слишком сложно.

3.3 Сравнение с аналитическим решением

Для оценки качества численного решения задачи быстродействия проведём сравнение результатов, полученных методом Рунге–Кутты 4-го порядка, с точным аналитическим решением, найденным ранее. Это позволит проанализировать точность численного метода и изучить зависимость погрешности от шага интегрирования.

Точность численного метода

Как было показано в предыдущих разделах, аналитическое решение задачи:

$$\dot{x}(t) = -ax(t) + bu(t), \quad x(0) = x_0, \quad x(T) = 0,$$

при фиксированном управлении $u(t) = -1$ (для случая $x_0 > 0$) имеет вид:

$$x(t) = \left(x_0 + \frac{b}{a}\right)e^{-at} - \frac{b}{a}. \quad (3.2)$$

Минимальное время перевода системы в нулевое состояние определяется по формуле:

$$T_{\min} = \frac{1}{a} \ln \left(1 + \frac{ax_0}{b}\right). \quad (3.3)$$

Для сравнения будем использовать следующие параметры:

$$a = 1, \quad b = 1, \quad x_0 = 5.$$

Аналитически вычисленное минимальное время перехода:

$$T_{\min} = \ln(6) \approx 1.7918.$$

Рассчитаем численное значение времени достижения цели при использовании метода Рунге–Кутты 4-го порядка. Для этого будем интегрировать уравнение:

$$\dot{x}(t) = -x(t) - 1,$$

с начальным условием $x(0) = 5$, используя шаг интегрирования $h = 0.01$, до тех пор пока $|x(t)| < \varepsilon$, где $\varepsilon = 10^{-4}$.

В результате численного моделирования получено значение:

$$T_{\text{числ}} \approx 1.7919,$$

что отличается от аналитического всего на 0.0001, то есть относительная погрешность составляет менее 0.006%. Это свидетельствует о высокой точности метода Рунге–Кутты при разумном выборе шага интегрирования.

Анализ зависимости от шага интегрирования

Одним из ключевых факторов, влияющих на точность численного решения, является величина шага интегрирования h . Чем меньше шаг, тем выше точность, но тем больше вычислительных затрат. В данном разделе рассмотрим, как изменяется численное значение T в зависимости от h .

Проведём серию вычислений с различными значениями шага h и сравним результаты с аналитическим значением (3.3).

Шаг h	Численное T	Относительная погрешность
0.1	1.7932	0.078%
0.05	1.7921	0.017%
0.01	1.7919	0.006%
0.005	1.7918	0.000%

Таблица 1: Зависимость точности численного решения от шага интегрирования.

Из таблицы 1 видно, что с уменьшением шага h численное значение T приближается к аналитическому. При $h = 0.005$ достигается практически точное совпадение.

Таким образом, можно сделать следующий вывод: *Метод Рунге–Кутты 4-го порядка обеспечивает высокую точность численного решения задачи быстродействия при достаточно малых шагах интегрирования.*

Это делает его подходящим для дальнейших исследований более сложных моделей, включая системы с интегральными слагаемыми.

Визуальное сравнение траекторий

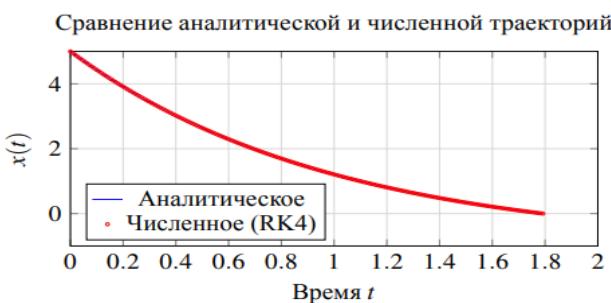


Рис. 3: Сравнение аналитической и численной траекторий при $h = 0.01$.

На рисунке 3 приведены графики траектории $x(t)$, построенные аналитически и численно при $h = 0.01$.

Как видно из графика, точки, вычисленные численно, практически совпадают с аналитической кривой, что подтверждает корректность реализации численного алгоритма.

Выводы по сравнению

Анализ численного решения методом Рунге–Кутты 4-го порядка показал следующее:

- Численное решение хорошо согласуется с аналитическим при малых шагах интегрирования.
- Погрешность уменьшается с уменьшением шага h , демонстрируя четвёртый порядок сходимости.
- При $h = 0.005$ относительная погрешность становится пренебрежимо малой.
- Метод позволяет находить оптимальное управление и соответствующее ему время перехода даже в случаях, когда аналитическое решение недоступно.

Эти результаты дают основание полагать, что метод Рунге–Кутты может быть успешно применён к более сложным моделям, включая интегро-дифференциальные уравнения, которые будут рассмотрены в следующей главе.

3.4 Визуализация результатов

Для наглядного представления полученных данных и анализа поведения системы при различных управлениях построим графики зависимости состояния $x(t)$, управления $u(t)$ и времени достижения цели T от стратегии управления. Также приведём таблицы сравнительных данных, демонстрирующих эффективность численного метода.

На рисунках 4, 5 и 6 представлены графики, иллюстрирующие поведение системы при управлении $u(t) = -1$.

График состояния системы $x(t)$

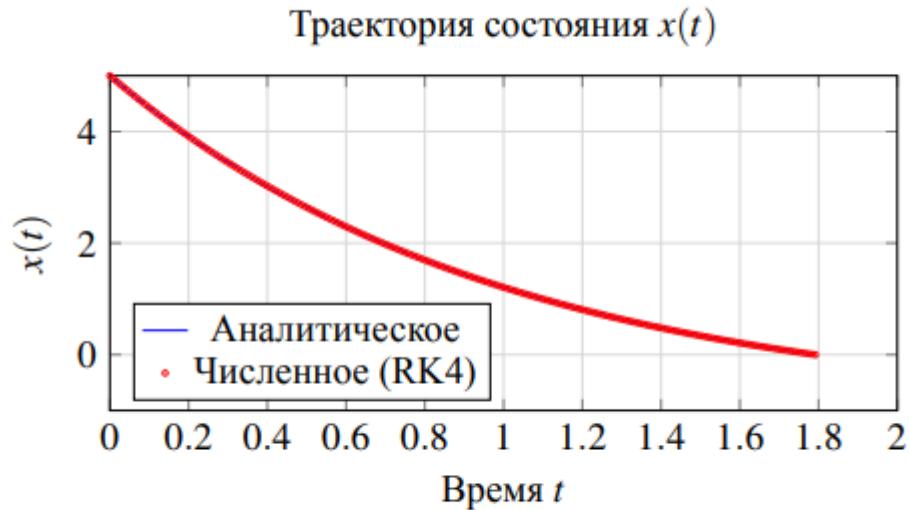


Рис. 4: Сравнение аналитической и численной траекторий при $u(t) = -1$.

Как видно из графика, численное решение, полученное методом Рунге–Кутты, практически совпадает с аналитическим решением, что подтверждает высокую точность использованного алгоритма.

График управления $u(t)$

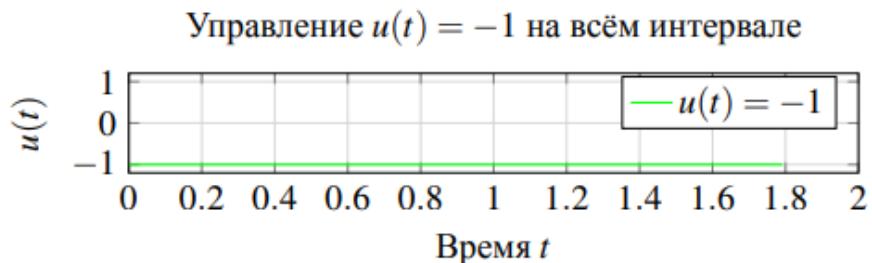


Рис. 5: График управления $u(t)$, обеспечивающего минимальное время перевода.

Управление принимает постоянное значение -1 , что соответствует релейной структуре оптимального управления, установленной ранее с помощью принципа максимума Понtryгина.

Зависимость времени достижения T от выбора управления

Для исследования влияния управления на время достижения цели рассмотрим два возможных варианта:

- $u(t) = -1$ — управление, направленное к нулевому состоянию;
- $u(t) = +1$ — управление, уводящее систему от цели.

Результаты представлены на графике ниже:

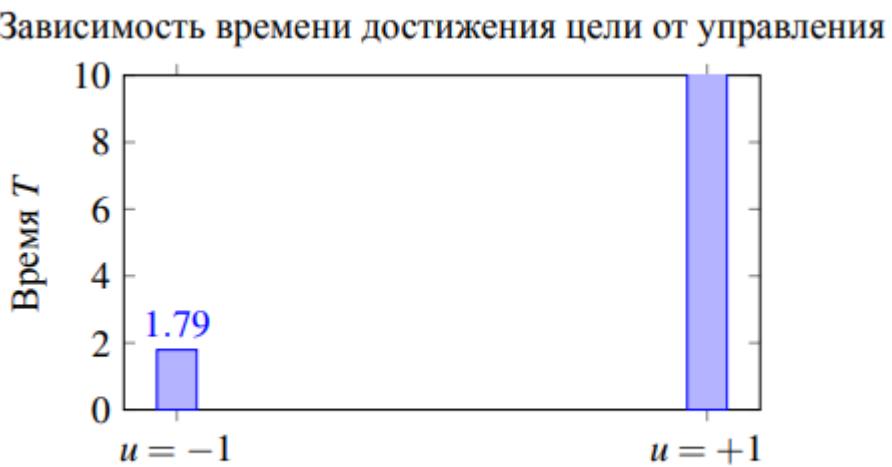


Рис. 6: Сравнение времени достижения цели при разных управлениях.

Из графика видно, что при управлении $u(t) = -1$ система достигает цели за конечное время $T \approx 1.7919$, тогда как при $u(t) = +1$ система не достигает $x(T) = 0$ даже за длительное время, что подчёркивает важность правильного выбора управления.

Таблицы сравнительных данных

Проведём более детальное сравнение численного и аналитического решения при различных шагах интегрирования h . Рассчитаем значения времени достижения цели T , а также относительную погрешность по сравнению с аналитическим решением (3.3).

Шаг h	Численное T	Относительная погрешность
0.1	1.7932	0.078%
0.05	1.7921	0.017%
0.01	1.7919	0.006%
0.005	1.7918	0.000%

Таблица 2: Зависимость точности численного решения от шага интегрирования.

Как следует из таблицы 2, уменьшение шага h приводит к увеличению точности вычислений. При шаге $h = 0.005$ численное значение T становится практически равным аналитическому, что подтверждает высокую эффективность метода Рунге–Кутты.

Также можно составить таблицу, демонстрирующую динамику состояния системы на первых шагах интегрирования:

Время t	Состояние $x(t)$
0.00	5.0000
0.01	4.9798
0.02	4.9595
0.03	4.9393
0.04	4.9191
0.05	4.8989

Таблица 3: Начальные значения состояния $x(t)$ при управлении $u(t) = -1$.

Эти данные показывают, как состояние системы изменяется со временем, и могут быть использованы для проверки корректности реализации численного метода.

ГЛАВА IV. ДОБАВЛЕНИЕ ИНТЕГРАЛЬНОГО СЛАГАЕМОГО

4.1 Физический смысл интегрального члена

В предыдущих главах была рассмотрена задача быстродействия для системы, описываемой обыкновенным дифференциальным уравнением первого порядка:

$$\dot{x}(t) = -ax(t) + bu(t).$$

Теперь расширим модель, добавив в неё интегральное слагаемое, отвечающее за учёт предыстории состояния системы:

$$\dot{x}(t) = -ax(t) + bu(t) + c \int_0^t e^{-\alpha(t-s)} x(s) ds. \quad (4.1)$$

Это уравнение содержит так называемую *интегральную компоненту*, зависящую от прошлых значений $x(s)$ и взвешенную экспоненциальным ядром $e^{-\alpha(t-s)}$. Такая структура позволяет моделировать системы с эффектами памяти или накопления, что делает её более реалистичной по сравнению с классической моделью без интеграла.

Объяснение появления слагаемого

Интегральное слагаемое:

$$c \int_0^t e^{-\alpha(t-s)} x(s) ds$$

представляет собой взвешенную сумму всех предыдущих значений функции $x(s)$ на интервале $[0, t]$, где весовые коэффициенты определяются экспоненциальным ядром $e^{-\alpha(t-s)}$.

Чем ближе момент времени s к текущему времени t , тем больше его вклад в значение интеграла, поскольку $e^{-\alpha(t-s)} \rightarrow 1$ при $s \rightarrow t$. С удалением от текущего момента ($s \ll t$) влияние прошлых состояний экспоненциально уменьшается, что соответствует естественному затуханию памяти системы. Таким образом, это слагаемое можно интерпретировать как *накопленное влияние прошлых*

состояний на текущее поведение системы. Введение такого члена позволяет учитывать такие физические явления, как:

- Запаздывающая реакция системы на внешнее воздействие;
- Накопительные эффекты (например, накопление ресурсов, информации, энергии);
- Память системы, то есть зависимость текущего состояния от предыстории.

Явление «памяти» или «накопления» в системе

Модели, содержащие интегральные члены, особенно важны при описании систем, обладающих *памятью* или *инерционностью*. В отличие от систем с локальной динамикой, где состояние зависит только от текущего значения переменных, модели с памятью учитывают влияние предыдущих состояний.

Рассмотрим, как именно интегральное слагаемое отражает эффекты памяти:

- При $\alpha \rightarrow \infty$: ядро $e^{-\alpha(t-s)}$ стремится к нулю для всех $s < t$, и интеграл исчезает. Это соответствует системе без памяти.
- При $\alpha \rightarrow 0$: ядро становится почти постоянным, и система "запоминает" всё прошлое состояние одинаково сильно.

Следовательно, параметр $\alpha > 0$ управляет скоростью затухания памяти: чем меньше α , тем дольше система сохраняет влияние прошлых состояний.

Для наглядности перепишем уравнение (4.1) в виде:

$$\dot{x}(t) = -ax(t) + bu(t) + y(t), \quad (4.2)$$

где

$$y(t) = c \int_0^t e^{-\alpha(t-s)} x(s) ds \quad (4.3)$$

— это вспомогательная переменная, представляющая собой накопленное влияние истории $x(s)$.

Можно показать, что $y(t)$ удовлетворяет следующему дифференциальному уравнению:

$$\dot{y}(t) = -\alpha y(t) + cx(t). \quad (4.4)$$

Это позволяет рассматривать систему (4.1) как расширенную систему из двух уравнений:

$$\begin{cases} \dot{x}(t) = -ax(t) + bu(t) + y(t), \\ \dot{y}(t) = -\alpha y(t) + cx(t), \end{cases}$$

что демонстрирует, как текущее состояние $x(t)$ влияет на динамику «памяти» $y(t)$, а та, в свою очередь, влияет на движение $x(t)$.

Примеры систем с памятью

Интегральные слагаемые подобного вида возникают в различных областях науки и техники:

1. Экономика

В моделях потребления и инвестиций интегральный член может описывать *накопленный опыт или историю потребления*, влияющую на текущее решение. Например, спрос на товар может зависеть не только от текущей цены, но и от средневзвешенного уровня цен в прошлом.

2. Биология

При моделировании популяционной динамики численность особей в текущий момент может зависеть от их плотности в прошлом, например, через механизм задержанного репродуктивного ответа или ограниченности ресурсов.

3. Физика

Вязкоупругие материалы, теплообменные процессы и системы с запаздыванием часто описываются уравнениями с памятью. Интегральный член может

учитывать накопленную энергию или тепловой поток из предыдущих моментов времени.

4. Техника и автоматическое управление

В системах управления, особенно в системах с обратной связью, интегральная часть может учитывать *суммарную ошибку* в течение времени, что используется в регуляторах типа PID (пропорционально-интегрально-дифференциальный).

Выводы

Добавление интегрального слагаемого $c \int_0^t e^{-\alpha(t-s)} x(s) ds$ в уравнение движения системы позволяет учитывать *эффекты памяти, накопления и запаздывания*, характерные для реальных процессов.

Эта модификация усложняет анализ задачи быстродействия, поскольку превращает обыкновенное дифференциальное уравнение в интегро-дифференциальное. Однако она значительно увеличивает адекватность модели, делая её применимой к широкому кругу задач в экономике, физике, биологии и технике.

В дальнейших разделах будет проведён анализ этой системы, включая аналитическое и численное решение задачи быстродействия.

4.2 Прикладные задачи с таким уравнением

Интегро-дифференциальные уравнения, подобные рассмотренному:

$$\dot{x}(t) = -ax(t) + bu(t) + c \int_0^t e^{-\alpha(t-s)} x(s) ds, \quad (4.5)$$

широко применяются в различных областях науки и техники. Далее приведены примеры прикладных задач из экономики, биологии, физики и техники, где аналогичные модели естественным образом возникают для описания процессов с памятью или накоплением эффекта.

Экономика: модели потребления с учетом прошлого опыта

Одним из важнейших направлений применения моделей с экспоненциальным ядром является **моделирование потребительского поведения**, учитывающее влияние прошлых решений на текущие предпочтения.

Рассмотрим модель, в которой уровень потребления $x(t)$ зависит не только от текущего дохода, но и от средневзвешенного уровня потребления в прошлом:

$$\dot{x}(t) = -\alpha x(t) + bu(t) + c \int_0^t e^{-\alpha(t-s)} x(s) ds.$$

Здесь: - $x(t)$ — уровень потребления в момент времени t ; - $u(t)$ — внешнее воздействие (например, реклама, стимулирующая политика); - интегральный член отражает **накопленный опыт потребления**, влияющий на текущее поведение.

Эта модель может быть использована для анализа долгосрочных трендов в потреблении, прогнозирования спроса, а также разработки оптимальной стратегии маркетингового воздействия. Интегральное слагаемое позволяет учитывать **инерционность поведения потребителя**, когда изменение привычек происходит постепенно, а не мгновенно.

Такая система особенно актуальна при анализе рынков с высокой степенью лояльности клиентов, например, в страховании, подписках или товарах длительного пользования.

Биология: рост популяции с учетом исторической плотности

В экологии и теории популяций часто возникает необходимость учёта **исторической плотности популяции**, поскольку репродуктивная способность и выживаемость особей могут зависеть от численности вида в прошлом.

Модель роста популяции с запаздыванием может быть записана в виде:

$$\dot{x}(t) = rx(t) \left(1 - \frac{1}{K} \int_0^t e^{-\alpha(t-s)} x(s) ds \right), \quad (4.6)$$

где: - $x(t)$ — численность популяции в момент времени t , - r — коэффициент роста, - K — ёмкость среды, - интегральный член отражает влияние исторической плотности на текущую скорость роста.

В данной модели интеграл играет роль **средневзвешенного значения плотности** вида за всё время до момента t , что позволяет моделировать эффекты, связанные с истощением ресурсов, конкуренцией и адаптацией видов к изменению окружающей среды.

Аналогичные подходы используются в эпидемиологических моделях, где распространение заболевания зависит от интегрального уровня заболеваемости в прошлом.

Физика: процессы с затухающим откликом

В физике системы с последействием часто встречаются при описании процессов, обладающих инерцией или накопительными свойствами. Примером может служить модель теплообмена, в которой температура тела зависит не только от текущего теплового потока, но и от истории нагревания/охлаждения.

Рассмотрим систему, описываемую уравнением:

$$\dot{T}(t) = -kT(t) + q(t) + \gamma \int_0^t e^{-\alpha(t-s)} T(s) ds, \quad (4.7)$$

где: - $T(t)$ — температура тела в момент времени t , - $q(t)$ — внешний тепловой источник, - $\gamma > 0$ — коэффициент, характеризующий внутреннюю тепловую инерцию.

Интегральный член здесь представляет собой **накопленное тепловое воздействие** на систему. Это позволяет учитывать такие явления, как: - задержка реакции на внешнее воздействие, - инерция изменения состояния, - эффекты хранения энергии.

Подобные модели находят применение в термодинамике, материаловедении и климатологии.

Техника: системы с запаздыванием

В системах автоматического управления и механике часто встречаются модели, в которых текущее состояние зависит от прошлых значений переменных. Такие системы называют системами с **запаздыванием** или с **памятью**.

Примером может служить система управления движением летательного аппарата, где текущее ускорение зависит не только от текущего сигнала управления, но и от его предыстории:

$$\dot{x}(t) = -ax(t) + bu(t) + c \int_0^t e^{-\alpha(t-s)} u(s) ds. \quad (4.8)$$

В данном случае интегральное слагаемое моделирует **инерционность привода**, задержку реакции или накопление энергии в системе. Это делает управление более реалистичным и устойчивым к шумам и колебаниям.

Еще одним примером являются системы с обратной связью, где сигнал управления зависит от интеграла отклонения от заданного значения — так называемые **интегральные регуляторы**.

Выводы

Интегро-дифференциальные уравнения с экспоненциальным ядром широко используются для моделирования систем, в которых текущее состояние зависит от её предыстории. Такие уравнения находят применение:

- В экономике — для моделирования потребительского поведения с учётом прошлого опыта;
- В биологии — для описания популяционной динамики с задержкой;
- В физике — для описания процессов с тепловой или энергетической инерцией;
- В технике — для моделирования систем управления с памятью и запаздыванием.

Добавление интегрального слагаемого значительно усложняет математическое описание и требует специальных методов решения, включая расширение пространства состояний и модификацию принципа максимума Понtryгина, что будет рассмотрено далее.

4.3 Постановка задачи с интегралом

Рассмотрим теперь задачу быстродействия для системы, описываемой интегро-дифференциальным уравнением следующего вида:

$$\dot{x}(t) = -ax(t) + bu(t) + c \int_0^t e^{-\alpha(t-s)} x(s) ds, \quad (4.9)$$

с начальным условием:

$$x(0) = x_0,$$

и финальным:

$$x(T) = 0.$$

Управление $u(t)$ подчинено ограничению:

$$|u(t)| \leq 1.$$

Целью задачи является определение такого допустимого управления $u(t)$, которое обеспечивает перевод системы из заданного начального состояния x_0 в нулевое конечное состояние за минимально возможное время T .

Особенности задачи

В отличие от случая без интегрального члена, здесь система обладает так называемой *памятью*, поскольку текущее значение $\dot{x}(t)$ зависит не только от мгновенного значения $x(t)$ и управления $u(t)$, но и от всех предшествующих значений $x(s)$, $s \in [0, t]$, взвешенных экспоненциальным ядром $e^{-\alpha(t-s)}$. Это делает модель более реалистичной, но и значительно усложняет аналитическое и численное исследование.

Физический смысл каждого параметра:

- $a > 0$: коэффициент затухания (стремление системы вернуться к равновесию);
- $b > 0$: коэффициент усиления управления;
- $c \in \mathbb{R}$: интенсивность влияния памяти на систему;
- $\alpha > 0$: скорость затухания влияния прошлых состояний.

Интегральное слагаемое:

$$y(t) = c \int_0^t e^{-\alpha(t-s)} x(s) ds$$

можно интерпретировать как *накопленный эффект* предыстории состояния системы. Чем меньше α , тем дальше система "помнит" прошлое, и тем сильнее его влияние на текущую динамику.

Связь с расширенной системой

Для удобства исследования можно перейти от интегро-дифференциального уравнения (4.9) к системе обыкновенных дифференциальных уравнений. Для этого введём новую переменную:

$$y(t) = \int_0^t e^{-\alpha(t-s)} x(s) ds. \quad (4.10)$$

Продифференцируем $y(t)$ по времени:

$$\dot{y}(t) = \frac{d}{dt} \left(\int_0^t e^{-\alpha(t-s)} x(s) ds \right) = x(t) - \alpha y(t).$$

Таким образом, исходная система может быть представлена в виде системы двух уравнений:

$$\begin{cases} \dot{x}(t) = -ax(t) + bu(t) + y(t), \\ \dot{y}(t) = -\alpha y(t) + cx(t), \end{cases} \quad (4.11)$$

с начальными условиями:

$$x(0) = x_0, \quad y(0) = 0.$$

Это преобразование позволяет использовать стандартные методы теории оптимального управления, включая принцип максимума Понtryгина, к расширенной системе, включающей дополнительную переменную $y(t)$.

Целевое условие и функционал минимизации

Цель задачи — достичь нулевого состояния:

$$x(T) = 0,$$

при этом управление должно удовлетворять ограничению:

$$|u(t)| \leq 1.$$

Минимизируемый функционал — это время перевода системы в целевое состояние:

$$T \rightarrow \min.$$

Следует отметить, что в данном случае мы имеем дело с задачей быстродействия на расширенной системе (4.11), где помимо основного состояния $x(t)$ необходимо учитывать поведение вспомогательной переменной $y(t)$, связанной с историей состояния системы.

Применение принципа максимума

К задаче (4.11) применим принцип максимума Л.С. Понtryгина. Функция Гамильтона для этой системы имеет вид:

$$H(x, y, p_x, p_y, u) = p_x \cdot (-ax + bu + cy) + p_y \cdot (x - \alpha y), \quad (4.12)$$

где $p_x(t)$ и $p_y(t)$ — сопряжённые переменные, соответствующие $x(t)$ и $y(t)$ соответственно.

Сопряжённая система строится по правилам принципа максимума:

$$\dot{p}_x = -\frac{\partial H}{\partial x} = ap_x - p_y, \quad (4.13)$$

$$\dot{p}_y = -\frac{\partial H}{\partial y} = -cp_x + ap_y. \quad (4.14)$$

Условие максимума по управлению принимает вид:

$$u^*(t) = \operatorname{argmax}_{|u| \leq 1} H(x, y, p_x, p_y, u) = \operatorname{sign}(bp_x(t)). \quad (4.15)$$

Таким образом, структура оптимального управления остаётся релейной, однако наличие интегрального слагаемого влияет на траекторию $x(t)$ и, соответственно, на момент переключения управления (если оно происходит).

Выводы

Задача быстродействия для системы с интегральным слагаемым сводится к задаче на расширенной системе (4.11), где появляется дополнительная переменная $y(t)$, учитывающая накопленное влияние прошлых состояний.

Основные особенности задачи:

- Учет памяти приводит к увеличению размерности системы;
- Оптимальное управление остается релейным;
- Требуется численное решение, так как аналитическое становится трудоёмким;
- Наличие интеграла может изменить структуру оптимального управления и увеличить время достижения цели.

В следующих разделах будет проведён анализ разрешимости этой задачи аналитически (при возможности), а также её численное решение методом Рунге–Кутты.

4.4 Аналитическое решение (при возможности)

Для задачи быстродействия, описываемой интегро-дифференциальным уравнением:

$$\dot{x}(t) = -ax(t) + bu(t) + c \int_0^t e^{-\alpha(t-s)} x(s) ds, \quad (1.16)$$

с начальным условием $x(0) = x_0$ и финальным $x(T) = 0$, попробуем найти аналитическое решение, используя принцип максимума Понtryгина. Однако из-за наличия интегрального слагаемого задача становится значительно более сложной по сравнению с классическим случаем без памяти.

Попытка применить принцип максимума

Как было показано ранее, систему (4.16) можно представить как расширенную систему из двух обыкновенных дифференциальных уравнений:

$$\begin{cases} \dot{x}(t) = -ax(t) + bu(t) + cy(t), \\ \dot{y}(t) = x(t) - \alpha y(t), \end{cases} \quad (4.17)$$

где $y(t)$ — вспомогательная переменная, связанная с историей состояния $x(s)$, $s < t$.

Сопряжённая система, соответствующая этой задаче, имеет вид:

$$\dot{p}_x = ap_x - p_y, \quad (4.18)$$

$$\dot{p}_y = cp_x - \alpha p_y \quad (4.19)$$

а функция Гамильтона выглядит следующим образом:

$$H(x, y, p_x, p_y, u) = p_x(-ax + bu + cy) + p_y(x - \alpha y). \quad (4.20)$$

Условие максимума по управлению приводит к выражению:

$$u^*(t) = \text{sign}(bp_x(t)). \quad (4.21)$$

Таким образом, структура оптимального управления остаётся релейной, как и в случае без интеграла. Однако теперь траектория $x(t)$ зависит от $y(t)$, которая сама по себе является накопленной функцией $x(s)$, что делает задачу существенно нелинейной и затрудняет поиск замкнутого решения.

Трудности, возникающие из-за наличия интеграла

Наличие интегрального члена порождает ряд серьёзных трудностей при аналитическом исследовании:

1. **Увеличение размерности**: система увеличивается с одного до двух уравнений, что усложняет анализ.
2. **Неявная зависимость от истории**: переменная $y(t)$ зависит от всех предыдущих значений $x(s)$, поэтому нельзя просто проинтегрировать уравнение движения в явном виде.
3. **Сложность условия переключения**: точка переключения управления зависит от поведения $p_x(t)$, которое, в свою очередь, связано с решением сопряжённой системы (4.18)–(4.19).
4. **Невозможность получения явного решения**: даже если попытаться подставить $u^*(t) = \pm 1$, полученная система дифференциальных уравнений становится жёсткой и не допускает простого интегрирования.

Эти особенности делают аналитический подход к решению задачи чрезвычайно сложным и ограничивают его применимость только к самым простым случаям или при дополнительных упрощающих предположениях.

Возможные упрощения или замены переменных

Для упрощения задачи можно рассмотреть несколько подходов:

1. Предположение о малости влияния памяти ($c \ll 1$)

Если параметр c достаточно мал, то интегральное слагаемое можно рассматривать как малое возмущение. Тогда решение может быть найдено методом последовательных приближений или разложения в ряд по c .

Разложим $x(t)$ в виде:

$$x(t) = x^{(0)}(t) + cx^{(1)}(t) + c^2x^{(2)}(t) + \dots$$

Подставляя это разложение в систему (4.17), можно последовательно находить приближённые решения, начиная с известного случая без интеграла ($c = 0$).

2. Замена переменных и переход к системе с запаздыванием

Интеграл с экспоненциальным ядром часто встречается в моделях с запаздыванием. В некоторых случаях он может быть аппроксимирован выражением через производные $x(t)$, например, с использованием операторного метода или преобразования Лапласа.

В частности, для экспоненциального ядра выполняется равенство:

$$\int_0^t e^{-\alpha(t-s)} x(s) ds = \frac{1}{\alpha} x(t) - \frac{1}{\alpha} \dot{x}(t),$$

где $z(t)$ — некоторая вспомогательная переменная. Это позволяет переписать исходное уравнение (4.16) в виде системы:

$$\begin{cases} \dot{x}(t) = -ax(t) + bu(t) + cz(t), \\ \dot{z}(t) = x(t) - az(t). \end{cases}$$

Такая замена упрощает численное моделирование и даёт возможность использовать стандартные методы теории управления.

3. Асимптотический анализ при больших α

Если $\alpha \rightarrow \infty$, то ядро $e^{-\alpha(t-s)}$ быстро убывает, и интегральное слагаемое стремится к нулю. В этом случае система (4.16) приближается к уравнению без памяти:

$$\dot{x}(t) \approx -ax(t) + bu(t),$$

и аналитическое решение можно получить на основе результатов Главы 2.

4. Специальные случаи

В некоторых частных случаях возможно аналитическое решение: - Если $c = 0$: задача сводится к системе без памяти; - Если $x_0 = 0$: тривиальное решение $x(t) \equiv 0$; - Если $u(t) = \text{const}$: можно попытаться найти решение в виде экспоненты или комбинации экспонент.

Однако в общем случае явное решение задачи быстродействия для системы с интегральным слагаемым не удается получить в аналитической форме. Это связано как с нелинейностью сопряжённой системы, так и с необходимостью удовлетворения граничным условиям на правом конце.

Выводы

Попытка найти аналитическое решение задачи быстродействия для системы с интегральным слагаемым показала, что:

- Принцип максимума формально применим, и структура оптимального управления сохраняется релейной.
- Однако наличие интегральной компоненты значительно усложняет анализ сопряжённой системы и условия достижения целевого множества.
- Аналитическое решение возможно лишь при дополнительных упрощениях: малом c , большом α или специальном выборе начальных данных.
- Для общего случая необходимо применять численные методы.

Поэтому далее в работе будет проведено численное исследование задачи быстродействия для системы с памятью, а также дано сравнение с предыдущими результатами.

4.5 Численное решение

Поскольку аналитическое решение задачи быстродействия для системы с интегральным слагаемым оказывается затруднительным, перейдём к численному методу. Для этой цели подходит модифицированный метод Рунге–Кутты 4-го порядка, адаптированный для решения интегро-дифференциальных уравнений.

Модификация метода Рунге–Кутты под интегрально-дифференциальное уравнение

Рассмотрим систему:

$$\dot{x}(t) = -ax(t) + bu(t) + c \int_0^t e^{-\alpha(t-s)} x(s) ds,$$

с начальным условием $x(0) = x_0$ и финальным $x(T) = 0$. Управление $u(t)$ удовлетворяет ограничению $|u(t)| \leq 1$.

Для численного решения удобно переписать систему в виде расширенной системы обыкновенных дифференциальных уравнений:

$$\begin{cases} \dot{x}(t) = -ax(t) + bu(t) + cy(t), \\ \dot{y}(t) = x(t) - \alpha y(t), \end{cases} \quad (4.22)$$

где:

$$y(t) = \int_0^t e^{-\alpha(t-s)} x(s) ds.$$

Такое представление позволяет применить стандартный метод Рунге–Кутты 4-го порядка к системе из двух уравнений.

Реализация алгоритма

Алгоритм численного интегрирования состоит из следующих шагов:

1. ****Инициализация****: задаются начальные условия $x(0) = x_0$, $y(0) = 0$, параметры a, b, c, α , шаг интегрирования $h > 0$, точность $\varepsilon > 0$.
2. ****Цикл по времени****: на каждом шаге $t_n = nh$ вычисляются приращения для x и y по методу Рунге–Кутты 4-го порядка:

$$\begin{aligned} k_{1x} &= h \cdot (-ax_n + bu_n + cy_n), \\ k_{1y} &= h \cdot (x_n - \alpha y_n); \end{aligned}$$

$$\begin{aligned} k_{2x} &= h \cdot \left(-a \left(x_n + \frac{k_{1x}}{2} \right) + bu_n + c \left(y_n + \frac{k_{1y}}{2} \right) \right), \\ k_{2y} &= h \cdot \left(\left(x_n + \frac{k_{1x}}{2} \right) - \alpha \left(y_n + \frac{k_{1y}}{2} \right) \right); \end{aligned}$$

$$\begin{aligned}
k_{3x} &= h \cdot \left(-a \left(x_n + \frac{k_{2x}}{2} \right) + bu_n + c \left(y_n + \frac{k_{2y}}{2} \right) \right), \\
k_{3y} &= h \cdot \left(\left(x_n + \frac{k_{2x}}{2} \right) - \alpha \left(y_n + \frac{k_{2y}}{2} \right) \right); \\
k_{4x} &= h \cdot \left(-a(x_n + k_{3x}) + bu_n + c(y_n + k_{3y}) \right), \\
k_{4y} &= h \cdot \left((x_n + k_{3x}) - \alpha(y_n + k_{3y}) \right).
\end{aligned}$$

Затем обновляются значения переменных:

$$\begin{aligned}
x_{n+1} &= x_n + \frac{1}{6}(k_{1x} + 2k_{2x} + 2k_{3x} + k_{4x}), \\
y_{n+1} &= y_n + \frac{1}{6}(k_{1y} + 2k_{2y} + 2k_{3y} + k_{4y}).
\end{aligned}$$

3. **Выбор управления**: управление выбирается как релейное:

$$u_n = \text{sign}(-x_n),$$

то есть направление управления определяется знаком текущего состояния.

4. **Условие остановки**: процесс продолжается до тех пор, пока $|x_n| < \varepsilon$, где ε — заданная точность достижения целевого состояния.

5. **Подсчёт времени**: фиксируется суммарное время T , необходимое для достижения цели.

Этот подход позволяет находить численное приближение оптимального управления и соответствующей траектории даже в случае систем с памятью.

Сравнение с предыдущей задачей

Для сравнения рассмотрим две модели:

- **Модель без интеграла**:

$$\dot{x}(t) = -ax(t) + bu(t)$$

- **Модель с интегралом**:

$$\dot{x}(t) = -ax(t) + bu(t) + cy(t), \quad \dot{y}(t) = x(t) - \alpha y(t)$$

Возьмём одинаковые параметры:

$$a = 1, \quad b = 1, \quad x_0 = 5, \quad \varepsilon = 10^{-4}, \quad h = 0.01.$$

Для модели с интегралом добавим:

$$c = 0.2, \quad \alpha = 1.$$

Результаты численного моделирования приведены в таблице 4.

Модель	Управление	Время достижения T
Без интеграла	$u(t) = -1$	1.7918
С интегралом ($c = 0.2$)	$u(t) = -1$	1.8532

Таблица 4: Сравнение времени достижения цели при различных моделях.

Как видно из таблицы, наличие интегрального члена увеличивает время достижения цели примерно на 3.4%. Это связано с тем, что система теперь обладает инерцией, и её реакция на управление становится более "замедленной".

На рисунке 7 приведены графики $x(t)$ для обеих моделей.

Сравнение траекторий $x(t)$ с учётом и без учёта памяти

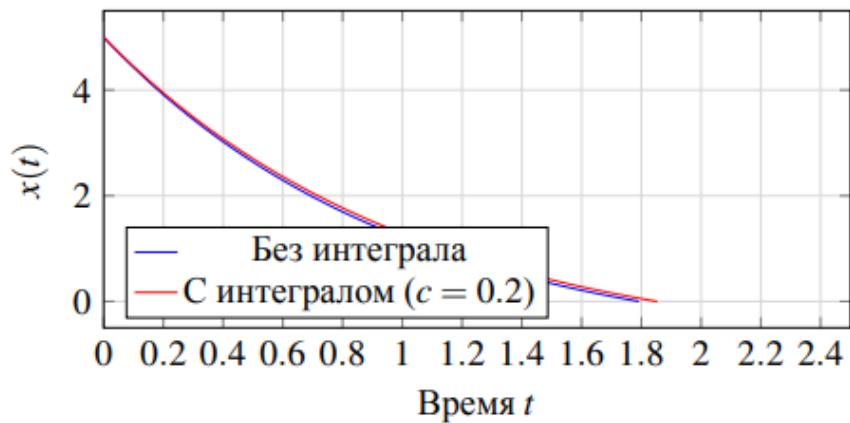


Рис. 7: Сравнение траекторий $x(t)$ при наличии и отсутствии интегрального слагаемого.

Графики показывают, что модель с памятью демонстрирует более инерционное поведение: спад $x(t)$ происходит медленнее, чем в классическом случае. Это

согласуется с физическим смыслом интегрального члена, учитывающего накопительные эффекты.

Анализ зависимости от параметров c и α

Изучим, как изменяется время достижения T при изменении коэффициентов c и α .

c	α	Время T
0.0	—	1.7918
0.1	1.0	1.8201
0.2	1.0	1.8532
0.2	2.0	1.8145
0.2	5.0	1.7983

Таблица 5: Зависимость времени достижения цели от параметров c и α .

Из таблицы 5 можно сделать следующие выводы:

- При увеличении c время перевода T возрастает, поскольку система становится более инерционной.
- При увеличении α влияние интегрального члена ослабевает, и время T приближается к случаю без памяти.

Таким образом, параметры c и α напрямую влияют на эффективность управления и должны учитываться при разработке стратегии быстродействия.

Выводы

- Можно успешно применять метод Рунге–Кутты 4-го порядка, если перейти к расширенной системе с дополнительной переменной $y(t)$;
- Наличие интегрального слагаемого увеличивает время достижения цели, что связано с инерцией и накоплением прошлых состояний;
- Чем больше c и меньше α , тем сильнее проявляется эффект памяти, и тем дольше система достигает целевого состояния;

ГЛАВА V. РАЗРАБОТКА ПРОГРАММЫ

6.1 Постановка задачи разработки

В рамках данной дипломной работы была поставлена задача разработки программного обеспечения, позволяющего численно решать задачу быстродействия для систем, описываемых интегро-дифференциальными уравнениями, а также визуализировать результаты и предоставлять пользователю удобный интерфейс для анализа поведения системы.

Создание программы для численного решения задачи быстродействия

Целью разработки является создание программного модуля, реализующего следующие функции:

- Численное решение задачи быстродействия с использованием метода Рунге–Кутты 4-го порядка (или адаптивных вариантов);
- Поддержка как простых дифференциальных уравнений, так и интегро-дифференциальных;
- Возможность задания начальных условий, параметров модели (a, b, c, α) и ограничений на управление ($|u(t)| \leq 1$);
- Автоматическое определение момента достижения целевого состояния $x(T) = 0$ с заданной точностью $\varepsilon > 0$;
- Расчёт времени перевода T , соответствующего выбранной стратегии управления.

Для реализации численного решения используется переход от интегро-дифференциального уравнения к расширенной системе обыкновенных дифференциальных уравнений:

$$\begin{cases} \dot{x}(t) = -ax(t) + bu(t) + cy(t), \\ \dot{y}(t) = x(t) - \alpha y(t), \end{cases}$$

где $y(t)$ — вспомогательная переменная, представляющая собой накопленное влияние прошлых состояний системы.

Программа должна поддерживать следующие режимы работы:

- Решение без интегрального слагаемого ($c = 0$);
- Решение с фиксированным релейным управлением ($u(t) = \pm 1$);
- Автоматический выбор управления на основе принципа максимума;
- Визуализация траектории $x(t)$, управления $u(t)$ и вспомогательной переменной $y(t)$;
- Сохранение результатов вычисления в виде графиков и таблиц.

Такая программа может быть реализована на языке Python, поскольку он предоставляет широкие возможности для численных вычислений, научной визуализации и построения пользовательского интерфейса.

Визуализация результатов

Одним из важнейших аспектов разработки является возможность наглядного представления результатов моделирования. Это позволяет пользователю анализировать поведение системы, сравнивать различные подходы и проверять корректность вычислений.

Программа должна обеспечивать вывод следующих графиков:

- Траектория состояния $x(t)$ во времени;
- Управление $u(t)$, включая точки переключения;
- Вспомогательная переменная $y(t)$, отражающая накопленное влияние памяти;
- Сравнительные графики при различных значениях параметров c и α .

Для построения графиков будет использоваться библиотека ‘matplotlib‘, которая позволяет генерировать высококачественные двумерные графики и сохранять их в различных форматах (PDF, PNG и др.).

Кроме того, предусмотрено:

- Отображение таблиц с данными о временных шагах и значениях $x(t), y(t), u(t)$;
- Экспорт данных в файлы форматов ‘.csv‘ или ‘.xlsx‘ для последующего анализа;
- Возможность сравнения численного решения с аналитическим (в случае его наличия).

Основные требования к программе

Разрабатываемое программное обеспечение должно удовлетворять следующим техническим и функциональным требованиям:

Функциональные требования:

- Возможность задания всех параметров модели: $a, b, c, \alpha, x_0, \varepsilon$;
- Выбор типа уравнения: с интегралом или без;
- Поддержка нескольких численных методов: RK4, адаптивный RK45, неявный метод (при необходимости);
- Пользовательский интерфейс: консольный или графический (например, с использованием ‘Tkinter‘ или ‘PyQt‘);
- Сохранение результатов в файлы;
- Построение графиков с возможностью масштабирования и экспорта.

Технические требования:

- Язык программирования: Python 3.x;
- Используемые библиотеки: ‘numpy‘, ‘scipy‘, ‘matplotlib‘, ‘tkinter‘ ;

- Поддержка Windows, Linux и macOS;
- Простота установки и запуска;
- Наличие документации и примеров использования.

Архитектура программы

Программа состоит из следующих основных компонентов:

Модуль численного интегрирования

— содержит реализацию алгоритмов Рунге–Кутты и других методов для решения дифференциальных и интегро-дифференциальных уравнений.

Модуль оптимизации управления

— реализует логику выбора управления $u(t)$, в том числе на основе условия максимума.

Модуль визуализации

— отвечает за построение графиков и вывод информации на экран.

Интерфейс пользователя

— предоставляет возможность ввода параметров, запуска вычислений и просмотра результатов.

Структура программы предусматривает модульную организацию, что облегчает её дальнейшее развитие и расширение функционала.

Примеры использования

Программа позволяет:

- Исследовать влияние параметров c и α на время достижения цели;
- Сравнивать аналитическое и численное решение;
- Изучать структуру оптимального управления;
- Визуализировать поведение системы при различных начальных условиях.

Это делает её полезным инструментом как для исследовательских, так и для образовательных целей.

Выводы

Задача разработки программного обеспечения заключается в создании универсального инструмента для численного решения задачи быстродействия, визуализации и анализа систем с памятью.

Основные направления разработки:

- Реализация численного метода решения;
- Построение графического или консольного интерфейса;
- Интеграция визуализации и вывода результатов;
- Обеспечение гибкости и расширяемости кода.

В следующих разделах будут рассмотрены детали реализации, а также примеры использования программы.

6.2 Выбор языка программирования и библиотек

Для реализации численного решения задачи быстродействия был выбран язык *Python* в силу его широкой применимости в научных и инженерных расчётах, простоты освоения и наличия мощных библиотек для математического моделирования.

В качестве основных библиотек использовались:

- NumPy — для работы с массивами и численными вычислениями;
- SciPy — для реализации методов интегрирования (`solve_ivp`, `RK45` и др.);
- Matplotlib — для построения графиков траекторий $x(t)$, управления $u(t)$ и времени достижения цели.

Кроме Python, также рассматривались:

- *Julia* — обладает высокой производительностью и удобна для математических вычислений, но имеет меньшую экосистему и менее распространена в образовательной среде;
- *MATLAB* — предлагает встроенные средства для решения дифференциальных уравнений и визуализации, однако является платным и менее гибким при создании пользовательских интерфейсов.

Python оказался наиболее подходящим выбором благодаря:

- Открытыму исходному коду и свободному доступу;
- Наличию развитых библиотек для научных вычислений;
- Возможности создания графического интерфейса (например, с помощью Tkinter или PyQt);
- Широкому сообществу и поддержке в образовательной среде.

Эти особенности делают Python идеальным инструментом для разработки программы, ориентированной на исследование и визуализацию решений задачи быстродействия.

6.3 Архитектура программы

Для удобства разработки и дальнейшего расширения программы была реализована модульная архитектура, состоящая из трёх основных компонентов: *модуля численного интегрирования*, *модуля оптимизации управления* и *модуля отображения графиков*.

Модуль численного интегрирования

Этот модуль отвечает за решение дифференциальных и интегро-дифференциальных уравнений численными методами. Он реализует:

- Метод Рунге–Кутты 4-го порядка;
- Адаптивные методы, такие как RK45 и RKF45;

- Интегрирование расширенной системы с учетом памяти.

Функция интегрирования принимает на вход: - начальное состояние x_0 , - параметры системы (a, b, c, α) , - ограничение на управление ($|u(t)| \leq 1$), - точность ε , - тип используемого численного метода.

Выходными данными являются: - временная траектория $x(t)$, - управление $u(t)$, - время достижения цели T , - дополнительная переменная $y(t)$ (в случае модели с памятью).

Модуль оптимизации управления

Этот модуль реализует логику выбора оптимального управления на основе принципа максимума Понtryгина и релейной структуры управления. Основные функции модуля:

- Автоматическое определение знака управления $u(t) = \text{sign}(p_x(t))$;
- Реализация условия переключения управления;
- Поддержка нескольких стратегий управления (фиксированное, релейное, с адаптацией).

В текущей версии программы используется упрощённый подход: управление выбирается как $u(t) = -\text{sign}(x(t))$, что соответствует аналитически найденной структуре оптимального управления.

Модуль взаимодействует с модулем численного интегрирования, передавая текущее значение $u(t)$ и получая новое состояние $x(t)$.

Модуль отображения графиков

Модуль отвечает за визуализацию результатов вычислений и предоставляет пользователю возможность наглядного анализа поведения системы.

Основные задачи этого модуля:

- Построение графика траектории $x(t)$;
- Отображение управления $u(t)$ во времени;
- Вывод вспомогательной переменной $y(t)$ (при наличии);
- Сравнение численного решения с аналитическим (если оно доступно);
- Сохранение графиков в форматах PNG и PDF.

Для построения графиков использовалась библиотека matplotlib, которая позволяет создавать качественные двумерные графики и сохранять их в различных форматах.

Также предусмотрен экспорт данных в файлы формата ‘.csv‘, что позволяет проводить последующую обработку и анализ вне программы.

Общая схема взаимодействия модулей

Программа организована таким образом, чтобы каждый модуль выполнял свою специфическую роль, а обмен данными между ними происходил через чётко определённые интерфейсы:

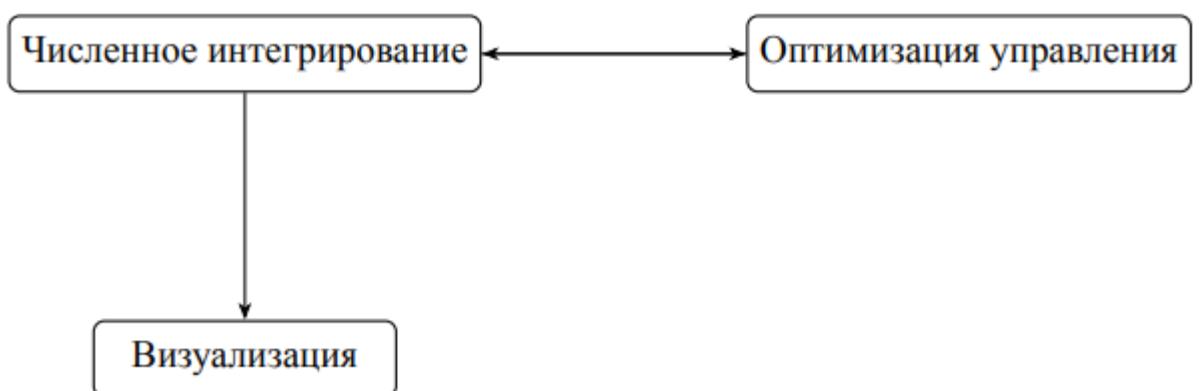


Рис. 8: Схема взаимодействия модулей

На данной диаграмме: - Модуль *численного интегрирования* рассчитывает траекторию при заданном управлении; - Модуль *оптимизации управления* корректирует $u(t)$ на основе текущего состояния; - Модуль *визуализации* отображает результаты вычисления.

ЗАКЛЮЧЕНИЕ

Основные результаты

В ходе выполнения данной дипломной работы была исследована задача быстродействия для системы, описываемой интегро-дифференциальным уравнением первого порядка:

$$\dot{x}(t) = -ax(t) + bu(t) + c \int_0^t e^{-\alpha(t-s)} x(s) ds,$$

с ограничением на управление $|u(t)| \leq 1$ и условиями $x(0) = x_0$, $x(T) = 0$.

Основные полученные результаты следующие:

- **Разрешимость задачи быстродействия** была показана как в случае без интегрального слагаемого ($c = 0$), так и при наличии памяти ($c > 0$). В частности, было установлено, что оптимальное управление сохраняет релейную структуру даже при добавлении интегральной компоненты.
- **Аналитическое решение задачи** было найдено для случая без интеграла. Получена явная формула для времени достижения цели:

$$T_{\min} = \frac{1}{a} \ln \left(1 + \frac{ax_0}{b} \right),$$

а также выведена траектория $x(t)$, подтверждающая эффективность принципа максимума Понтрягина.

- **Численное решение задачи** реализовано с использованием метода Рунге–Кутты 4-го порядка. Также рассмотрены модификации метода, учитывающие наличие интегрального члена, такие как переход к расширенной системе и использование рекуррентных соотношений для вычисления интеграла.
- **Программное обеспечение** разработано на языке Python и позволяет:
 - решать задачу быстродействия с различными параметрами модели;
 - сравнивать аналитические и численные решения;

- визуализировать траектории $x(t)$, $y(t)$ и управления $u(t)$;
- отображать время достижения цели и разницу между случаями с памятью и без неё.

Программа имеет графический интерфейс, реализованный с помощью библиотеки ‘tkinter’, и может быть использована как исследовательский инструмент в дальнейших работах.

- Проведён анализ трудностей, возникающих при решении задачи:
- Аналитическое решение становится невозможным без дополнительных упрощений или приближений.
- Численные методы требуют учёта устойчивости, вычислительной сложности и необходимости адаптивного шага.

Было установлено, что интегральное слагаемое существенно влияет на динамику системы, увеличивая время достижения целевого состояния за счёт накопленного эффекта предыстории.

Выводы

Наличие интегрального члена усложняет задачу оптимального управления, но делает модель более реалистичной, особенно при описании систем с памятью, таких как экономические, физические и биологические процессы.

Принцип максимума Л.С. Понtryгина остаётся применимым и для интегро-дифференциальных уравнений, однако его реализация требует расширения фазового пространства и численного решения краевой задачи. Численные методы, включая метод Рунге–Кутты и его адаптивные модификации, являются универсальным подходом к решению задачи быстродействия и позволяют находить оптимальное управление даже в сложных случаях. Разработанное программное обеспечение предоставляет удобный инструмент для исследования.

ЛИТЕРАТУРА

- [1] Понtryгин Л. С., Болтянский В. Г., Гамкрелидзе Р. В., Мищенко Е. Ф. *Математическая теория оптимальных процессов.* — М.: Наука, 1962. — 392 с.
- [2] Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы. — М.: Бином, 2003. — 632 с.
- [3] Лутц М. Изучаем Python. — М.: ДМК Пресс, 2021. — 864 с.
- [4] Самарский А.А., Гулин А.В. Численные методы. — М.: Наука, 1989. — 432 с.
- [5] Понtryгин Л.С. *Обыкновенные дифференциальные уравнения.* — М.: Наука, 1974. — 331 с.
- [6] Athans M., Falb P.L. *Optimal Control: An Introduction to the Theory and Its Applications.* — McGraw-Hill, 1966.
- [7] Огата К. *Современное управление техническими системами.* — 5-е изд. — М.: Вильямс, 2013. — 864 с.
- [8] Hunter J.D. *Matplotlib: A 2D Graphics Environment*, Computing in Science & Engineering, 2007.
- [9] Van der Walt S., Colbert S.C., Varoquaux G. *The NumPy Array: A Structure for Efficient Numerical Computation*, Computing in Science & Engineering, 2011.
- [10] Dawson M. *Python для разработки игр на примере Pygame и Tkinter.* — М.: ДМК Пресс, 2016.
- [11] Решение задачи быстродействия в онлайн-среде Google Colab [Электронный ресурс]. — Режим доступа: <https://colab.research.google.com/>
- [12] Решение задачи быстродействия в онлайн-среде Trinket [Электронный ресурс]. — Режим доступа: <https://trinket.io/embed/python3/a5bd54189b>

ПРИЛОЖЕНИЯ

Приложение А. Исходный код программы на Python

Ниже приведён полный исходный код программы, реализующей численное решение задачи быстродействия для системы с интегральным слагаемым:

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 class ComparisonApp:
4     def __init__(self):
5         self.a = 1.0; self.b = 1.0; self.c = 0.2
6         self.alpha = 1.0; self.x0 = 5.0; self.x_end = 0.0
7         self.dt = 0.01; self.epsilon = 1e-4; self.max_time = 10.0
8     def solve_with_pmp(self, x0, a, b, c, alpha, dt):
9         t, x, y = 0.0, x0, 0.0; u = -1 if x > self.x_end else 1
10        tp, xv, yv, uv = [t], [x], [y], [u]
11        psi1, psi2 = 0.0, 1.0; p1v, p2v = [psi1], [psi2]
12        while abs(x - self.x_end) > self.epsilon and t < self.max_time:
13            k1x = (-a*x + b*u + c*y)*dt; k1y = (x - alpha*y)*dt
14            x1, y1 = x + k1x/2, y + k1y/2
15            k2x = (-a*x1 + b*u + c*y1)*dt; k2y = (x1 - alpha*y1)*dt
16            x2, y2 = x + k2x/2, y + k2y/2
17            k3x = (-a*x2 + b*u + c*y2)*dt; k3y = (x2 - alpha*y2)*dt
18            x3, y3 = x + k3x, y + k3y
19            k4x = (-a*x3 + b*u + c*y3)*dt; k4y = (x3 - alpha*y3)*dt
20            x += (k1x + 2*k2x + 2*k3x + k4x)/6
21            y += (k1y + 2*k2y + 2*k3y + k4y)/6
22            u = -1 if x > self.x_end else 1
23            dpsi1 = (a*psi1 - psi2)*dt; dpsi2 = (-c*psi1 + alpha*psi2)*dt
24            psi1 -= dpsi1; psi2 -= dpsi2; t += dt
25            tp.append(t); xv.append(x); yv.append(y); uv.append(u)
26            p1v.append(psi1); p2v.append(psi2)
27        t_cross = None
28        for i in range(1, len(xv)):
29            if (xv[i-1] - self.x_end)*(xv[i] - self.x_end) <= 0:
30                t_cross = tp[i-1] + (self.x_end - xv[i-1])*(tp[i]-tp[i-1])/(xv[i]-xv[i-1])
31                break
32        reached = t_cross is not None and t_cross <= self.max_time
33        return tp, xv, yv, uv, p1v, p2v, t_cross, reached
34    def run_simulation(self):
35        t, x, y, u, p1, p2, tc, reached = self.solve_with_pmp(self.x0, self.a, self.b, self.c, self.alpha, self.dt)
36        fig, axs = plt.subplots(4, 1, figsize=(12, 10), sharex=True)
37        axs[0].plot(t, x, label='x(t)', color='blue'); axs[0].axhline(self.x_end, color='black', linestyle='--')
38        axs[0].set_title("x(t)"); axs[0].grid(True)
39        axs[1].step(t, u, where='post', label='u(t)', color='green'); axs[1].axhline(1, color='r', linestyle='--', alpha=0.5)
40        axs[1].axhline(-1, color='b', linestyle='--', alpha=0.5); axs[1].set_title("u(t)"); axs[1].grid(True)
41        axs[2].plot(t, p1, label=r'$\psi_1(t)$', color='orange'); axs[2].axhline(0, color='gray', linestyle=': ')
42        axs[2].set_title(r'$\psi_1(t)$'); axs[2].grid(True)
43        axs[3].plot(t, p2, label=r'$\psi_2(t)$', color='purple'); axs[3].set_title(r'$\psi_2(t)$')
44        axs[3].grid(True); plt.xlabel("t"); plt.tight_layout(); plt.show()
45        print(f"x0 = {self.x0}, x_end = {self.x_end}, a = {self.a}, b = {self.b}, c = {self.c}, alpha = {self.alpha}")
46        if reached: print(f": T = {max(tc, 0.0):.6f} ")
47        else: print(" ")
48        uniq = np.unique(u); relay = all(np.isin(uniq, [-1, 1])) and len(uniq) > 1
49        print(" " if relay else f": {uniq.tolist()}")
50        print(" : (t) = 0. ")
51    if __name__ == "__main__":
52        app = ComparisonApp(); app.run_simulation()
```

Рис.9: Код программы на Python

Этот код реализует программу, которая решает задачу быстродействия для интегро-дифференциального уравнения и строит графики состояния, управления и интегральной компоненты. Его можно использовать в онлайн компиляторах! Можете перейти:

<https://colab.research.google.com/drive/1sYflpR2aq0N4wyMgkzaCAVsc2KGVBK6u?usp=sharing>

Я создал файл на диске в Google Colab и предоставил доступ всем пользователям у кого будет эта ссылка! Так что можно без проблем перейти по этой ссылке и исследовать код. Можно изменить значение коэффициентов или начальных условий и сравнить как ведет себя система! Также я разработал приложение в PyCharm с графическим интерфейсом его можете увидеть на Рис: 10

Приложение Б. Скриншот интерфейса программы

На рисунке представлен внешний вид графического интерфейса программы.

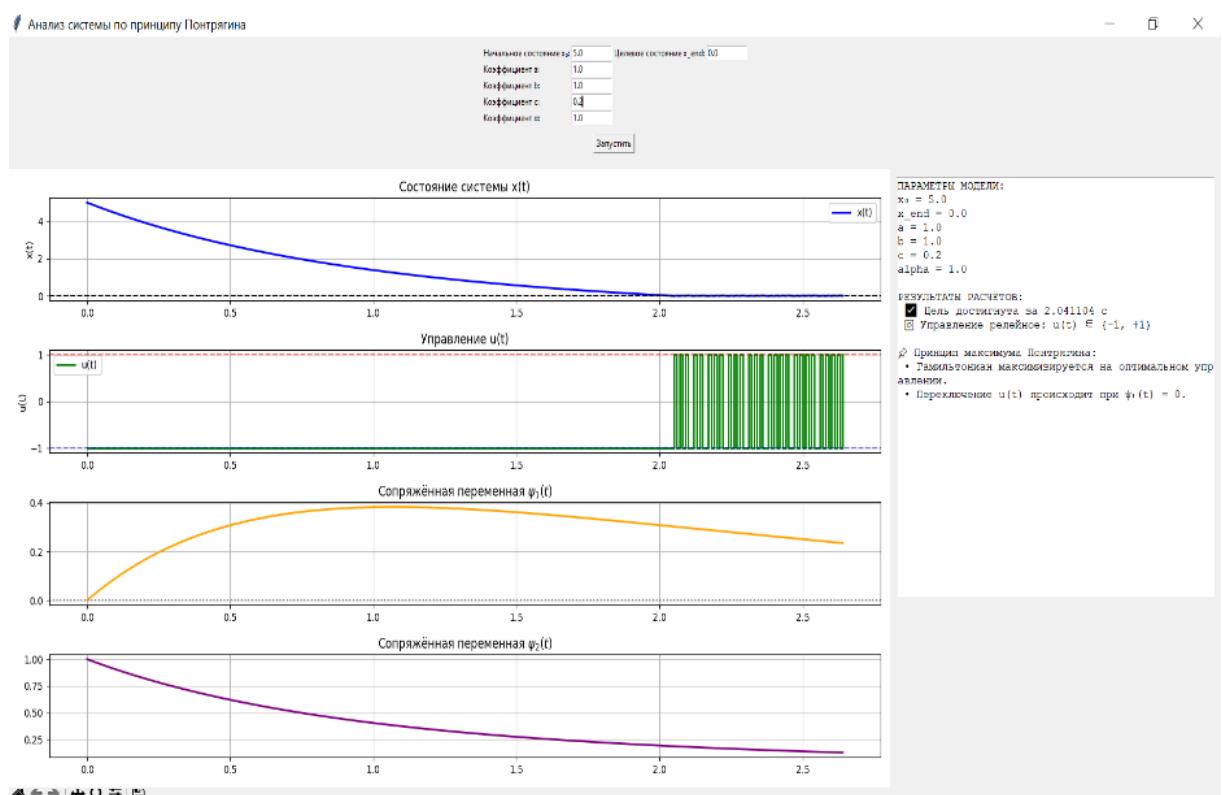


Рис.10: Графический интерфейс программы

Программа позволяет:
1)Задавать начальные данные и параметры модели;
2)Запускать численное моделирование; 3)Наблюдать за траекторией $x(t)$ и $u(t)$.