

Санкт-Петербургский государственный университет

Кафедра теории управления

Попков Александр Сергеевич

Магистерская диссертация

**Модификация адаптивного метода оптимального управления
для некоторых классов нелинейных систем**

Направление 01.04.02

«Прикладная математика и информатика»

Магистерская программа «Методы прикладной математики в задачах управления»

Научный руководитель,
доктор физ.-мат. наук,
профессор
Смирнов Н. В.

Санкт-Петербург

2017

Содержание

Введение	3
Постановка задачи	4
Обзор литературы	6
Глава 1. Линейная задача оптимального управления	8
1.1. Постановка задачи. Сведение к ИЗЛП	8
1.2. Адаптивный метод	11
1.3. Множество достижимости и множество управляемости	14
Глава 2. Нелинейная задача оптимального управления	20
2.1. Метод позиционного управления Габасова	20
2.2. Модификация метода позиционного управления	21
2.3. Обоснование сходимости	25
2.4. Общий алгоритм и программная реализация	29
Глава 3. Примеры приложения модифицированного метода	33
3.1. Управление маятником	33
3.2. Демпфирование колебаний спутника	36
Заключение	40
Список литературы	42
Приложение	44

Введение

Среди современных направлений прикладной математики особое место занимает математическая теория управления. В ней изучаются подходы к нахождению управляющего воздействия на динамический процесс для достижения той или иной цели. В данной диссертации рассматривается частный случай такой задачи: построение терминального управления, переводящего систему из одного заданного состояния в другое, либо в некоторое множество допустимых конечных состояний за заданное время. В случае существования нескольких таких управлений среди них выбирается оптимальное, т. е. доставляющее минимум некоторому заданному функционалу (критерию качества) среди всех допустимых управлений. Поиск управлений осуществляется в специальном классе функций — кусочно-постоянном. Для описания динамического процесса используются системы обыкновенных дифференциальных уравнений.

Кроме того, большой интерес представляет нахождение управления в режиме реального времени: управление строится одновременно с движением системы и учитывает изменяющиеся внутренние и внешние факторы. Такие задачи имеют широкое распространение в различных сферах: управление количеством топлива, поступающего в двигатель транспортного средства, управление направлением движения беспилотного аппарата, распределение инвестиций в секторы экономики и множество других задач.

Многие процессы управления техническими объектами имеют объективно нелинейный характер, следовательно для их точного моделирования и в последствии управления требуется аппарат построения управления для нелинейных систем.

В теории обыкновенных дифференциальных уравнений известно аналитическое решение задачи Коши для линейных систем любой размерности: выведена формула Коши. Это привело к развитию методов решения линейных задач управления. Однако, аналитические решения для нелинейных систем дифференциальных уравнений существуют лишь для очень узких видов нелинейностей, что делает невозможным изобретение точных методов нахождения управления для систем с нелинейной правой частью. В данной работе описан метод численного построения управления для нелинейных по фазовым переменным систем при помощи многократной линеаризации правой части системы и нахождения точного оптимального управления для такой системы.

Постановка задачи

Пусть динамический процесс описывается системой обыкновенных дифференциальных уравнений следующего вида:

$$\dot{x} = f(x) + Bu. \quad (1)$$

Здесь x — n -мерный вектор фазовых переменных, функций времени $x_1(t), \dots, x_n(t)$; u — r -мерный вектор управлений, функций времени $u_1(t), \dots, u_r(t)$; $f(x)$ — непрерывно-дифференцируемая n -мерная вектор-функция переменной x : $f(x) \in \mathbf{C}^1(\mathbb{R}^n)$, B — матрица размера $n \times r$, \dot{x} — производная вектора x по времени.

Задача терминального управления состоит в переводе системы из состояния x_* в состояние x^* за заданное время T :

$$x(0) = x_*, \quad x(T) = x^*. \quad (2)$$

Каждая компонента вектора управления имеет двустороннее ограничение:

$$l_i^{(1)} \leq u_i(t) \leq l_i^{(2)}, \quad i = \overline{1, r}. \quad (3)$$

При этом управление ищется в классе кусочно-постоянных функций с периодом квантования $h = T/N$, где $N \in \mathbb{N}$:

$$u_i(t) = u_i((k-1)h) = u_{i,k}, \quad t \in [(k-1)h, kh), \quad k = \overline{1, N}, \quad i = \overline{1, r}. \quad (4)$$

Замечание 1. В зависимости от специфики конкретной прикладной задачи число N может быть либо фиксированным (например, в задаче ежеквартального распределения инвестиций $N = \frac{T}{3 \text{ месяца}}$), либо любым натуральным числом не превышающим некоторого предела: $N \leq \overline{N}$ (этим пределом может быть, например, тактовая частота процессора бортового компьютера беспилотника).

Определение 1. Будем называть функцию $u(t)$ *допустимым управлением (решением)*, если она удовлетворяет условиям (3), (4), а функция $x(t)$, являющаяся решением системы (1), замкнутой управлением $u(t)$, удовлетворяет терминальным условиям (2).

Наконец, зададим функционал качества для выбора оптимального управления среди допустимых:

$$\int_0^T \|u(t)\|_1 dt \longrightarrow \min_u, \quad (5)$$

где $\|u(t)\|_1 = \sum_{i=1}^r |u_i(t)|$.

Определение 2. Будем называть функцию $u^0(t)$ *оптимальным управлением (решением)*, если она является допустимым управлением и доставляет минимум функционалу (5) среди всех допустимых управлений.

Главная задача данной работы — разработка численного метода построения решения для задачи (1)–(5), изучение его теоретических свойств. Предполагается верификация метода на контрольных примерах.

Обзор литературы

Одним из основателей математической теории управления является Рудольф Калман, внесший весомый вклад в развитие этого направления [1].

Самым известным методом решения задачи оптимального управления является принцип максимума Понтрягина [2]. Он был сформулирован в 1958 году коллективом ученых под руководством Л. С. Понтрягина в виде необходимых условий оптимального решения. Метод является эффективным для множества частных случаев, но не является универсальным ввиду необходимости решения нелинейных алгебраических уравнений.

Широкой популярностью обладает метод динамического программирования, основанный на принципе оптимальности Беллмана [3].

Существенный вклад в математическую теорию процессов управления и теорию автоматического регулирования внес В. И. Зубов [4].

Основной литературой для данной работы послужили наработки Р. Ф. Габасова и его коллектива. Ими был представлен метод построения оптимального управления для линейных систем путем сведения к интервальной задаче линейного программирования (ИЗЛП) [5], описан эффективный метод решения этой задачи [6]. Был предложен метод управления нелинейными системами, основанный на замене нелинейной функции ее кусочно-линейной аппроксимацией [7].

Метод Габасова для линейного случая был апробирован автором этой работы на нескольких моделях: управление вращательным движением вала электродвигателя [8], управление движением четырехколесной тележки [9], распределение инвестиций в отрасли многопродуктовой экономики [10], и связанная с этой темой статья [11]. В работе [12] решается задача распреде-

ления инвестиций при наличии нескольких целевых функций, использован аппарат многокритериальной оптимизации. Управление для нелинейной модели квадрокоптера построено в статье [13], в работе [14] сформулирован метод управления нелинейной системой на основе последовательных линеаризаций нелинейной системы. Эта идея развита в настоящей магистерской диссертации.

Глава 1. Линейная задача оптимального управления

В этой главе описан метод поиска оптимального управления для линейной задачи, на основе которого в дальнейшем строится алгоритм для нелинейной задачи (1)–(5). Метод состоит из двух взаимосвязанных этапов: сведение к интервальной задаче линейного программирования (ИЗЛП), изложенное в пункте 1.1, и решение ИЗЛП одним из методов линейного программирования, например, специально разработанным для этой задачи адаптивным методом (пункт 1.2).

В пункте 1.3 исследована структура множеств достижимости и управляемости линейной задачи, показан ряд их свойств, описан способ нахождения внутренних и внешних оценок этих множеств.

1.1. Постановка задачи. Сведение к ИЗЛП

Рассмотрим частный случай задачи (1)–(5): пусть правая часть системы (1) является линейной функцией относительно фазового вектора x и вектора управлений u , т. е. $f(x) = Ax + c$.

$$\begin{aligned} \int_0^T \|u(t)\|_1 dt &\longrightarrow \min_u, \\ \dot{x} &= Ax + Bu + c, \\ x(0) &= x_*, \quad x(T) = x^*, \\ l_i^{(1)} &\leq u_i(t) \leq l_i^{(2)}, \quad i = \overline{1, r}. \end{aligned} \tag{1.1}$$

При этом управление $u(t)$ выбирается из класса (4).

Покажем способ сведения задачи (1.1) к задаче линейного программирования [5, 8].

Для начала предположим, что $\forall i = \overline{1, r}, l_i^{(1)} \geq 0$. Это означает, что все компоненты вектора управлений будут неотрицательны на всем отрезке $[0, T]$. Выпишем формулу Коши при $t = T$ и воспользуемся представлени-

ем управления в виде (4):

$$x^* = e^{AT} x_* + \int_0^T e^{A(T-t)} c dt + \sum_{k=1}^N \sum_{i=1}^r \left(\int_{(k-1)h}^{kh} e^{A(T-t)} b_i dt u_{i,k} \right),$$

здесь b_i обозначает i -ый столбец матрицы B ; $u_{i,k}$, $i = \overline{1, r}$, $k = \overline{1, N}$ — совокупность из rN неизвестных переменных, смысл которых определяется формулой (4); $h = T/N$.

Выражение (5) с учетом (4) переписывается в виде

$$-\frac{T}{N} \sum_{k=1}^N \sum_{i=1}^r u_{i,k} \rightarrow \max_{u_{i,k}}.$$

Учитывая прямые ограничения на управление (3), получим задачу линейного программирования

$$\begin{aligned} \gamma^T U &\rightarrow \max_U, \\ DU &= g, \\ L_1 &\leq U \leq L_2. \end{aligned} \tag{1.2}$$

Здесь $U = (u_{1,1}, u_{2,1}, \dots, u_{r,1}, u_{1,2}, u_{2,2}, \dots, u_{r,N})^T$ — искомый rN -мерный вектор управления, определяющий размерность задачи.

D — матрица размера $n \times rN$, в которой столбцом под номером $(k-1)r + i$ является вектор

$$\int_{(k-1)h}^{kh} e^{A(T-t)} b_i dt.$$

g — n -мерный вектор, определяемый выражением

$$x^* - e^{AT} x_* - \int_0^T e^{A(T-t)} c dt.$$

γ — вектор-столбец, состоящий из rN чисел $-\frac{T}{N}$.

L_1, L_2 — rN -мерные векторы, состоящие из N повторений векторов $(l_1^{(1)}, \dots, l_r^{(1)})$ и $(l_1^{(2)}, \dots, l_r^{(2)})$ соответственно.

Предположим теперь, что найдется номер i^* такой, что $l_{i^*}^{(1)} < 0$. В случае $l_{i^*}^{(2)} \leq 0$ можем сделать замену переменной $v_{i^*} = -u_{i^*}$. Тогда $-l_{i^*}^{(2)} \leq v_{i^*} \leq -l_{i^*}^{(1)}$, где $-l_{i^*}^{(2)} \geq 0$. Матрица B , стоящая перед вектором управлений, изменится на $\tilde{B} = (b_1, b_2, \dots, b_{i^*-1}, -b_{i^*}, b_{i^*+1}, \dots, b_r)$. Функционал качества (5) не изменится т. к. $|v_{i^*}| = |u_{i^*}|$. Компоненты нового вектора управлений будут неотрицательны, следовательно сможем перейти к задаче (1.2) без увеличения ее размерности.

В случае $l_{i^*}^{(2)} > 0$ сделаем замену $u_{i^*} = v_{i^*}^{(2)} - v_{i^*}^{(1)}$, где новые управления имеют ограничения $0 \leq v_{i^*}^{(1)} \leq -l_{i^*}^{(1)}$, $0 \leq v_{i^*}^{(2)} \leq l_{i^*}^{(2)}$, т. е. являются неотрицательными величинами на отрезке $[0, T]$. Тогда новый вектор управлений будет иметь вид

$$(u_1(t), u_2(t), \dots, u_{i^*-1}(t), v_{i^*}^{(1)}, v_{i^*}^{(2)}, u_{i^*+1}(t), \dots, u_r(t)),$$

а матрица B изменится на

$$\tilde{B} = (b_1, b_2, \dots, b_{i^*-1}, -b_{i^*}, b_{i^*}, b_{i^*+1}, \dots, b_r).$$

Благодаря тому, что строится оптимальное управление с точки зрения функционала (5), на любом отрезке $[(k-1)h, kh)$, $k = \overline{1, N}$, одна из переменных $v_{i^*}^{(1)}$, $v_{i^*}^{(2)}$ будет равна нулю. Значит, слагаемое $|u_{i^*}(t)|$ подинтегрального выражения функционала (4) можно представить в виде $|v_{i^*}^{(1)}(t)| + |v_{i^*}^{(2)}(t)|$ или с учетом ограничений $v_{i^*}^{(1)}(t) + v_{i^*}^{(2)}(t)$. После такой замены все компоненты вектора управлений будут неотрицательны, следовательно сможем перейти к задаче линейного программирования (1.2), однако, ее размерность увеличится с rN до $rN + N$.

В худшем случае, когда все нижние ограничения имеют отрицательный знак, а все верхние — положительный, задача (1.2) будет иметь размерность $2rN$.

Таким образом, показан переход от линейной задачи оптимально-

го управления (1.1) к интервальной задаче линейного программирования (1.2). Ее размерность зависит от количества r управляющих параметров u_i и натурального числа N и составит от rN до $2rN$.

После того как будет найдено оптимальное решение U^0 задачи (1.2) (либо показано его отсутствие), по формуле (4) ему будет сопоставлено оптимальное управление $u^0(t)$. Оптимальное движение $x^0(t)$ будет получено интегрированием на отрезке $[0, T]$ системы дифференциальных уравнений из (1.1) замкнутой оптимальным управлением:

$$\dot{x} = Ax + Bu^0(t) + c, \quad x(0) = x_*.$$

В итоге, на выходе метода будет пара оптимальных движения и управления $(x^0(t), u^0(t))$.

1.2. Адаптивный метод

Определение 3. Будем называть *интервальной задачей линейного программирования (ИЗЛП)* задачу на поиск условного максимума следующего вида [6]:

$$\begin{aligned} c^T x &\rightarrow \max_x, \\ b_* &\leq Ax \leq b^*, \\ d_* &\leq x \leq d^*. \end{aligned} \tag{1.3}$$

В первой строке выписана *целевая функция* $c^T x$, на второй строке находятся *основные ограничения* задачи, на третьей — *прямые ограничения*. Обозначения в формуле (1.3) не связаны с обозначениями в (1.1).

Легко увидеть, что задача (1.2) является частным случаем ИЗЛП. Вектор U из (1.2) соответствует вектору x из (1.3), матрица D соответствует матрице A , $\gamma = c$, $L_1 = d_*$, $L_2 = d^*$, $g = b_* = b^*$.

Интервальная задача линейного программирования (1.3) может быть решена стандартными методами, например, симплекс-методом. Однако, в

работах Р. Ф. Габасова [6] был разработан специальный метод для решения ИЗЛП — адаптивный метод линейного программирования. Он имеет ряд преимуществ перед классическими методами, например, не требует сведения ИЗЛП к канонической задаче линейного программирования. Также метод не требует перемещения по вершинам допустимого множества, что является трудоемкой процедурой в задачах большой размерности, он использует произвольные точки данного множества. Опишем вкратце его алгоритм.

Определение 4. Под *планом* задачи (1.3) будем понимать вектор x , удовлетворяющий основным и прямым ограничениям задачи. *Оптимальным планом* будем называть план x^0 , доставляющий максимальное значение целевой функции среди всего множества планов.

Обозначим за m количество строк матрицы A , n — количество столбцов. Пусть $I = \{1, \dots, m\}$ — множество индексов строк матрицы A , $J = \{1, \dots, n\}$ — множество индексов столбцов. Выберем два подмножества $I_{op} \subseteq I$ и $J_{op} \subseteq J$ так, чтобы они имели одинаковое количество элементов $|I_{op}| = |J_{op}|$, а матрица составленная из строк матрицы A , с номерами входящими в множество I_{op} , и столбцов A , номера которых содержатся в J_{op} , являлась невырожденной: $\det(A(I_{op}, J_{op})) \neq 0$.

Определение 5. Будем называть совокупность $K_{op} = \{I_{op}, J_{op}\}$ *опорой* задачи (1.3), а матрицу $A_{op} = A(I_{op}, J_{op})$ — *опорной матрицей*. Под *опорным планом* будем понимать пару плана с опорной матрицей $\{x, K_{op}\}$.

Адаптивный метод является итеративным, его алгоритм состоит в замене опорного плана таким образом, чтобы значение целевой функции не уменьшалось на каждом шаге.

Пусть $\{x, K_{op}\}$ — опорный план на k -ой итерации метода. Введем

множества $I_n = I \setminus I_{op}$, $J_n = J \setminus J_{op}$.

В работе [6] был выведен критерий оптимальности опорного плана. Отметим, что важную роль играет опора, т. к. критерий может не распознать оптимальный план при неудачном выборе опоры.

С помощью критерия может быть найден *псевдоплан* \varkappa — специальный вектор, для которого выполняются условия оптимальности при текущей опоре K_{op} , но не гарантировано, что \varkappa принадлежит множеству планов.

Если \varkappa является планом, то он оптимальный план, и задача (1.3) решена. В случае, если для него не выполняется хотя бы одно из основных или прямых ограничений, осуществим замену плана. Рассмотрим отрезок $x + l(\varkappa - x)$, $l \in [0, 1]$. Видно, что начало отрезка (при $l = 0$) является планом, конец ($l = 1$) — не план. Следовательно, существует $l^* \in (0, 1)$ такое, что $\bar{x} = x + l^*(\varkappa - x)$ лежит на границе множества планов (т. к. это выпуклый многогранник). Примем \bar{x} за новый план.

Заменяем теперь опору. Для \bar{x} найдется такой номер $i_* \in I_n$, что основное двустороннее неравенство под этим номером обратится в равенство с одной стороны, либо найдется $j_* \in J_{op}$, на котором основное неравенство с этим индексом перейдет в равенство. В первом случае положим $k_* = i_*$, во втором — $k_* = j_*$.

Проверим условия оптимальности для опорного плана $\{\bar{x}, K_{op}\}$. Если он не оптимален, найдется либо номер $k_0 = i_0 \in I_{op}$, либо номер $k_0 = j_0 \in J_n$ условия, на котором нарушается условие оптимальности.

Опора будет заменена в соответствии с одним из четырех случаев.

1. $k_0 = j_0 \in J_n$, $k_* = j_* \in J_{op}$. В этом случае $\bar{I}_{op} = I_{op}$, $\bar{J}_{op} = (J_{op} \setminus j_*) \cup j_0$.

В опорной матрице A_{op} заменяется столбец.

2. $k_0 = j_0 \in J_n$, $k_* = i_* \in I_n$. В этом случае $\bar{I}_{op} = I_{op} \cup i_*$, $\bar{J}_{op} = J_{op} \cup j_0$.

В опорную матрицу добавляются строка и столбец.

3. $k_0 = i_0 \in I_{op}$, $k_* = j_* \in J_{op}$. В этом случае $\bar{I}_{op} = I_{op} \setminus i_0$, $\bar{J}_{op} = J_{op} \setminus j_*$.

Из опорной матрицы удаляются строка и столбец.

4. $k_0 = i_0 \in I_{op}$, $k_* = i_* \in I_n$. В этом случае $\bar{I}_{op} = (I_{op} \setminus i_0) \cup i_*$, $\bar{J}_{op} = J_{op}$.

В опорной матрице заменяется строка.

Далее процедура повторяется для нового опорного плана $\{\bar{x}, \bar{K}_{op}\}$, где $\bar{K}_{op} = \{\bar{I}_{op}, \bar{J}_{op}\}$.

После каждой замены плана и опоры вычисляется $\beta = c^T(\bar{x} - \varkappa)$. При $\beta = 0$ процесс останавливается, оптимальное решение $x^0 = \bar{x}$.

Доказано, что процесс сходится за конечное количество шагов.

Чтобы найти опорный план для первого шага метода, необходимо решить вспомогательную задачу, для которой начальный опорный план находится тривиально, а оптимальным решением является план задачи (1.3). Таким образом, необходимо дважды применить алгоритм, что делает адаптивный метод двухфазным.

1.3. Множество достижимости и множество управляемости

Рассмотрим задачу построения множества достижимости для задачи (1.1).

Определение 6. Будем называть *множеством достижимости* $Z_1(t)$ задачи (1.1) множество всевозможных состояний $z^{(1)}(t)$, в которые система $\dot{x} = Ax + Bu + c$ может попасть за время t при начальном условии $x(0) = x_*$ и выборе управления вида (4), удовлетворяющего ограничению (3):

$$Z_1(t) = \{z^{(1)}(t) = x(t) | \dot{x}(\tau) \equiv Ax(\tau) + Bu(\tau) + c, x(0) = x_*, \tau \in [0, t], \\ u(\tau) \text{ удовлетворяет (3)–(4)}\}.$$

Любое допустимое состояние $z^{(1)}(t)$ может быть вычислено по формуле по формуле Коши:

$$z^{(1)}(t) = e^{At}x_* + \int_0^t e^{A(t-\tau)}c \, d\tau + \sum_{s=1}^{k-1} \sum_{i=1}^r \left(\int_{(s-1)h}^{sh} e^{A(t-\tau)}b_i \, d\tau \, u_{i,s} \right) + \sum_{i=1}^r \left(\int_{(k-1)h}^t e^{A(t-\tau)}b_i \, d\tau \, u_{i,k} \right),$$

где $l_i^{(1)} \leq u_{i,s} \leq l_i^{(2)}$, $i = \overline{1, r}$, $s = \overline{1, k}$, $t \in ((k-1)h, kh]$, $k = \overline{1, N}$, $h = T/N$.

Введя соответствующие обозначения, получим то же самое в матричном виде:

$$z^{(1)}(t) = D_1(t)U_1 + q_1(t), \quad L_1^{(1)} \leq U_1 \leq L_1^{(2)}.$$

Размер матрицы $D_1(t) - n \times rk$, $z^{(1)}(t)$ и $q_1(t) - n$ -мерные векторы, $L_1^{(1)}$, $L_1^{(2)}$, $U_1 -$ векторы размера rk , $t \in ((k-1)h, kh]$, $k = \overline{1, N}$.

Таким образом, зафиксировав t , сможем найти множество $Z_1(t)$ путем варьирования вектор U_1 .

Найдем теперь формулу для вычисления множества управляемости в момент t .

Определение 7. Будем называть *множеством управляемости* $Z_2(t)$ задачи (1.1) множество всевозможных состояний $z^{(2)}(t)$, из которых система $\dot{x} = Ax + Bu + c$ может попасть за время $T - t$ в состояние x^* при выборе управления вида (4), удовлетворяющего ограничению (3):

$$Z_2(t) = \{z^{(2)} = x(t) | \dot{x}(\tau) \equiv Ax(\tau) + Bu(\tau) + c, x(T) = x^*, \tau \in [t, T], u(t) \text{ удовлетворяет (3)-(4)}\}.$$

Преобразовав формулу Коши, получим:

$$z^{(2)}(t) = e^{-A(T-t)}x^* - \int_t^T e^{-A(\tau-t)}c \, d\tau - \sum_{s=k+1}^N \sum_{i=1}^r \left(\int_{(s-1)h}^{sh} e^{-A(\tau-t)}b_i \, d\tau \, u_{i,s} \right) - \sum_{i=1}^r \left(\int_t^{kh} e^{-A(\tau-t)}b_i \, d\tau \, u_{i,k} \right),$$

где $l_i^{(1)} \leq u_{i,s} \leq l_i^{(2)}$, $i = \overline{1, r}$, $s = \overline{k, N}$, $t \in ((k-1)h, kh]$, $k = \overline{1, N}$,
 $h = T/N$.

Перепишем в матричном виде:

$$z^{(2)}(t) = D_2(t)U_2 + q_2(t), \quad L_2^{(1)} \leq U_2 \leq L_2^{(2)}.$$

Размер матрицы $D_2(t) - n \times r(N - k + 1)$, $z^{(2)}(t)$ и $q_2(t) - n$ -мерные векторы, $L_2^{(1)}$, $L_2^{(2)}$, $U_2 -$ векторы размера $r(N - k + 1)$, $t \in ((k-1)h, kh]$, $k = \overline{1, N}$.

Таким образом, зафиксировав t , сможем найти множество $Z_2(t)$ путем варьирования вектор U_2 .

Видно, что формулы для нахождения множеств $Z_1(t)$ и $Z_2(t)$ имеют один и тот же вид, а компоненты векторов U_1 и U_2 должны удовлетворять двусторонним ограничениям, поэтому задачу поиска этих двух множеств можно объединить в одну задачу: найти множество \mathbf{Z} значений вектора z , удовлетворяющего соотношению $z = Du + q$, где вектор u принадлежит области $\mathbf{U} = [l_1^{(1)}; l_2^{(2)}] \times [l_2^{(1)}; l_2^{(2)}] \times \dots \times [l_m^{(1)}; l_m^{(2)}]$. Условимся, что размерность вектора z равна n , а вектора $u - m$. Текущие обозначения переменных и их размерностей не будем связывать с обозначениями использовавшимися до этого момента.

В итоге, получили задачу нахождения множества \mathbf{Z} :

$$z \in \mathbf{Z} \iff \exists u \in \mathbf{U} : z = Du + q. \quad (1.4)$$

Произведем замену переменных

$$v = Ku + g, \quad g_i = -\frac{l_i^{(1)} + l_i^{(2)}}{l_i^{(2)} - l_i^{(1)}}, \quad k_{ij} = \begin{cases} \frac{2}{l_i^{(2)} - l_i^{(1)}}, & i = j \\ 0, & i \neq j \end{cases}.$$

$$y = z + Pg - q, \quad P = DK^{-1}.$$

Тогда формула (1.4) перейдет в

$$y = Pv, \quad -1 \leq v_j \leq 1, \quad j = \overline{1, s}. \quad (1.5)$$

Новое множество допустимых управлений будем обозначать $\mathbf{V} = [-1; 1] \times [-1; 1] \times \dots \times [-1; 1]$, а множество достижимых состояний $\mathbf{Y} = \{y \in \mathbf{R}^n \mid \exists v \in \mathbf{V} : y = Pv\}$.

Рассмотрим структуру множества \mathbf{Y} .

Свойство 1. \mathbf{Y} — непустое ограниченное замкнутое выпуклое множество.

Следует из того, что множество \mathbf{V} является непустым, ограниченным, замкнутым и выпуклым, а линейное отображение сохраняет вышеперечисленные свойства.

Свойство 2. \mathbf{Y} — выпуклая оболочка точек $\{p^1, p^2, \dots, p^{2^m}\}$, где $p^k = Pv^k$, v^k — вершины \mathbf{V} .

Множество \mathbf{V} очевидно является выпуклой оболочкой своих вершин, т. е. $\forall v \in \mathbf{V} \exists \lambda_k, k = \overline{1, 2^m} : v = \sum_{k=1}^{2^m} \lambda_k v^k$, где $\sum_{k=1}^{2^m} \lambda_k = 1$, $\lambda_k \geq 0$, здесь v^k — различные векторы, состоящие из комбинаций -1 и 1 . Тогда $\forall y \in \mathbf{Y} \exists \lambda_k : y = \sum_{k=1}^{2^m} \lambda_k Pv^k = \sum_{k=1}^{2^m} \lambda_k p^k$, $\sum_{k=1}^{2^m} \lambda_k = 1$, $\lambda_k \geq 0$, $k = \overline{1, 2^m}$. Следовательно, множество \mathbf{Y} — выпуклая оболочка векторов $p^k = Pv^k$.

Свойство 3. \mathbf{Y} — выпуклый многогранник, вершинами которого является подмножество набора векторов $\{p^1, p^2, \dots, p^{2^m}\}$, где $p^k = Pv^k$, v^k — вершины \mathbf{V} .

Следует из свойства 2 и свойств выпуклых оболочек.

Свойство 4. Если y принадлежит \mathbf{Y} , то и $-y \in \mathbf{Y}$.

Если $y \in \mathbf{Y}$, то $\exists v \in \mathbf{V} : y = Pv$. Но и $-v \in \mathbf{V}$, а следовательно, $P(-v) = -y \in \mathbf{Y}$.

Свойство 5. Множество \mathbf{Y} описывается системой неравенств типа $-b \leq Ay \leq b$.

\mathbf{Y} — выпуклый многогранник, следовательно может быть описан системой линейных неравенств. По свойству 4, если $\forall y \in \mathbf{Y}$ имеет место неравенство $a_i y \leq b_i$, то оно имеет место и для $-y : a_i(-y) \leq b_i$. Тогда $-b_i \leq a_i y$.

Свойство 2 говорит о том, что множество \mathbf{Y} может быть построено как выпуклая оболочка точек $\{p^1, p^2, \dots, p^{2^m}\}$, $p^k = Pv^k$. Однако, такой подход не реализуем на практике даже при сравнительно небольших значениях m , например, при $m = 30$ имеем около 10^9 точек. Впрочем, выбрав некоторое подмножество этих точек и построив их выпуклую оболочку, получим внутреннюю оценку множества достижимости.

Домножим вектор Pv на n -мерную вектор-строку λ слева и найдем максимум этой функции по $v \in \mathbf{V}$. Это $|\lambda p^1| + |\lambda p^2| + \dots + |\lambda p^s|$ (p^j — столбцы матрицы P). Здесь $v_j = \text{sgn}(\lambda p^j)$, $j = \overline{1, m}$ — это некоторая вершина \mathbf{V} . Перебирая некоторым образом различные варианты значений λ будем получать вершины v^i множества \mathbf{V} и точки $y^i = Pv^i$.

Рассмотрим частный случай такого подхода. Выберем $\lambda = (1, 0, \dots, 0)$. Тогда $\lambda Pv = p_1 v$, где p_1 — 1-ая строка P . Возьмем

$$v = (\text{sgn}(p_{11}), \text{sgn}(p_{12}), \dots, \text{sgn}(p_{1r}))$$

и найдем $y = Pv$ — это вершина \mathbf{Y} , в которой достигается максимальное значение по 1-ой переменной y_1 . Теперь выберем $\lambda = (-1, 0, \dots, 0)$, $\lambda Pv = -p_1 v$. Тогда

$$v = (-\text{sgn}(p_{11}), -\text{sgn}(p_{12}), \dots, -\text{sgn}(p_{1r})),$$

а $y = Pv$ — это вершина \mathbf{Y} , в которой достигается минимальное значение по y_1 . Прделаав то же самое для y_2, y_3, \dots, y_n , получим не более $2n$ вершин.

Внешней же оценкой этого множества будет служить n -мерный параллелепипед, сторонами которого являются отрезки от минимального до

максимального значения каждой переменной $y_i, i = \overline{1, n}$.

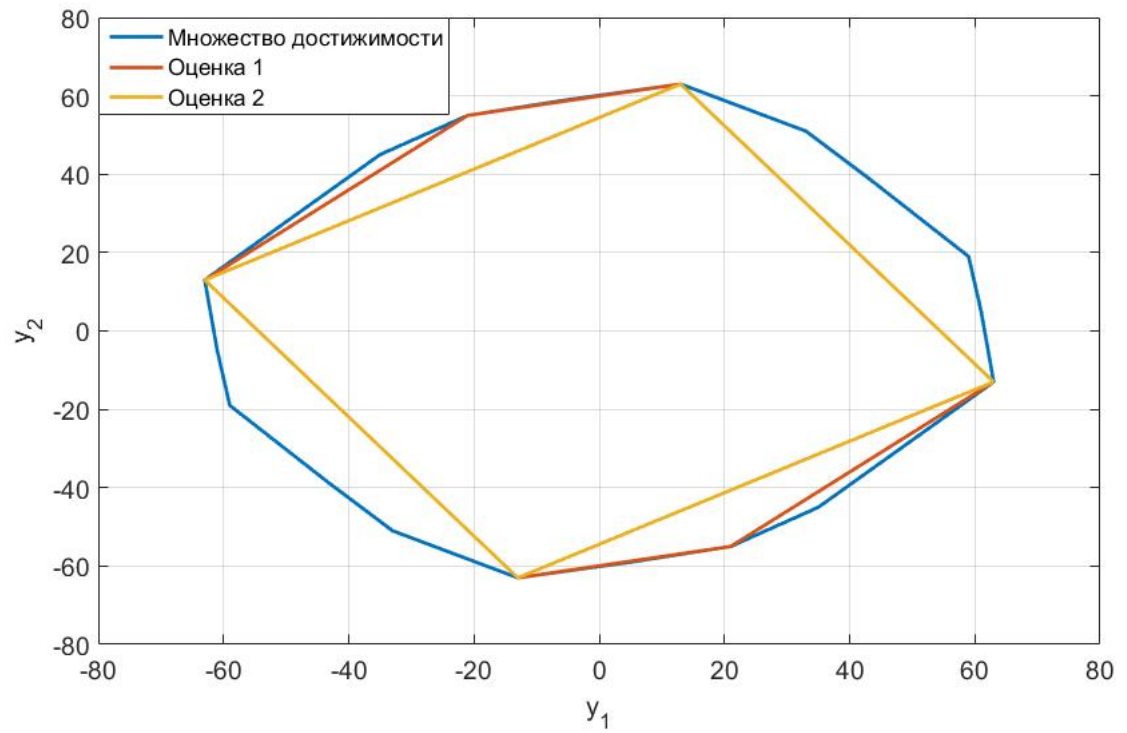


Рис. 1: Пример множества достижимости и 2 внутренние оценки.

Глава 2. Нелинейная задача оптимального управления

В данной главе описан метод позиционного управления нелинейной системой, предложенный Габасовым. На его основе предложен приближенный метод решения задачи оптимального управления для нелинейной системы, сформулирован алгоритм, доказан ряд свойств метода.

2.1. Метод позиционного управления Габасова

В работе [7] изложен метод нахождения оптимального управления задачи (1)–(5). Основная идея состоит в замене нелинейной функции $f(x)$ кусочно-линейной аппроксимацией во всем пространстве \mathbb{R}^n .

Рассмотрим скалярный случай. Пусть начальное положение системы x_* находится на участке I , где $f(x)$ аппроксимируется функцией $a_1x + c_1$, терминальное положение x^* расположено в участке II , где аппроксимацией $f(x)$ является линейная функция $a_2x + c_2$. Участки I и II — соседние, их общая граница располагается в точке \hat{x} , $a_1\hat{x} + c_1 = a_2\hat{x} + c_2$. Тогда вместо (1)–(2) получим:

$$\begin{aligned}\dot{x}^1 &= a_1x^1 + bu + c_1, & t \in [0, N_1h], \\ x^1(0) &= x_*, & x^1(N_1h) = \hat{x}, \\ \dot{x}^2 &= a_2x^2 + bu + c_2, & t \in [N_1h, T], \\ x^2(N_1h) &= \hat{x}, & x^2(T) = x^*.\end{aligned}$$

N_1 — натуральное число между 1 и N . Учитывая условия (3)–(5), получим задачу, которая может быть сведена к интервальной задаче линейного программирования с параметром N_1 , подобно тому, как это делается в пункте 1.1. Та в свою очередь может быть решена адаптивным методом, описанным в пункте 1.2. Оптимальная величина N_1 находится перебором. Далее, найденным оптимальным управлением замыкается система (1) и интегрируется на отрезке $[0, h]$, в результате чего получаем реальное движение

$x^0(t)$ на этом отрезке.

Теперь необходимо решить задачу (1)–(5) при начальном положении $x^0(h)$ в момент времени h . Если это положение находится в области I , получим систему

$$\begin{aligned} \dot{x}^1 &= a_1 x^1 + bu + c_1, & t \in [h, N_1 h], \\ x^1(h) &= x^0(h), & x^1(N_1 h) = \hat{x}, \\ \dot{x}^2 &= a_2 x^2 + bu + c_2, & t \in [N_1 h, T], \\ x^2(N_1 h) &= \hat{x}, & x^2(T) = x^*. \end{aligned}$$

В противном случае имеем

$$\begin{aligned} \dot{x}^2 &= a_2 x^2 + bu + c_2, & t \in [h, T], \\ x^2(h) &= x^0(h), & x^2(T) = x^*. \end{aligned}$$

Как и на предыдущем шаге сведем эту задачу к ИЗЛП. В результате будет найдено оптимальное управление. При замыкании им нелинейной системы, будет найдена траектория $x^0(t)$ при $t \in [h, 2h]$. Прделав то же самое при $t \in [2h, 3h], \dots [(N-1)h, T]$, получим оптимальное движение $x^0(t)$ на отрезке $[0, T]$.

В случае необходимости движения через l участков, задача линейного программирования будет содержать $l-1$ параметр N_1, N_2, N_{l-1} .

2.2. Модификация метода позиционного управления

Запишем постановку задачи (1)–(5) еще раз:

$$\begin{aligned} \int_0^T \|u(t)\|_1 dt &\longrightarrow \min_u, \\ \dot{x} &= f(x) + Bu, \\ x(0) &= x_*, \quad x(T) = x^*, \\ l_i^{(1)} &\leq u_i(t) \leq l_i^{(2)}, \quad i = \overline{1, r}. \end{aligned} \tag{2.1}$$

Управление выбирается в классе кусочно-постоянных функций вида (4).

По аналогии с решением задачи (1.1) хотелось бы свести текущую задачу к задаче математического программирования, однако, такой подход не осуществим ввиду отсутствия аналога формулы Коши для нелинейных систем дифференциальных уравнений. Поэтому попытаемся построить приближенное решение путем сведения задачи (2.1) к задаче (1.1) [14].

Единственное различие этих задач — наличие нелинейной функции $f(x)$ в правой части системы дифференциальных уравнений в (2.1). Получим линейную задачу, если заменим $f(x)$ некоторым линейным приближением. По заданию это непрерывно-дифференцируемая во всем пространстве \mathbb{R}^n функция, значит, можем линейризовать ее суммой нулевого и первого членов ряда Тейлора:

$$f(x) \approx f(\xi) + \left. \frac{df(x)}{dx} \right|_{x=\xi} (x - \xi).$$

Для j -ой компоненты вектор-функции $f(x)$ имеем:

$$f_j(x) \approx f_j(\xi) + \sum_{i=1}^n \left. \frac{\partial f_j(x_1, \dots, x_i, \dots, x_n)}{\partial x_i} \right|_{i=\overline{1, n}}^{x_i=\xi_i} (x_i - \xi_i).$$

ξ — некоторая точка пространства \mathbf{R}^n .

Введем теперь обозначения:

- $A_\xi = \left. \frac{df(x)}{dx} \right|_{x=\xi}$ — матрица Якоби функции $f(x)$, ее размерность — $n \times n$,
- $c_\xi = f(\xi) - \left. \frac{df(x)}{dx} \right|_{x=\xi} \xi$ — n -мерный вектор.

Учитывая новые обозначения, зададим линейную задачу оптимального управления с линейным приближением $f(x)$ взятым в точке ξ :

$$\begin{aligned} \int_0^T \|u(t)\|_1 dt &\longrightarrow \min_u, \\ \dot{x} &= A_\xi + Bu + c_\xi, \\ x(0) &= x_*, \quad x(T) = x^*, \\ l_i^{(1)} &\leq u_i(t) \leq l_i^{(2)}, \quad i = \overline{1, r}. \end{aligned} \tag{2.2}$$

Важным вопросом является выбор конкретного значения ξ , в следующем пункте показано, что, выбрав его равным начальной точке движения системы, отклонение решения линейной системы от решения нелинейной системы, взятой с тем же управлением, в своем разложении в ряд Тейлора будет иметь нулевой, первый и второй члены равными нулю. Поэтому будем искать решение задачи (2.2) при $\xi = x_*$.

По алгоритму, описанному в главе 1, для этой задачи может быть найдено точное оптимальное управление $u(t)$, переводящее систему из точки x_* в точку x^* за время T и соответствующее этому управлению движение $x(t)$. Однако реальная траектория движения объекта $x^0(t)$ описывается системой (1), замкнутой оптимальным управлением $u(t)$, ее значение в общем случае не совпадает с $x(t)$.

Найдем решение задачи (2.2) при $\xi = x_*$. Обозначим за $x^1(t)$ и $u^1(t)$ оптимальные траекторию и управление соответственно. Согласно формуле (4), $u^1(t)$ является кусочно-постоянной вектор-функцией, т. е. на каждом из участков времени $[(k-1)h, kh)$ она является постоянным вектором: $u^1(t) = u_k^1, k = \overline{1, N}$. Замкнем нелинейную систему (1) управлением $u^1(t)$ на отрезке $[0, h]$, т. е. вектором u_1^1 :

$$\dot{x}(t) = f(x(t)) + Bu_1^1, \quad x(0) = x_*,$$

и проинтегрируем ее на этом участке. Решение замкнутой системы обозначим $x^0(t)$. Траектории $x^0(t)$ и $x^1(t)$ связаны соотношением $x^0(t) = x^1(t) + z(t), t \in [0, h]$, где невязка $z(t)$ возникает за счет неточности аппроксимации $f(x)$ ее линейным приближением.

С целью избежания накопления ошибок, в момент $t = h = T/N$ пересчитаем управление на оставшемся промежутке $[h, T]$ с учетом текущей позиции $x^0(h)$. Найдем оптимальное управление $u^2(t)$ в виде совокупности векторов $u_k^2, k = \overline{2, N}$ и оптимальную траекторию $x^2(t)$, как решение

линейной задачи

$$\begin{aligned} \int_h^T \|u(t)\|_1 dt &\rightarrow \min_u, \\ \dot{x} &= A_{x^0(h)}x + Bu + c_{x^0(h)}, \\ x(h) &= x^0(h), \quad x(T) = x^*, \\ l_i^{(1)} &\leq u_i(t) \leq l_i^{(2)}, \quad i = \overline{1, r}. \end{aligned}$$

Здесь $A_{x^0(h)} = \left. \frac{df(x)}{dx} \right|_{x=x^0(h)}$, $c_{x^0(h)} = f(x^0(h)) - \left. \frac{df(x)}{dx} \right|_{x=x^0(h)} x^0(h)$.

Реальную траекторию $x^0(t)$ при $t = [h, 2h]$ найдем из уравнения

$$\dot{x} = f(x) + Bu_2^2, \quad x(h) = x^0(h)$$

Отклонение реальной траектории $x^0(t)$ от $x^2(t)$ обозначим за $z(t) = x^0(t) - x^2(t)$, $t \in [h, 2h]$.

Решая аналогичную задачу при $t \in [2h, 3h], \dots, [(N-1)h, T]$, получим траекторию $x^0(t)$ на промежутке $[0, T]$ и управление

$$u^0(t) = u_k^k, \quad t \in [(k-1)h, kh], \quad k = \overline{1, N},$$

зависящее от позиций x_* , $x^0(h)$, $x^0(2h)$, \dots , $x^0((N-1)h)$.

В итоге, решив N раз линейную задачу оптимального управления, получим приближенное решение нелинейной задачи (2.1).

Замечание 2. Предполагается, что в каждый момент времени $t = h, 2h, \dots, (N-1)h$ можно точно оценить текущее положение системы $x^0(h)$, $x^0(2h)$, \dots , $x^0((N-1)h)$ соответственно.

Замечание 3. Предполагается, что на каждом шаге точка x^* принадлежит области достижимости вспомогательной линейной задачи.

Замечание 4. При наличии вычислительных систем, способных быстро решать вспомогательные линейные задачи, можно считать, что управление строится в режиме реального времени.

Замечание 5. Преимуществом модифицированного метода по сравнению с методом Габасова, изложенным в пункте 2.1, является отсутствие

необходимости строить кусочно-линейную аппроксимацию функции $f(x)$. Линейные задачи не содержат параметров, возникающих при переходе с одного участка на другой, что значительно ускоряет работу метода.

2.3. Обоснование сходимости

Для начала покажем, что наилучшим выбором точки ξ является начальное положение x_* (с точки зрения разложения в ряд Тейлора отклонения решения линейной задачи от нелинейной при одинаковом управлении).

Утверждение 1. Если функция $f(x)$ — непрерывно-дифференцируемая функция аргумента x , то при $\xi = x_*$ отклонение

$$z(t) = \tilde{x}(t) - x(t) = o(t^2),$$

где $x(t)$ — оптимальная траектория задачи (2.2) при $\xi = x_*$, а $\tilde{x}(t)$ — решение замкнутой тем же управлением системы (1) с начальным условием $x(0) = x_*$.

Доказательство. Пусть найдено оптимальное управление $u(t)$ задачи (2.2) при $\xi = x_*$, а $x(t)$ является оптимальной траекторией этой задачи. Тогда реальное движение $\tilde{x}(t)$ при $t \geq 0$ может быть найдено интегрированием системы

$$\dot{x} = f(x) + Bu(t), \quad x(0) = x_*.$$

Отклонение $\tilde{x}(t)$ системы от решения задачи (2.2) может быть описано уравнением

$$\begin{aligned} \dot{z}(t) &= \dot{\tilde{x}}(t) - \dot{x}(t) = \\ &= f(\tilde{x}(t)) + Bu(t) - f(x_*) - \frac{df(x)}{dx} \Big|_{x=x_*} (x(t) - x_*) - Bu(t) = \\ &= f(z(t) + x(t)) - f(x_*) - \frac{df(x)}{dx} \Big|_{x=x_*} (x(t) - x_*). \end{aligned} \quad (2.3)$$

При этом $z(0) = \tilde{x}(0) - x(0) = x_* - x_* = 0$. Если $f(x)$ — дифференцируема, то отклонение $z(t)$ будет дважды дифференцируемой функцией и может

быть разложено в ряд Тейлора:

$$z(t) = z(0) + \dot{z}(0)t + \ddot{z}(0)\frac{t^2}{2} + o(t^2).$$

Покажем, что $z(0)$, $\dot{z}(0)$, $\ddot{z}(0)$ равны 0.

$z(0) = 0$ – очевидно, т. к. отклонение в начальный момент отсутствует.

Подставим значение $t = 0$ в формулу (2.3):

$$\dot{z}(0) = f(x_*) - f(x_*) - \left. \frac{df(x)}{dx} \right|_{x=x_*} (x_* - x_*) = 0.$$

Чтобы показать $\ddot{z}(0) = 0$ продифференцируем выражение (2.3):

$$\ddot{z}(t) = \left. \frac{df(x)}{dx} \right|_{x=x(t)+z(t)} (\dot{x}(t) + \dot{z}(t)) - \left. \frac{df(x)}{dx} \right|_{x=x_*} \dot{x}(t).$$

В точке $t = 0$ имеем

$$\ddot{z}(0) = \left. \frac{df(x)}{dx} \right|_{x=x_*} (\dot{x}(0) + \dot{z}(0)) - \left. \frac{df(x)}{dx} \right|_{x=x_*} \dot{x}(0) = 0.$$

Таким образом $z(t) = o(t^2)$ или в других обозначениях

$$\tilde{x}(t) = x(t) + o(t^2)$$

□

Для обоснования метода покажем, что в пределе при $N \rightarrow \infty$ финальное положение системы $x^0(T)$ будет стремиться к x^* .

Утверждение 2. $\lim_{N \rightarrow \infty} x^0(T) = x^*$ или что то же $\lim_{h \rightarrow 0} x^0(T) = x^*$

Доказательство. Рассмотрим отклонение $z(t)$ реальной траектории $x^0(t)$ от решения вспомогательной линейной задачи $x^N(t)$ на последнем шаге. Представим $z(t)$ в виде ряда Тейлора взятого в точке $T - h$:

$$z(t) = z(T-h) + \dot{z}(T-h)(t-T+h) + \ddot{z}(T-h)\frac{(t-T+h)^2}{2} + o((t-T+h)^2).$$

Подобно тому как это делается в утверждении 1, можно показать, что $z(T-h)$, $\dot{z}(T-h)$, $\ddot{z}(T-h)$ равны 0. Тогда в финальной точке $t = T$ имеем $z(T) = o(h^2)$.

Устремив h к 0, получим $\lim_{h \rightarrow 0} z(T) = 0$. Учитывая, что $z(T) = x^0(T) - x^N(T)$, а $x^N(T) = x^*$, окончательно имеем $\lim_{h \rightarrow 0} x^0(T) = x^*$. □

Перед решением задачи оптимального управления желательно удостовериться в существовании допустимого решения этой задачи. Для линейного случая в пункте 1.3 были построены оценки множества достижимости $Z_1(t)$ и множества управляемости $Z_2(t)$. Для доказательства существования допустимого решения задачи (1.1), достаточно удостовериться в одном из двух:

- Существует внутренняя оценка $\tilde{Z}_1(T)$ множества достижимости $Z_1(T)$, такая что $x^* \in \tilde{Z}_1(T)$.
- Существует внутренняя оценка $\tilde{Z}_2(T)$ множества управляемости $Z_2(T)$, такая что $x_* \in \tilde{Z}_2(T)$.

Чтобы доказать отсутствие допустимого решения, необходимо показать одно из двух:

- Существует внешняя оценка $\bar{Z}_1(T)$ множества достижимости $Z_1(T)$, такая что $x^* \notin \bar{Z}_1(T)$.
- Существует внешняя оценка $\bar{Z}_2(T)$ множества управляемости $Z_2(T)$, такая что $x_* \notin \bar{Z}_2(T)$.

Предположим теперь, что для задачи (2.2) при $\xi = x_*$ существует допустимое решение. Найдем оптимальное управление $u^1(t)$, замкнем систему (1) этим управлением и проинтегрируем при $t = [0, T]$. Обозначим полученную функцию за $\tilde{x}(t)$ ($\tilde{x}(t) = x^0(t)$ при $t \in [0, h]$). Возникает вопрос: как далеко будет $\tilde{x}(T)$ от x^* ? Зададимся задачей найти условия на функцию $f(x)$, чтобы выполнялось неравенство $\|\tilde{x}(T) - x^*\| < \varepsilon$. Для этого необходимо ограничить отклонение производной $f(x)$ от ее значения в

точке x_* : $\|A_x - A_{x_*}\| < \delta$ для любого $x \in \mathbb{R}^n$. Связь между ε и δ описана в утверждении 3.

Утверждение 3. Пусть норма производной функции $f(x)$ ограничена на $\forall x \in \mathbb{R}^n$: $\frac{df(x)}{dx} = A_{x_*} + \mathcal{A}(x - x_*)$, $\|\mathcal{A}(x - x_*)\| < \delta$. Введем обозначение $M = f(x_*) + \sum_{i=1}^r \max(|b_i l_i^{(1)}|, |b_i l_i^{(2)}|)$. Тогда верхнюю оценку ε отклонения решения $\tilde{x}(t)$ нелинейной системы с управлением, полученным на первом шаге, от решения $x^1(t)$ линейной системы с управлением, полученным на первом шаге, можно найти по формуле

$$\frac{4M\gamma(\omega)}{(\alpha + \omega)^2} \delta e^{(\alpha + \omega + \|A_{x_*}\| + \delta)t} = \varepsilon,$$

где $\omega > 0$, $\gamma(\omega) \geq 1$, α — максимальное значение вещественной части собственных чисел матрицы A_{x_*} .

Доказательство. Воспользуемся теоремой Лагранжа о среднем значении: $f(x(t)) = f(x_*) + \frac{df(x)}{dx} \Big|_{x=x_*+\varkappa(t)} (x(t) - x_*)$, где $\varkappa(t)$ — некоторая точка промежутка $(0, x(t) - x_*)$. Введем $z_0(t) = \tilde{x}(t) - x_*$, тогда нелинейное уравнение (1), замкнутое управлением $u^1(t)$ переписется в виде

$$\begin{aligned} \dot{z}_0(t) &= \frac{df(x)}{dx} \Big|_{x=x_*+\varkappa(t)} z_0(t) + f(x_*) + Bu^1(t) = \\ &= (A_{x_*} + \mathcal{A}(\varkappa))z_0(t) + f(x_*) + Bu^1(t). \end{aligned}$$

Обозначим $z_1(t) = x^1(t) - x_*$, тогда уравнение (2.2) при $\xi = x_*$, замкнутое управлением $u^1(t)$, будет выглядеть как

$$\dot{z}_1(t) = \frac{df(x)}{dx} \Big|_{x=x_*} z_1(t) + f(x_*) + Bu^1(t) = A_{x_*} z_1(t) + f(x_*) + Bu^1(t).$$

Оценим норму разности z_0 и z_1 :

$$\begin{aligned} \|z_0(t) - z_1(t)\| &\leq \int_0^t \|A_{x_*}(z_0(\tau) - z_1(\tau))\| d\tau + \\ &+ \int_0^t \|\mathcal{A}(\varkappa(\tau))z_0(\tau) + \mathcal{A}(\varkappa(\tau))(z_1(\tau) - z_1(\tau))\| d\tau \leq \\ &\leq (\|A_{x_*}\| + \delta) \int_0^t \|z_0(\tau) - z_1(\tau)\| d\tau + \delta \int_0^t \|z_1(\tau)\| d\tau \end{aligned}$$

$z_1(\tau)$ — решение линейного уравнения, его норма может быть оценена сверху:

$$\|z_1(\tau)\| \leq \frac{2M\gamma(\omega)}{\alpha + \omega} e^{(\alpha+\omega)\tau}.$$

Учитывая, что $\|z_0(t) - z_1(t)\| = \|\tilde{x}(t) - x^1(t)\|$, получим

$$\|\tilde{x}(t) - x^1(t)\| \leq (\|A_{x_*}\| + \delta) \int_0^t \|\tilde{x}(\tau) - x^1(\tau)\| d\tau + \frac{4M\gamma(\omega)}{(\alpha + \omega)^2} \delta e^{(\alpha+\omega)t}.$$

После применения леммы Гронуолла:

$$\|\tilde{x}(t) - x^1(t)\| \leq \frac{4M\gamma(\omega)}{(\alpha + \omega)^2} \delta e^{(\alpha+\omega+\|A_{x_*}\|+\delta)t} = \varepsilon.$$

□

2.4. Общий алгоритм и программная реализация

В этом пункте опишем общий алгоритм решения задачи (2.1) методом, изложенным в пункте 2.2.

Входные данные

Система (1) задается непрерывно-дифференцируемой n -мерной функцией $f(x)$ и матрицей B размера $n \times r$. Определяются конечный момент времени — положительное число T , граничные условия в виде двух n -мерных векторов: начальное положение системы x_* и финальное состояние x^* . Задаются нижние $l_i^{(1)}$ и верхние $l_i^{(2)}$, $i = \overline{1, r}$, ограничения на компоненты управляющего вектора. Последним параметром является положительное целое число N .

Предварительные вычисления

На начальном этапе вычисляется матрица

$$A_\xi = \left. \frac{df(x)}{dx} \right|_{x=\xi}$$

и вектор

$$c_\xi = f(\xi) - \left. \frac{df(x)}{dx} \right|_{x=\xi} \xi$$

в общем виде для произвольной точки ξ .

Шаг 1

1. Определяются матрица A_{x_*} и вектор c_{x_*} путем подстановки x_* вместо ξ в A_ξ и c_ξ . Получаем вспомогательную линейную задачу:

$$\int_0^T \|u(t)\|_1 dt \rightarrow \min_u,$$

$$\dot{x}(t) = A_{x_*}x(t) + Bu(t) + c_{x_*},$$

$$x(0) = x_*, \quad x(T) = x^*,$$

$$l_i^{(1)} \leq u_i(t) \leq l_i^{(2)}, \quad i = \overline{1, r}.$$

2. Сведение вспомогательной системы к задаче линейного программирования вида (1.3). Ее размер в зависимости от ограничений на управление может составить от rN до $2rN$.
3. Решение ИЗЛП одним из методов линейного программирования, например, адаптивным методом. На выходе имеем последовательность компонент вектора управлений $u_{i,k}$, $k = \overline{1, N}$, $i = \overline{1, r}$.
4. Система (1) замыкается вектором $u^1 = (u_{1,1}, u_{2,1}, \dots, u_{r,1})^T$ и интегрируется на отрезке $[0, h]$ с начальным условием $x(0) = x_*$. В результате имеем оптимальное движение $x^0(t)$ на отрезке $[0, h]$. Точка $x^0(h)$ служит начальным положением для следующего шага.

Шаг j

1. Введем обозначения: $\tau = (j - 1)h$, $\chi = x^0((j - 1)h)$. Строятся матрицы A_χ и вектор c_χ подстановкой χ вместо ξ в A_ξ и c_ξ . Получаем вспомогательную линейную задачу:

$$\int_\tau^T \|u(t)\|_1 dt \rightarrow \min_u,$$

$$\dot{x}(t) = A_\chi x(t) + Bu(t) + c_\chi,$$

$$x(\tau) = \chi, \quad x(T) = x^*,$$

$$l_i^{(1)} \leq u_i(t) \leq l_i^{(2)}, \quad i = \overline{1, r}.$$

2. Сведение полученной системы к ИЗЛП вида (1.3). Ее размер от $r(N - j + 1)$ до $2r(N - j + 1)$.
3. Решение ИЗЛП методами линейного программирования. На выходе имеем последовательность компонент вектора управлений $u_{i,k}$, $k = \overline{j, N}$, $i = \overline{1, r}$.
4. Система (1) замыкается вектором $u^j = (u_{1,j}, u_{2,j}, \dots, u_{r,j})^T$ и интегрируется на отрезке $[(j-1)h, jh]$ с начальным условием $x((j-1)h) = \chi$. В результате имеем оптимальное движение $x^0(t)$ на отрезке $[(j-1)h, jh]$.

Результат. После N шагов имеем оптимальное кусочно-постоянное управление $u^0(t) = u^0((k-1)h) = u^k$, $t \in [(k-1)h, kh)$, $k = \overline{1, N}$ и оптимальную траекторию $x^0(t)$ на всем отрезке $[0, T]$.

Численная реализация Данный алгоритм был реализован в среде Matlab в виде блока программ. Для простоты используется встроенная функция решения задачи линейного программирования `linprog` вместо адаптивного метода. Ниже представлен список скриптов и функций.

- *nonlin.m* — основной скрипт. В начале задаются входные данные, перечисленные в начале пункта. Программа в цикле от 1 до N использует функцию решения линейной задачи *lin.m*.
- *lin.m* — решение линейной задачи оптимального управления. Входными параметрами являются $A, b, c, x_*, x^*, T, l^{(1)}, l^{(2)}, N$ из задачи (1.1). Использует функции сведения к ИЗЛП, решения системы дифференциальных уравнений, вычисления определенного интеграла. На выходе программы — оптимальное управление и оптимальное движение.

- *reduction.m* — сведение к ИЗЛП по алгоритму пункта 1.1. На вход подаются A, b, c, x_*, x^*, T, N . На выходе γ, D, g из уравнения (1.2).
- *runge45.m* — решение задачи Коши классическим методом Рунге–Кутты.
- *simpson.m* — вычисление определенного интеграла методом Симпсона.

Коды программ выставлены в разделе «Приложение».

Глава 3. Примеры приложения модифицированного метода

В этом разделе описаны приложения метода к различным нелинейным моделям. В первом пункте рассмотрена модель математического маятника. На его примере показаны свойства сходимости метода: уменьшение отклонения финального положения системы от желаемого состояния и уменьшение значения функционала качества при увеличении числа N .

Во втором пункте продемонстрирована работа с билинейной системой на примере вращения искусственного спутника Земли. Показана работа метода при вмешательстве внешних возмущений.

3.1. Управление маятником

Поставим задачу успокоения математического маятника за 6 секунд оптимальным по расходу топлива образом:

$$J(u) = \int_0^6 |u(t)| dt \rightarrow \min_u,$$

$$\ddot{x} + \sin x = u,$$

$$-4 \leq u(t) \leq 4,$$

$$x(0) = \pi, \quad \dot{x}(0) = 1,$$

$$x(6) = \dot{x}(6) = 0.$$

После перехода от уравнения второго порядка к системе из двух уравнений первого порядка, получим функцию $f(x) = (x_2, -\sin(x_1))^T$ и вектор $B = (0, 1)^T$.

Выпишем матрицу A_ξ и вектор c_ξ для произвольного вектора ξ :

$$A_\xi = \begin{pmatrix} 0 & 1 \\ -\cos \xi_1 & 0 \end{pmatrix}, \quad c_\xi = \begin{pmatrix} 0 \\ \xi_1 \cos \xi_1 - \sin \xi_1 \end{pmatrix}.$$

Для снятия знака модуля при сведении вспомогательных линейных задач к ИЗЛП сделаем замену $u = v^{(2)} - v^{(1)}$.

Выпишем линейную задачу для первого шага процесса:

$$\int_0^6 (v^{(1)}(t) + v^{(2)}(t)) dt \rightarrow \min_u,$$

$$\dot{x}_1 = x_2,$$

$$\dot{x}_2 = x_1 - v^{(1)} + v^{(2)} - \pi,$$

$$0 \leq v^{(1)}(t) \leq 4,$$

$$0 \leq v^{(2)}(t) \leq 4,$$

$$x_1(0) = \pi, \quad x_2(0) = 1,$$

$$x_1(6) = x_2(6) = 0.$$

Задача была решена в среде Matlab алгоритмом пункта 2.4 при различных значениях величины N .

Продемонстрируем полученные результаты на графиках. На рис. 2 проиллюстрированы фазовые портреты при $N = 4; 8; 20; 60$.

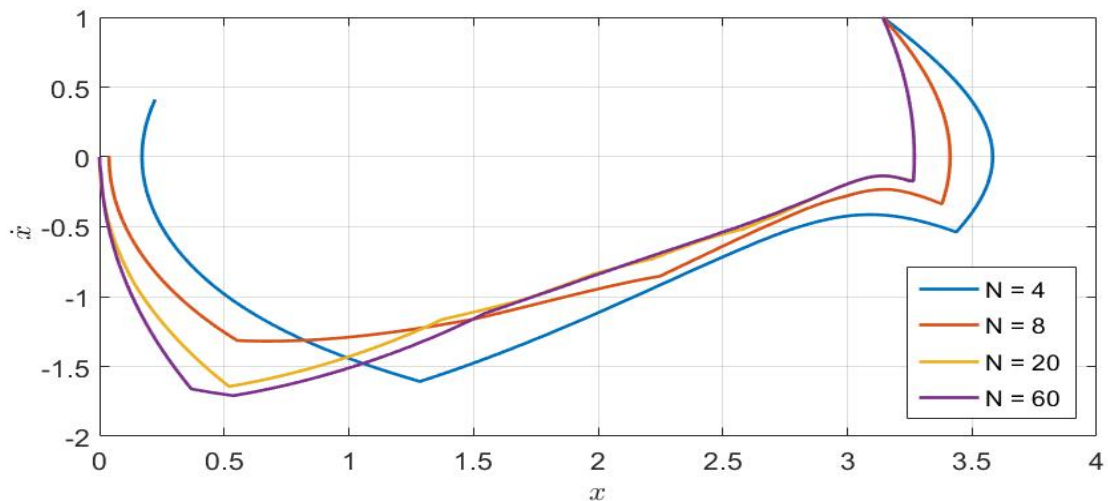


Рис. 2: Фазовые траектории маятника при различных N

На рис. 3 изображена зависимость управления от времени при тех же значениях N .

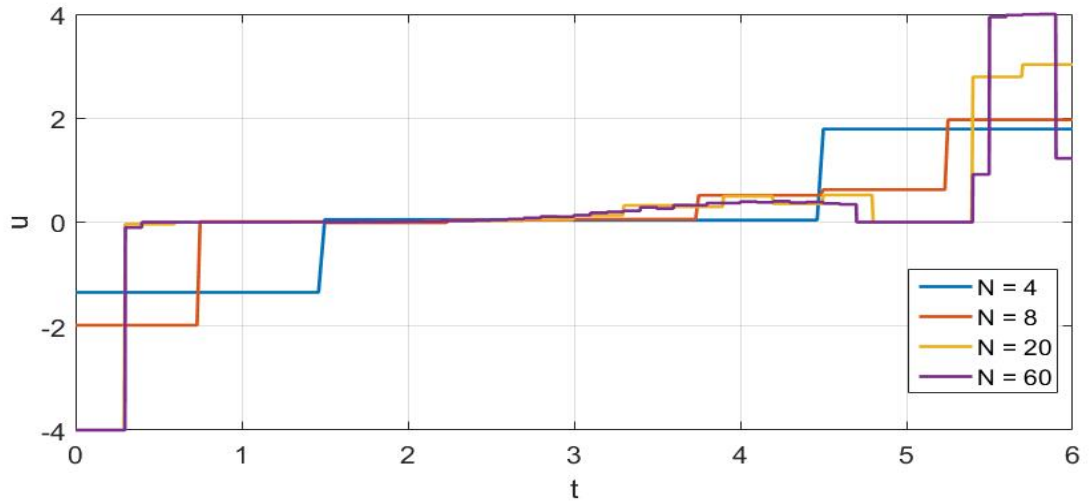


Рис. 3: Оптимальное управление при различных N

Важно продемонстрировать предельные свойства метода. Так на рис. 4 в логарифмическом масштабе представлено отклонение нормы финального положения системы $x^0(6)$ от желаемого состояния x^* в зависимости от N , изменяющегося от 2 до 60.

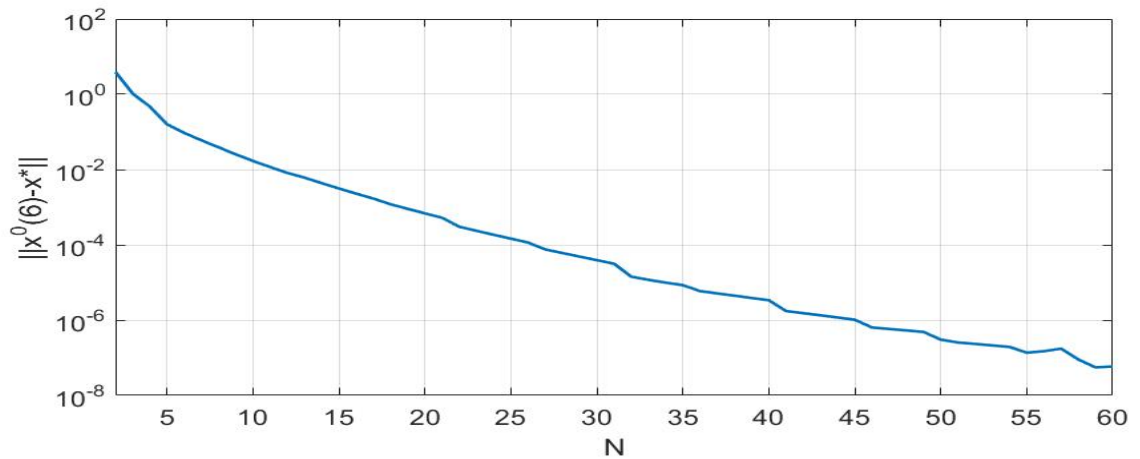


Рис. 4: Отклонение финального положения $x^0(6)$ от x^*

Рис. 5 иллюстрирует сходимость значения функционала качества $J(u)$ при увеличении значения N .

При $N = 60$ значение интеграла от модуля управления $J(u) = 3,600$, отклонение $\|x^0(t) - x^*\| = 5,87 \cdot 10^{-8}$.

Наконец, на рис. 6 покажем график нормы отклонения $z(t)$ реальной

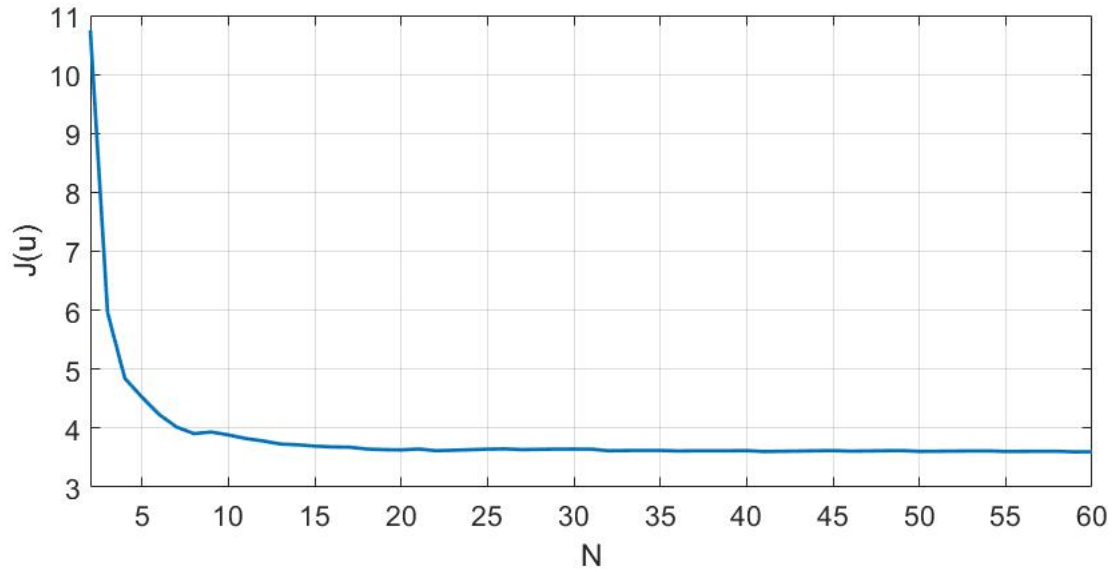


Рис. 5: Сходимость значения $J(u)$

траектории $x^0(t)$ от решений соответствующих вспомогательных линейных систем при $N = 60$.

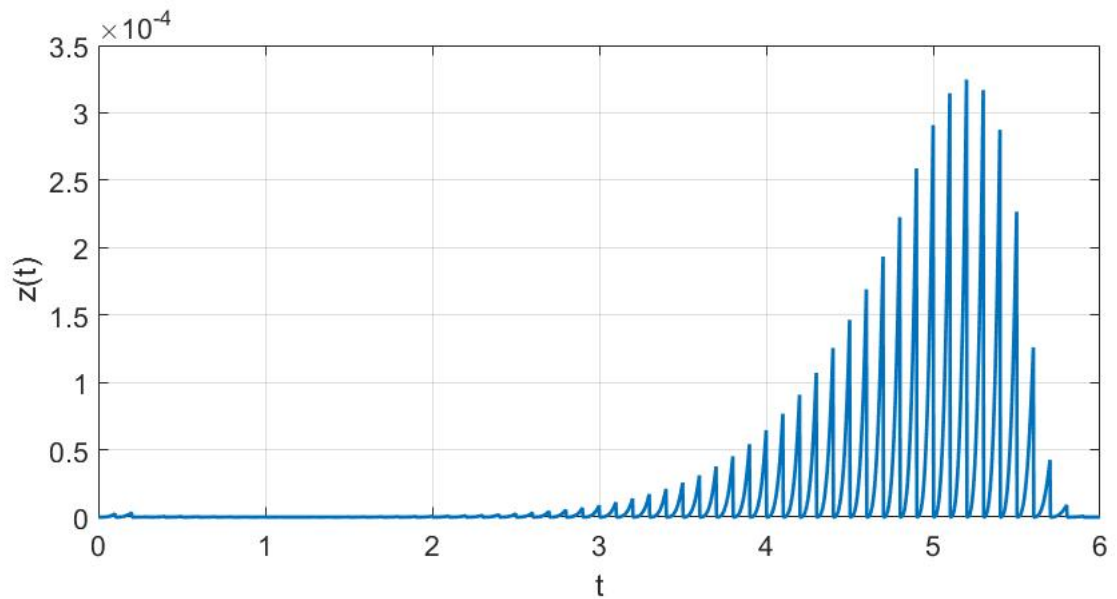


Рис. 6: Отклонение $z(t)$

3.2. Демпфирование колебаний спутника

Рассмотрим задачу оптимального демпфирования колебаний искусственного спутника Земли (ИСЗ). Согласно [15] вращение ИСЗ вокруг соб-

ственной оси может быть описано динамическими уравнениями Эйлера:

$$I_A \dot{p} + (I_C - I_B)qr = a_1 u_1,$$

$$I_B \dot{q} + (I_A - I_C)rp = a_2 u_2,$$

$$I_C \dot{r} + (I_B - I_A)pq = a_3 u_3.$$

В этих уравнениях I_A, I_B, I_C — главные центральные моменты инерции, p, q, r — проекции угловой скорости на главные центральные оси инерции. Предполагается, что спутник управляется по средствам тяг u_1, u_2, u_3 , создаваемых тремя двигателями, a_1, a_2, a_3 — плечи приложения сил.

Ставится задача подобрать управление таким образом, чтобы за заданное время T система перешла из некоторого начального положения в точку $(0, 0, 0)^T$ оптимальным по расходу топлива образом.

После ряда преобразований получим систему

$$\int_0^1 (|u_1| + |u_2| + |u_3|) dt \rightarrow \min,$$

$$\begin{cases} \dot{x}_1 = b_1 x_2 x_3 + c_1 u_1, \\ \dot{x}_2 = b_2 x_1 x_3 + c_2 u_2, \\ \dot{x}_3 = b_3 x_1 x_2 + c_3 u_3, \\ x(0) = x^0, \quad x(1) = 0, \\ |u_i| \leq L. \end{cases}$$

Здесь фазовые переменные $x = (x_1, x_2, x_3)^T = (pT, qT, rT)^T$, $b_1 = \frac{I_C - I_B}{I_A}$, $b_2 = \frac{I_A - I_C}{I_B}$, $b_3 = \frac{I_B - I_A}{I_C}$, $c_1 = \frac{a_1 T^2}{I_A}$, $c_2 = \frac{a_2 T^2}{I_B}$, $c_3 = \frac{a_3 T^2}{I_C}$, L — максимальная тяга двигателей.

Задача была решена алгоритмом, описанным в 2.4 при следующих значениях параметров: $b_1 = -0,25$, $b_2 = -2$, $b_3 = 1,5$, $c_1 = 1$, $c_2 = 4$, $c_3 = 2$, $x^0 = (1, 2, -1)^T$, $L = 3$, $N = 10$. Результат решения задачи без влияния возмущений продемонстрирован на рис. 7, 8. Функционал качества $J(u^0) = 2,097$, отклонение от желаемого состояния $\|x^0(1)\| = 8,6 \cdot 10^{-5}$.

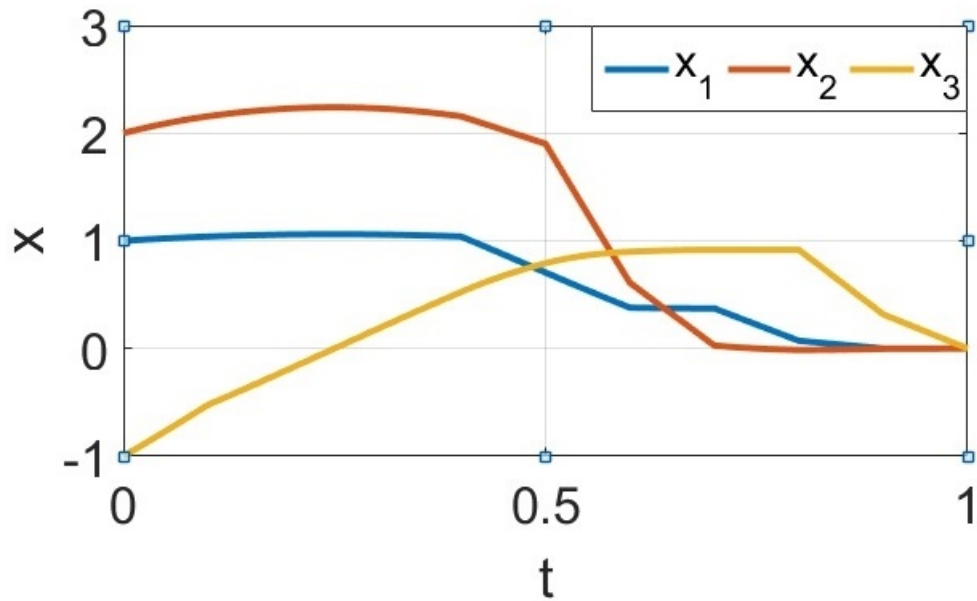


Рис. 7: Траектория $x^0(t)$ для системы без возмущения

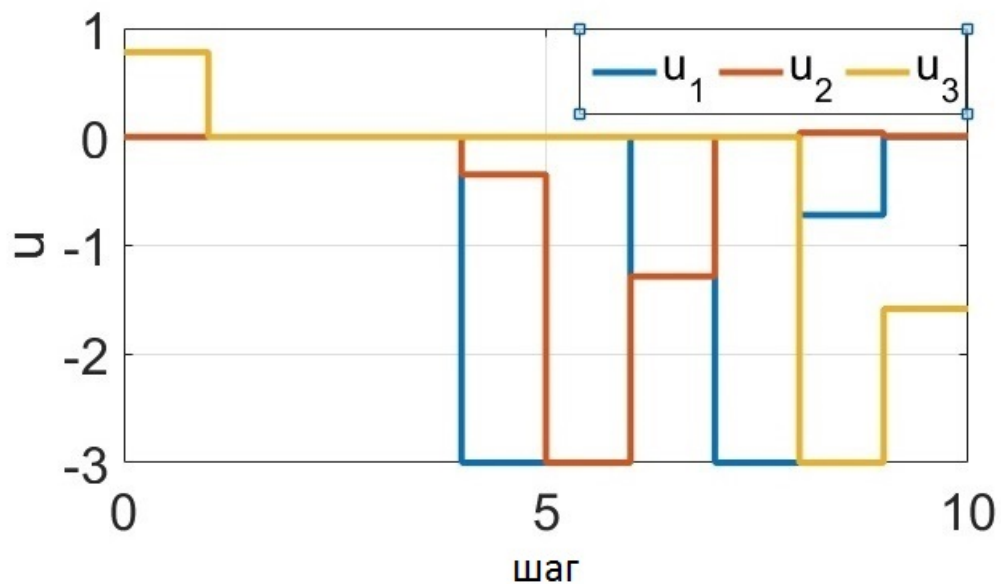


Рис. 8: Управление $u^0(t)$ для системы без возмущения

Прибавим к правой части системы дифференциальных уравнений функцию

$$g(t) = \left(4(1 - t) \sin^2(10\pi t), -3 \sin(4\pi t), -8 \cos(20\pi t) \right)^T.$$

В результате для возмущенной системы получим значение минимизируемого функционала $J(u^0) = 3,081$, а отклонение от начала координат $\|x^0(1)\| = 0,16$. Динамика изменения фазовых переменных $x^0(t)$ и управ-

ления $u^0(t)$ проиллюстрирована на рис. 9, 10.

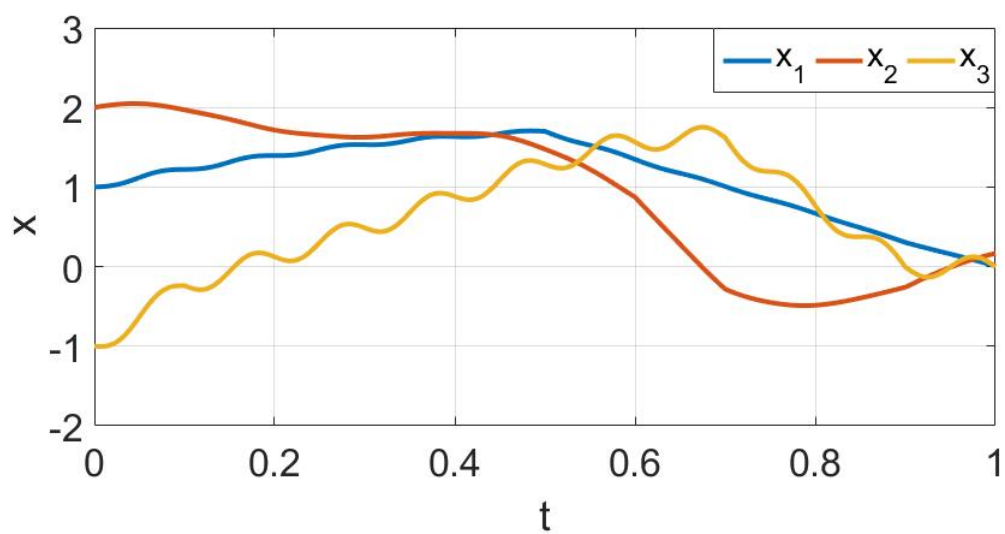


Рис. 9: Траектория $x^0(t)$ для системы с возмущением

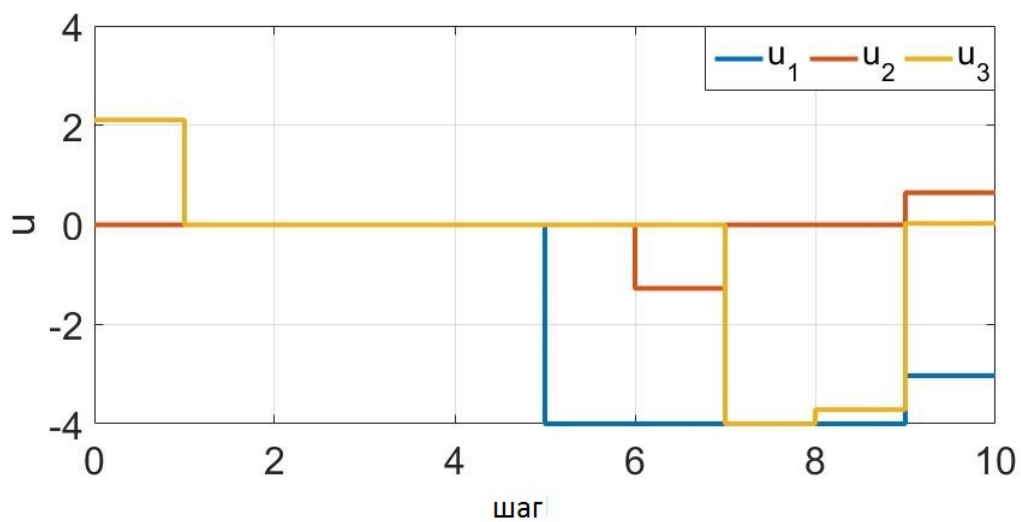


Рис. 10: Управление $u^0(t)$ для системы с возмущением

Заключение

К результатам данной выпускной квалификационной работы можно отнести следующие пункты:

- рассмотрена структура множеств достижимости и управляемости линейной задачи оптимального управления (1.1), предложены способы построения внутренней и внешней оценок;
- изучен метод построения оптимального позиционного управления для нелинейной задачи (1)–(5), предложенный Р. Ф. Габасовым,
- разработана модификация этого метода, основанная на последовательном решении линейных задач с учетом текущего состояния;
- показано, что наиболее точный результат работы модифицированного метода получается при линеаризации системы в начальной точке движения на каждом элементарном отрезке времени;
- доказана сходимости к желаемому состоянию системы в конечный момент времени при увеличении частоты дискретизации кусочно-постоянного управления;
- выведены достаточные условия на правую часть системы, при которых решение окажется в заданной окрестности терминального состояния;
- алгоритм метода реализован в виде блока программ в среде Matlab;
- модифицированный метод апробирован на задаче управления математическим маятником, на примере этой задачи проиллюстрированы предельные свойства метода;
- продемонстрирована работа метода на примере управления вращением искусственного спутника Земли вокруг своей оси, рассмотрен случай

управления при воздействии внешних возмущений;

- принято участие на 7 конференциях с последующей публикацией. Три из них проиндексированы в РИНЦ [8, 10, 11], три — в Scopus [9, 13, 14].

Список литературы

- [1] Калман Р. Е. Об общей теории систем управления. Труды I Международного конгресса ИФАК. М.: Изд-во АН СССР, 1961. Т. 2. С. 521–547.
- [2] Понтрягин Л. С., Болтянский В. Г., Гамкрелидзе Р. В., Мищенко Е. Ф. Математическая теория оптимальных процессов. М.: Наука, 1969. 384 с.
- [3] Беллман Р., Гликсберг И., Гросс О. Некоторые вопросы математической теории процессов управления. М.: ИЛ, 1962. 336 с.
- [4] Зубов В. И. Математические методы исследования систем автоматического регулирования. Л.: Машиностроение, 1974. 336 с.
- [5] Балашевич Н. В., Габасов Р., Кириллова Ф. М. Численные методы программной и позиционной оптимизации линейных систем управления // Журн. вычисл. математики и мат. физики. 2000. Вып. 40, №6.
- [6] Альсевич В. В., Габасов Р., Глушенков В. С. Оптимизация линейных экономических моделей. Минск: Изд-во БГУ, 2000. 211 с.
- [7] Габасов Р., Кириллова Ф. М., Ружицкая Е. А. Демпфирование и стабилизация маятника при больших начальных возмущениях. // Изв. РАН. Теория и системы управл. 2001. №1. С. 29–38.
- [8] Попков А. С., Баранов О. В. Об оптимальном управлении вращательным движением вала электродвигателя // Процессы управления и устойчивость. 2014. Т. 1(17). №1 С. 31–36.
- [9] Popkov A. S., Baranov O. V., Smirnov N. V. Application of adaptive method of linear programming for technical objects control // 2014 International Conference on Computer Technologies in Physical and Engineering Applications (ICCTPEA), 2014. P. 141–142.

- [10] Попков А. С. Идентификация динамической модели межотраслевого баланса для экономики России и оптимальное распределение инвестиций на ее основе // Процессы управления и устойчивость. 2015. Т. 2(18). №1 С. 696–701.
- [11] Белоусова М. В., Попков А. С. Построение динамической модели МОБ на основе WIOD // Процессы управления и устойчивость. 2016. Т. 3(19). №1. С. 601–606.
- [12] Popkov A. Multicriteria Regulation of Investments in the Economy of the Russian Federation // Proceedings of the International Workshop on Applications in Information Technology (IWAIT-2015), The University of Aizu Press, 2015. P. 68–70.
- [13] Popkov A. S., Smirnov N. V., Baranov O. V. Real-time quadcopter optimal stabilization // "Stability and Control Processes" in Memory of V.I. Zubov (SCP), 2015 International Conference, 2015. P. 123–125.
- [14] Popkov A. S. Application of the adaptive method for optimal stabilization of a nonlinear object // 2016 International Conference Stability and Oscillations of Nonlinear Control Systems (Pyatnitskiy's Conference), 2016. P. 1–3.
- [15] Бабаджанянц Л. К., Потоцкая И. М. Управление по критерию расхода в механических системах. СПб.: С.-Петербург. гос. ун-т, 2003. 137 с.

Приложение

В этом разделе представлен код программ, описанных в пункте 2.4.

nonlin.m

Основной скрипт. В начале задаются входные данные, перечисленные в начале пункта 2.4. Программа в цикле от 1 до N использует функцию решения линейной задачи.

```
1 clear all; clc;
2
3 x0 = [pi; 1];
4 x1 = [0; 0];
5 t0 = 0;
6 t1 = 6;
7 F = @(x,u)[x(2); -sin(x(1))+u];
8 A = @(xs)[0 1; -cos(xs(1)) 0];
9 b = [0; 1];
10 c = @(xs)[0; xs(1)*cos(xs(1))-sin(xs(1))];
11 g = @(t)0;
12 L = 4;
13
14 T = t0;
15 Y = x0;
16 Y1 = x0;
17 U = [];
18 for i = 1:N
19     k = (t1-t0)/(N*(t1-T(end)));
20     k1 = k*0;
21     xs = Y(:,end)*(1-k1)+x1*k1;
22     [tt,xx,uu,u,w,flag] = lin(A(xs),b,c(xs),Y(:,end),x1,T(end),t1,L,N+1-i);
23     [t,y] = ode45(@(t,x)(F(x,u(:,1))+g(t))],[T(end), T(end)+(t1-t0)/N], Y(:,end));
24     [t,y1] = ode45(@(t,x)(A(xs)*x+b*u(:,1)+c(xs)+g(t))],[T(end), T(end)+(t1-t0)/N], Y(:,end));
25     T = [T, t(2:end)'];
26     Y = [Y, y(2:end,:)'];
27     Y1 = [Y1, y1(2:end,:)'];
28     U = [U, u(:,1)*ones(1,length(t)-1)];
29     u1(:,i) = u(:,1);
30 end
31 z = zeros(1,length(Y));
32 for i = 1:length(Y)
33     z(i) = norm(Y1(:,i)-Y(:,i));
34 end
35 U = [U, u(:,1)];
```

lin.m

Решение линейной задачи оптимального управления. Входными параметрами являются $A, b, c, x_*, x^*, T, l^{(1)}, l^{(2)}, N$ из задачи (1.1). Использует функции сведения к ИЗЛП, решения системы дифференциальных уравнений, вычисления определенного интеграла. На выходе программы — оптимальное управление и оптимальное движение.

```
1 function [t1,x1,u1,u,w,flag] = lin(A,b0,d,x0,g,Tstart,Tend,L,N)
2
3 b = [-b0 b0];
4 H = eye(2);
5 L1 = zeros(2,1);
6 L2 = L*ones(2,1);
7 delta = 1e-10;
8 epsilon = 1e-7;
9 q = 1;
10 % q = 0, если максимизируем x(Tend), q = 1, если минимизируем интеграл от управления
11 f = 0;
12 % f = 0, если план не известен, f = 1, если известен
13 qf = 0;
14 % qf = 0, если опора не известна, qf = 1, если известна
15 st = 0;
16 % st = 0, если матрицы A и b стационарны
17
18
19 l = length(L1);
20 for i = 1:l
21     for j = 1:N
22         LL1(j+(i-1)*N,1) = L1(i,1);
23         LL2(j+(i-1)*N,1) = L2(i,1);
24     end
25 end
26 if q == 0
27     if st == 0
28         [D1, g0, c1] = reduction_st(A, b, d, Tstart, Tend, c, x0, H, g, N, delta);
29     else
30         [D1, g0, c1] = reduction(A, b, d, Tstart, Tend, c, x0, H, g, N, delta);
31     end
32 elseif q == 1
33     if st == 0
34         [D1, g0, c1] = reduction1_st(A, b, d, Tstart, Tend, x0, H, g, N, delta);
35     else
36         [D1, g0, c1] = reduction1(A, b, d, Tstart, Tend, x0, H, g, N, delta);
37     end
38 end
```

```

39
40 [u,~,flag] = linprog(-c1,[],[],D1,g0,LL1,LL2);
41 for i = 1:l
42     w(1:(length(u)/l),i) = u(((i-1)*length(u)/l+1):(i*length(u)/l),1);
43 end
44 w = w';
45 x1 = x0;
46 t1 = Tstart;
47
48 syms t
49 if st == 0
50     N1 = 100;
51     Tn = (Tend-Tstart)/(N*N1);
52     for i = 1:N
53         for j = 1:N1
54             t1(N1*(i-1)+j+1) = Tstart + (N1*(i-1)+j)*Tn;
55             w1(:,N1*(i-1)+j) = w(:,i);
56             x1(:,N1*(i-1)+j+1) = expm(A*Tn)*x1(:,N1*(i-1)+j) + ...
57                 simpson(0,Tn,expm(A*Tn)*(b*w1(:,N1*(i-1)+j)+d),expm(A*Tn/2)*...
58                     (b*w1(:,N1*(i-1)+j)+d),b*w1(:,N1*(i-1)+j)+d);
59         end
60     end
61     w1(:,N1*N+1) = w(:,N);
62 else
63     for i = 1:length(w)
64         [x2, t2] = runge45(A, b*w(:,i), t1(length(t1)), x1(:,length(t1)), ...
65             (Tend-Tstart)/N, delta);
66     end
67     x1 = [x1 x2(:,2:length(t2))];
68     t1 = [t1 t2(2:length(t2))];
69
70 end
71 u = w(2,:)-w(1,:);
72 u1 = w1(2,:)-w1(1,:);

```

reduction.m

Сведение к ИЗЛП по алгоритму пункта 1.1. На вход подаются A , b , c , x_* , x^* , T , N . На выходе γ , D , g из уравнения (1.2).

```

1 function [D1, g0, c1] = reduction(A, b, Tstart, Tend, x0, H, g, N, delta)
2 t = sym('t');
3 n = size(H);
4 m = size(b);
5 bb1 = zeros(m(1),1);
6
7 A1 = (-1)*A';
8 h = 0.5*(Tend - Tstart)/N;

```

```

9  for j = 1:n(1)
10     zi(:,1,j) = H(j,:)';
11  end
12  t1(1,1) = Tend;
13  for j = 1:m(2)
14     b1(:,1,j) = double(subs(b(:,j), 't', Tend));
15  end
16  for i = 1:(2*N)
17     t1(i+1,1) = Tend - i*h;
18     for j = 1:m(2)
19         b1(:,i+1,j) = double(subs(b(:,j), 't', t1(i,1)));
20     end
21     for j = 1:n(1)
22         zi(:,i+1,j) = runge45(A1, bb1, t1(i,1), zi(:,i,j), -h, delta);
23     end
24  end
25  G = zeros(n(2),n(1),2*N+1);
26  for i = 1:(2*N+1)
27     for j = 1:n(1)
28         G(:,j,i) = zi(:,i,j);
29     end
30  end
31  g0 = g - H*runge45(A, bb1, Tstart, x0, (Tend - Tstart), delta*1e4);
32
33
34  for j = 1:m(2)
35     for k = N:-1:1
36         c1(:,N-k+1+(j-1)*N) = -2*h;
37         D1(:,N-k+1+(j-1)*N) = simpson(t1(2*k+1,1), t1(2*k-1,1), ...
38             G(:, :, 2*k+1)'*b1(:,2*k+1,j), G(:, :, 2*k)'*b1(:,2*k,j), ...
39             G(:, :, 2*k-1)'*b1(:,2*k-1,j));
40     end
41  end
42  c1 = c1';

```

runge45.m

Решение задачи Коши классическим методом Рунге–Кутты.

```

1  function [x1, t1] = runge45(A, b, t0, x0, h, delta)
2  t = sym('t');
3  s = 5;
4  f = double(subs(A*x0+b, t0));
5  d = (1/max(abs(t0), abs(t0+h)))^(s+1) + norm(f)^(s+1);
6  h0 = (delta/d)^(1/(s+1));
7  k01 = h0*f;
8  k02 = h0*double(subs(A*(f+0.5*k01)+b, t0+0.5*h0));
9  k03 = h0*double(subs(A*(f+0.5*k02)+b, t0+0.5*h0));
10 k04 = h0*double(subs(A*(f+k03)+b, t0+h0));

```

```

11 x00 = x0 + (1/6)*(k01 + 2*k02 + 2*k03 + k04);
12 f1 = double(subs(A*x00+b,t0+h0));
13 d1 = (1/max(abs(t0+h0),abs(t0+h)))^(s+1) + norm(f1)^(s+1);
14 h01 = (delta/d1)^(1/(s+1));
15 h1(1) = min(h0, h01)*sign(h);
16 z = x0;
17 z1 = x0;
18 tt = t0;
19 i = 1;
20 x1 = 0;
21 while sign(h)*(t0 + h) > sign(h)*(tt(i))
22     k1(:,i) = h1(i)*double(subs(A*z(:,i)+b, tt(i)));
23     k2(:,i) = h1(i)*double(subs(A*(z(:,i)+0.5*k1(:,i))+b, tt(i)+0.5*h1(i)));
24     k3(:,i) = h1(i)*double(subs(A*(z(:,i)+0.5*k2(:,i))+b, tt(i)+0.5*h1(i)));
25     k4(:,i) = h1(i)*double(subs(A*(z(:,i)+k3(:,i))+b, tt(i)+h1(i)));
26     z(:,i+1) = z(:,i) + (1/6)*(k1(:,i) + 2*k2(:,i)+ 2*k3(:,i) + k4(:,i));
27     z1(:,i+1) = z1(:,i) + 0.5*(-k1(:,i) + 2*k2(:,i)+ 2*k3(:,i) - k4(:,i));
28     ro = z(:,i+1) - z1(:,i+1);
29     if norm(ro) > delta*2^s
30         h1(i) = 0.5*h1(i);
31         i = i-1;
32     elseif norm(ro) > delta && norm(ro) <= delta*2^s
33         h1(i+1) = 0.5*h1(i);
34         tt(i+1) = tt(i) + h1(i);
35     elseif norm(ro) >= delta*2^(-s-1) && norm(ro) <= delta
36         h1(i+1) = h1(i);
37         tt(i+1) = tt(i) + h1(i);
38     elseif norm(ro) < delta*2^(-s-1)
39         h1(i+1) = 2*h1(i);
40         tt(i+1) = tt(i) + h1(i);
41     end
42     i = i+1;
43 end
44 tt(i) = t0+h;
45 h1(i-1) = tt(i)-tt(i-1);
46 k1(:,i-1) = h1(i-1)*double(subs(A*z(:,i-1)+b, tt(i-1)));
47 k2(:,i-1) = h1(i-1)*double(subs(A*(z(:,i-1)+0.5*k1(:,i-1))+b, tt(i-1)+0.5*h1(i-1)));
48 k3(:,i-1) = h1(i-1)*double(subs(A*(z(:,i-1)+0.5*k2(:,i-1))+b, tt(i-1)+0.5*h1(i-1)));
49 k4(:,i-1) = h1(i-1)*double(subs(A*(z(:,i-1)+k3(:,i-1))+b, tt(i-1)+h1(i-1)));
50 z(:,i) = z(:,i-1) + (1/6)*(k1(:,i-1) + 2*k2(:,i-1)+ 2*k3(:,i-1) + k4(:,i-1));
51 z1(:,i) = z1(:,i-1) + 0.5*(-k1(:,i-1) + 2*k2(:,i-1)+ 2*k3(:,i-1) - k4(:,i-1));
52 if nargout == 1
53     x1 = z(:,i);
54 else
55     x1 = z;
56     t1 = tt;
57 end

```


`simpson.m`

Вычисление определенного интеграла методом Симпсона.

```
1 function I = simpson(a,b,x1,x2,x3)
2 I = ((b-a)/6)*(x1 + 4*x2 + x3);
3 end
```