МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
Кафедра методов оптимального управления

СЕЛЯХ
Никита Евгеньевич

СИНТЕЗ ОПТИМАЛЬНЫХ СИСТЕМ И УПРАВЛЕНИЙ ПО ПРОГНОЗИРУЮЩЕЙ МОДЕЛИ НА ОСНОВЕ МЕТОДОВ ПАРАМЕТРИЧЕСКОГО ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Дипломная работа

Научный руководитель
канд. физ.-мат. наук
доцент Н.М. Дмитрук

Допущена к защите

«_____» ____________________ 2020 г.
Зав. кафедрой методов оптимального управления
канд. физ.-мат. наук, доцент Н.М. Дмитрук

Минск 2020

ОГЛАВЛЕНИЕ

С.

РЕФЕРАТ

Дипломная работа, 62 с., 20 рис., 14 источников

ОПТИМАЛЬНОЕ УПРАВЛЕНИЕ, УПРАВЛЕНИЕ ПО ПРОГНОЗИРУЮЩЕЙ МОДЕЛИ, ОБРАТНАЯ СВЯЗЬ, ПРОБЛЕМА СИНТЕЗА, АЛГОРИТМ, МНОГОПАРАМЕТРИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

Объект исследования — классическая задача синтеза оптимальных линейных систем управления.

Цель работы — предложить методы синтеза оптимальных обратных связей в линейной задаче оптимального управления на основе идей параметрического программирования, по аналогии с подходами, использующимися в явных схемах управления по прогнозирующей модели.

Методы решения — методы управления по прогнозирующей модели, методы оптимального управления, многопараметрического линейного программирования; программный пакет MPT для решения задач многопараметрического программирования и оптимального управления.

Основные результаты работы — на основе идей и подходов из явных методов управления по прогнозирующей модели, предложены идеи и методы построения оптимальных обратных связей на основе параметрического линейного программирования. Результатом является синтез оптимальных обратных управлений в линейно задаче оптимального управления с ограничениями в явном, аналитическом виде. Предложенный метод проиллюстрирован не примерах решения задач стабилизации и задачи оптимального управления линейными системами, с использованием пакета MPT.

РЭФЕРАТ

Дыпломная работа, 62 с., 20 мал., 14 крыніц

АПТЫМАЛЬНАЕ КИРАВАННЕ, КИРАВАННЕ ПА ПРАГНАЗЫИ?НАИ? МАДЕЛИ, ЗВАРОТНАЯ СУВЯЗЬ, ПРАБЛЕМА СІНТЭЗУ, АЛГАРЫТМ, ШМАТПАРАМЕТРЫЧНЫХ ПРАГРАМАВАННЕ

Аб'ект даследавання — класічная задача сінтэзу аптымальных лінейных сістэм кіравання.

Мэта працы — прапанаваць метады сінтэзу аптымальных зваротных сувязяў у лінейнай задачы аптымальнага кіравання на аснове ідэй параметрычнага праграмавання, па аналогіі з падыходамі, якія выкарыстоўваюцца ў відавочных схемах кіравання па прагназуючай мадэлі.

Метады рашэння — метады кіравання па прагназуючай мадэлі, метады аптымальнага кіравання, многопараметрического лінейнага праграмавання; праграмны пакет MPT для вырашэння задач многопараметрического праграмавання і аптымальнага кіравання.

Асноўныя вынікі працы — на аснове ідэй і падыходаў з відавочных метадаў кіравання па прагназуючай мадэлі, прапанаваны ідэі і метады пабудовы аптымальных зваротных сувязяў на аснове параметрычнага лінейнага праграмавання. Вынікам з'яўляецца сінтэз аптымальных зваротных упраўленняў ў лінейнай задачы аптымальнага кіравання з абмежаваннямі ў відавочным, аналітычным выглядзе. Прапанаваны метад праілюстраваны на прыкладах рашэння задач стабілізацыі і задачы аптымальнага кіравання лінейнымі сістэмамі, з выкарыстаннем пакета MPT.

ABSTRACT

Graduation work, 62 c., 20 fig., 14 sources

OPTIMAL CONTROL, MODEL PREDICTIVE CONTROL, FEEDBACK, SYNTHESIS PROBLEM, ALGORITHM, MULTI-PARAMETER PROGRAMMING

Object of research — classical problem of synthesis of optimal linear control systems.

The aim of this work is to propose methods for synthesizing optimal feedbacks in a linear optimal control problem based on the ideas of parametric programming, by analogy with the approaches used in explicit control schemes based on a predictive model.

Solution methods — predictive model control methods, optimal control methods, multiparametric linear programming; MPT software package for solving problems of multiparametric programming and optimal control.

Main results — based on ideas and approaches from explicit control methods based on the predictive model, the ideas and methods for constructing optimal feedbacks based on parametric linear programming are proposed. The result is a synthesis of optimal inverse controls in a linear optimal control problem with constraints in an explicit, analytical form. The proposed method is illustrated by examples of solving stabilization problems and optimal control problems for linear systems using the MPT package.

ВВЕДЕНИЕ

Существуют два взгляда на теорию оптимального управления. Согласно одному из них, возникшему после открытия принципа максимума Понтрягина, теория оптимального управления — раздел современного (неклассического) вариационного исчисления. Другой взгляд трактует теорию оптимального управления как раздел современной теории управления, представляющей естественное развитие классической теории управления. В соответствии с этим в первом случае под управлениями понимаются элементы функциональных пространств, по которым ищется экстремум выбранного функционала качества, и основным вопросом теории считается анализ решения экстремальной задачи (существование, единственность, непрерывная зависимость решений, необходимые и достаточные условия оптимальности и т.п.).

Во втором случае управление — это процесс, в котором для достижения нужного поведения объекта управления в каждый текущий момент времени создаются целенаправленные (управляющие) воздействия на объект управления в зависимости от доступной к этому моменту информации о поведении объекта и действующих на него возмущениях.

Управление называется программным, если (программные) управляющие воздействия (программы) планируются по априорной информации до начала процесса управления и не корректируются в процессе управления. При позиционном управлении (позиционные) управляющие воздействия создаются в процессе управления по текущим позициям, которые аккумулируют информацию, доступную к текущему моменту.

Построение оптимальных позиционных управляющих воздействий называется синтезом оптимальных систем управления и является основным вопросом теории оптимального управления во втором случае.

В данной дипломной работе будет рассмотрена классическая задача синтеза. При построении оптимальных обратных связей задача оптимального управления погружается в семейство задач, зависящее от позиции процесса управления — момента времени и состояния. Если принять позицию за параметры, а задачу рассматривать в классе дискретных управлений, то задача оптимального управления может рассматриваться как задача многопараметрического линейного программирования.

Параметрическое программирование — раздел теории математического программирования, занимающийся изучением задач оптимизации, содержащих параметры, и ставящий целью описание решение для всего пространства/заданного подмножества параметров (в отличие от анализа чувствительности, который изучаем малые изменения параметров).

Результаты и алгоритмы параметрического программирования привлекли внимание инженеров по управлению в связи с задачами стабилизации линейных систем ограниченными обратными связями на основе методов теории управления по прогнозирующей модели (Model Predictive Control — MPC). MPC в сочетании с многопараметрическим линейным и квадратичным программированием получило название явного MPC, поскольку решение задачи — стабилизирующая обратная связь — получается в явном виде.

В настоящей работе будут рассмотрены основные положения явного MPC, а затем идея построения обратных связей на основе параметрического программирования будут развиты на проблему синтеза оптимальной системы.

ГЛАВА 1

ОБЗОР ЛИТЕРАТУРЫ

В настоящей главе приводится обзор основных результатов теории оптимального управления. Сначала обсуждаются формулировки задач, возникающих при оптимизации динамических систем управления. Затем приводятся основные результаты качественной теории — принцип максимума Л.С. Понтрягина [2] и динамическое программирование Р. Беллмана [1].

В данной главе рассматриваются основные результаты теории оптимального управления, классический подход к построению обратных связей, а также оптимальное управление в реальном времени и реализация оптимальной обратной связи

Кроме того, обсуждаются современные методы управления в реальном времени: реализация оптимальных обратных связей в задачах оптимального управления с конечным горизонтом и теория управления по прогнозирующей модели для решения классической проблемы теории управления — стабилизации нелинейных динамических систем при наличии ограничений.

1.1 Основные результаты теории оптимального управления

Постановка любой конкретной задачи оптимального управления включает в себя 5 необходимых элементов: промежуток управления, математическую модель управляемого объекта, класс управлений и ограничений на них, ограничения на фазовую траекторию, критерий качества. Рассмотрим их подробнее.

1) Промежуток управления. Прежде всего задачи оптимального управления разделяются на непрерывные, рассматриваемые на некотором промежутке времени $T = [t_0, t_f]$, и дискретные, в которых динамический процесс рассматривается в дискретные моменты времени $k = 0, 1, ...N$, где N — натуральное число.

По продолжительности процесса различаются задачи с фиксированным и нефиксированным временем окончания процесса. Выделяются также задачи на бесконечном интервале.

2) Математическая модель. Динамика изучаемого процесса моделирует-

ся, как правило, дифференциальными (для непрерывных систем)

$$\dot{x}(t) = f(x(t), u(t), t), t \in [t_0, t_f], \tag{1.1}$$

или разностными уравнениями (для дискретных систем)

$$x(k+1) = f(x(k), u(k), k), k = 0, 1, ...,$$

где n-вектор x называется состоянием системы, r-вектор u называется управлением, функция $f : \mathbb{R} \times \ \mathbb{R}^r \times \mathbb{R} \to \mathbb{R}^n$ задана.

Число переменных состояния n называется порядком системы управления, число r — числом входов.

Далее будем рассматривать непрерывные системы вида (1.1).

3) Класс управлений и ограничения на них. Для непрерывного процесса управления указывается класс функций, из которого выбираются управления. Это могут быть: измеримые, дискретные, кусочно-непрерывные, гладкие, импульсные функции и т.д.

Кроме класса доступных управлений задается множество $U \subset \ \mathbb{R}$ — множество допустимых значений управления. Как правило, U — компакт в $\mathbb{R}^r$.

Далее будем рассматривать управления из класса кусочно-непрерывных функций.

Определение 1.1 Кусочно-непрерывная функция $u(\cdot) = (u(t), t \in [t_0, t_f])$ называется доступным управлением, если $u(t) \in U, t \in [t_0, t_f]$.

4) Ограничения на фазовую траекторию. Ограничения на переменные состояния могут накладываться:

- в начальный момент времени t_0:

$$x(t_0) \in X_0;$$

- в конечный момент времени t_f — такие ограничения называются терминальными:

$$x(t_f) \in X_f;$$

- в изолированные моменты $t_i \in [t_0, t_f], i = \overline{1, m}$, из промежутка управления — промежуточные фазовые ограничения:

$$X(t_i) \in X_i, i = \overline{1, m},$$

- на всем промежутке управления — фазовые ограничения:

$$x(t) \in X(t), t \in [t_0, t_f],$$

где $X_0, X_f, X_i, i = \overline{1, m}, X(t), t \in [t_0, t_f]$, — заданные подмножества пространства состояний.

Задача управления с $x(t_f) \in X_f$ называется:

- задачей со свободным правым концом траектории, если $X_f = \mathbb{R}^n$,
- задачей с закрепленным правым концом траектории, если $X_f = \{x_f\}$,
- задачей с подвижным правым концом траектории, если X_f содержит более одной точки и не совпадает с R^n.

Аналогичная классификация имеет место для задач с ограничениями на левый конец траектории $x_0 \in X_0$.

Выделяют также смешанные ограничения, учитывающие связи между переменными состояния и переменными управления:

$$(u(t), x(t)) \in S \subseteq \mathbb{R}^r \times \mathbb{R}^n, t \in [t_0, t_f[.$$

Определение 1.2 Доступное управление $u(\cdot)$ называется допустимым (или, программой), если оно порождает траекторию $x(\cdot) = (x(t), t \in [t_0, t_f]$, удовлетворяющую всем заданным ограничениям задачи.

5) Критерий качества. Качество допустимого управления оценивается так называемым критерием качества

Существуют четыре типа критерия качества:

i) критерий качества Майера (терминальный критерий)

$$J(u) = \varphi(x(t_f)),$$

ii) критерий качества Лагранжа (интегральный критерий)

$$J(u) = \int_{t_0}^{t_f} f_0(x(t), u(t), t)dt,$$

iii) критерий качества Больца

$$J(u) = \varphi(x(t_f)) + \int_{t_0}^{t_f} f_0(x(t), u(t), t)dt,$$

iv) задачи быстродействия (являются задачами с нефиксированной продолжительностью процесса).

$$J(u) = t_f - t_0 \to \min .$$

Определение 1.3 Допустимое управление $u^0(\cdot)$ называется оптимальным управлением (оптимальной программой), если на нем критерий качества достигает экстремального значения (min или max):

$$J(u^0) = extr J(u),$$

где минимум (максимум) берется по всем допустимым управлениям.

1.2 Принцип максимума и динамическое программирование

В теории оптимального управления существует два фундаментальных результата: принцип максимума Л.С. Понтрягина [2] и динамическое программирование Р. Беллмана. [1] Приведем эти результаты на примере простейшей задачи оптимального управления.

$$\begin{gathered}
J(u) = \varphi(x(t_f)) + \int_{t_0}^{t_f} f_0(x(t), u(t), t)dt \to \min, \\
\dot{x}(t) = f(x(t), u(t), t), x(t_0) = x_0, \\
u(t) \in U, t \in [t_0, t_f].
\end{gathered} \tag{1.2}$$

Задача названа простейшей, поскольку не содержит ограничений на состояния, а только ограничения на управления.

1.2.1 Принцип максимума Понтрягина

Принципом максимума называется основное необходимое условие оптимальности в задачах оптимального управления, связанное с максимизацией гамильтониана:

$$H(x, \psi, u, t) = \psi' f(x, u, t) - f_0(x, u, t) = \sum_{j=1}^{n} \psi_j f_j(x, u, t) - f_0(x, u, t).$$

Здесь $\psi = \psi(t) \in \mathbb{R}^n$ — сопряженная переменная.

Теорема 1.1 Пусть $u^0(\cdot), x^0(\cdot)$ — оптимальное управление и траектория в задаче (1.2), $\psi^0(\cdot)$ — соответствующее решение сопряженной системы

$$\dot{\psi}^0(t) = -\frac{\partial H}{\partial x}(x^0(t), \psi^0(t), u^0(t), t),$$

с начальным условием

$$\psi^0(t_f) = -\frac{\partial \varphi}{\partial x}(x^0(t_f)).$$

Тогда для любого $t \in [t_0, t_f]$, управление $u^0(t)$ удовлетворяет условию:

$$H(x^0(t), \psi^0(t), u^0(t), t) = \max_{v \in U} H(x^0(t), \psi^0(t), v, t), t \in [t_0, t_f].$$

Для того чтобы решить задачу с помощью принципа максимума обычно поступают следующим образом. Функцию $H(x, \psi, u, t)$ рассматривают как функцию r переменных $u = (u_1, ..., u_r)$. Далее проводят поточечную оптимизацию для каждого фиксированного набора (x, ψ, t)

$$u(x, \psi, t) = \arg\max_{v \in U} H(x, \psi, v, t). \tag{1.3}$$

Если исходная задача (1.2) имеет решение, функция (1.3) определена на непустом множестве значений (x, ψ, t).

Пусть u в виде (1.3) найдена, тогда можно рассмотреть следующую систему с граничными условиями:

$$\dot{x} = \frac{\partial H}{\partial \psi}(x, \psi, u(x, \psi, t), t) = f(x, \psi, u(x, \psi, t), t),\ x(t_0) = x_0,$$

$$\dot{\psi} = -\frac{\partial H}{\partial x}(x, \psi, u(x, \psi, t), t),\ \psi(t_f) = -\frac{\partial \varphi(x(t_f))}{\partial x}.$$

Таким образом получена специальная краевая задача, которая называется краевой задачей принципа максимума.

Можно ожидать, что имеются лишь отдельные изолированные пары функций $x(\cdot), \psi(\cdot)$, удовлетворяющие краевой задаче принципа максимума. Подставив одну такую пару в (1.3), получим:

$$u(t) = u(x(t), \psi(t), t), t \in [t_0, t_f], \tag{1.4}$$

которая удовлетворяет принципу максимума и, значит, может претендовать на роль оптимального управления, а функция $x(t) = x(t \mid t_0, x_0, u(\cdot)), t \in$

$[t_0, t_f]$, — на роль оптимальной траектории в задаче.

Отметим, что принцип максимума в задаче (1.2) является лишь необходимым условием оптимальности, поэтому построенное управление не может быть оптимальным. Построенная функция называется экстремалью Понтрягина (1.4).

1.2.2 Динамическое программирование

Рассмотрим задачу (1.2) и предположим, что она имеет решение. Следуя динамическому программированию [1], погрузим задачу (1.2) в семейство задач

$$
\begin{gathered}
J_{\tau,z}(u) = \varphi(x(t_f)) + \int_{\tau}^{t_f} f_0(x(t), u(t), t)dt \to \min, \\
\dot{x}(t) = f(x, u),\ x(\tau) = z, \\
u(t) \in U,\ t \in T = [\tau, t_f],
\end{gathered} \tag{1.5}
$$

зависящих от скаляра $\tau \in T$ и n-вектора z.

Пару (τ, z) назовем позицией в задаче (1.2). Обозначим через

$$B(\tau, z) = \min J_{\tau,z}(u)$$

минимальное значение критерия качества в задаче (1.5) для позиции (τ, z). Если для позиции (τ, z) задача (1.5) не имеет решения, положим $B(\tau, z) = +\infty$. Пусть

$$X_\tau = \{z \in \mathbb{R} : B(\tau, z) < +\infty\}.$$

Функцию

$$B(\tau, z), z \in X_\tau, \tau \in T, \tag{1.6}$$

называют функцией Беллмана.

Уравнение в частных производных, которому удовлетворяет функция (1.6), называют уравнением Беллмана

$$-\frac{\partial B(\tau, z)}{\partial \tau} = \min_{v \in U} \left\{ \frac{\partial B'(\tau, z)}{\partial z} f(z, v) + f_0(z, v) \right\}, z \in X_\tau, \tau \in T.$$

Выделяя из семейства (1.5) задачу с $\tau = t_f$, находим граничное условие для уравнения Беллмана

$$B(t_f, z) = \begin{cases} \varphi(z), z \in X, \\ +\infty, z \notin X. \end{cases}$$

Схема применения динамического программирования состоит в следующем: для задач составляется уравнение Беллмана. По решению уравнения строиться позиционное решение.

Основным преимуществом динамического программирования является тот факт, что получаемое решение имеет вид управления типа обратной связи, тогда как принцип максимума строит только программное решение. Подробно обратные связи обсуждаются в следующем разделе.

1.3 Классический подход к построению оптимальных обратных связей

Классический подход к построению оптимальной обратной связи продемонстрируем на примере следующей задачи:

$$J(u) = \varphi(x(t_f)) \to \min, \tag{1.7}$$

$$\dot{x} = f(x, u, t),\ x(t_0) = x_0 \tag{1.8}$$

$$x(t) \in X \in \mathbb{R}^n \tag{1.9}$$

$$u(t) \in U \in \mathbb{R}^r,\ t \in [t_0, t_f] \tag{1.10}$$

В поставленной задаче нужно минимизировать критерий качества (1.7) на траекториях системы (1.8), которые в каждый момент времени лежат в заданном множестве X, согласно (1.9), с помощью ограниченных управляющих воздействий (1.10).

В (1.7) - (1.10): t_0, t_f — заданы,
$x = x(t) \in \mathbb{R}^n$ — состояния системы управления в момент t,
$u = u(t) \in \mathbb{R}^r$ — значения управляющего воздействия в t.
$f : \mathbb{R}^n \times \mathbb{R}^r \times \mathbb{R}^n$ обеспечивает существование и продолжимость решения уравнения (1.8) на промежутке времени $T = [t_0; t_f]$.

Задачу (1.7) – (1.10) будем рассматривать в классе кусочно-непрерывных управляющих воздействий u.

Допустимое программное управление (программа) — кусочно-непрерывная функция $u(t) \in U$, $t \in [t_0, t_f]$, если она порождает траекторию (t), $t \in [t_0, t_f]$, системы (1.8), удовлетворяющей (1.9).

Допустимое программное управление — оптимальное (оптимальная программа), если на нем критерий качества (1.7) достигает оптимального значения: $J(u^0) = \min J(u)$, где минимум берется по всем программам.

Для введения понятия классической оптимальной обратной связи погрузим задачу (1.7) – (1.10) в следующее семейство задач:

$$\varphi(x(t_f)) \to \min, \tag{1.11}$$

$$\dot{x} = f(x, u, t),\ x(\tau) = z,$$
$$x(t) \in X,\ \ u(t) \in U,\ t \in [\tau, t_f].$$

Пусть $u^0(t|\tau, z),\ t \in T_\tau,$ — оптимальная программа задачи (1.11) для позиции (τ, z); X_τ — множество состояний z таких, что для позиции (τ, z) существуют программные решения задачи (1.11).

Функция

$$u^0(\tau, z) = u^0(t|\tau, z),\ z \in X_\tau, \tau \in T, \tag{1.12}$$

— оптимальная обратная связь.

Построение (1.12) — синтез оптимальной системы управления.

Подстановка функции (1.12) в уравнение (1.9) — замыкание системы управления

$$\dot{x} = f(x, u^0(t, x), t),\ x(t_0) = x_0 \tag{1.13}$$

Полученное уравнение — математическая модель оптимальной автоматической системы управления.

Для любого состояния x_0 решение уравнения (1.13) с начальным состоянием $x(t_0) = x_0$ — оптимальная траектория для задачи (1.7) – (1.10).

Отметим, что если оптимальная обратная связь построена в классе кусочно-непрерывных управлений, то во многих задачах уравнение (1.13) представляет собой дифференциальное уравнение с разрывной правой частью и, вообще говоря, не имеет классического решения. В этих случаях используют обобщенное решение.

Чтобы избежать аналитических трудностей, таких как определение решения замкнутой системы управления, в задаче оптимального управления (1.7) – (1.10) перейдем от класса кусочно-непрерывных управлений к дискретным управлениям.

Разобьем $[t_0, t_f]$ на $N \in \mathbb{N}$ частей: $T_h = \{t_0, t_0 + h, ..., t_f - h\}$, где $h = \frac{t_f - t_0}{N}$ — период квантования.

Дискретное управление имеет вид:

$$u(t) = u(s),\ t \in [s, s + h],\ s \in T_h.$$

Понятие программы и оптимальной программы в классе дискретных управлений аналогично определению в классе кусочно-непрерывных управлений.

Семейство, в которое погружается задача (1.7) – (1.10) будет выглядеть так же как и (1.11). Однако теперь семейство зависит от $z \in X$, $\tau \in T_h$.

Оптимальная *дискретная* обратная связь — функция

$$u^0(\tau, z) = u^0(t|\tau, z),\ z \in X_\tau,\ \tau \in T_h.$$

Еще раз обратим внимание на то, что динамическое программирование строит решение в виде обратной связи. Однако, его реализуемость на практике сдерживается так называемым проклятием размерности. Для задач с $n > 3$, содержащими ограничения на управления, динамическое программирование, как правило, не применимо.

1.4 Оптимальное управление в реальном времени и реализация оптимальной обратной связи

Проанализируем, как используется оптимальная дискретная обратная связь $u^0(\tau, z) = u^0(t|\tau, z),\ \tau \in T_h,\ z \in \mathbb{X}_\tau$, в конкретном процессе управления.

Процесс начинается в момент t_0 с подачи на вход объекта управляющего воздействия $u^*(t) \equiv u^*(t_0) = u^0(t_0, x_0)$, $t \geq t_0$.

В момент времени $\tau = t_0 + h$ становится известным состояние объекта $x^*(t_0 + h)$. Здесь верхний индекс * означает, что состояние $x^*(t_0 + h)$ может отличаться от идеального состояния $x(t_0 + h)$ системы (1.8) в силу немоделируемых возмущений. Временем отыскания значения $u^0(t_0 + h, x^*(t_0 + h))$ пренебрежем. С момента $t_0 + h$ на вход объекта подается управляющее воздействие $u^*(t) \equiv u^*(t_0 + h) = u^0(t_0 + h, x^*(t_0 + h))$, $t \geq t_0 + h$.

Продолжая этот процесс, в момент времени $\tau \in T_h$ измеряется состояние объекта $x^*(\tau)$, и с момента τ на вход объекта подается управляющее воздействие $u^*(t) \equiv u^*(\tau) = u^0(\tau, x^*(\tau))$, $t \geq \tau$.

Получается последовательность состояний объекта

$$x^*(t_0) = x_0,\ x^*(t_0 + h),\ \ldots,\ x^*(\tau),\ \ldots,\ x(t_f),$$

и соответствующая ей последовательность управляющих воздействий $u^*(t_0)$,

$u^*(t_0+h)$, ..., $u^*(\tau)$, ..., $u(t_f-h)$. Они удовлетворяют тождеству

$$x^*(t) \equiv f(x^*(t), u^*(t), t) + w^*(t),\ t \in T,\ x^*(t_0) = x_0$$

где

$$u^*(t), t \in T:\ u^*(t) = u^0(\tau, x^*(\tau)),\ t \in [\tau, \tau+h], \tau \in T_h, \tag{1.14}$$

— реализация оптимальной обратной связи (1.7) в конкретном процессе управления.

Отсюда видно, что в конкретном процессе управления оптимальная обратная связь (1.7) не используется целиком на всей области ее определения. Нужны лишь ее значения (1.14) вдоль одной последовательности измеряемых состояний физического объекта $x^*(\tau), \tau \in T_h$, и достаточно уметь для каждой текущей позиции $(\tau, x^*(\tau)), \tau \in T_h$, вычислять значение реализации $u^*(t) = u^0(\tau, x^*(\tau)),\ t \in [\tau, \tau+h]$, оптимальной обратной связи за время $s(\tau) < h$.

Будем говорить, что процесс управления осуществляется в реальном времени, если для каждого текущего момента $\tau \in T_h$ значение $u^*(\tau)$ вычисляется за время $s(\tau) < h$, т.е. до получения следующего измерения $x^*(\tau+h)$. Устройство, способное реализовать оптимальную обратную связь в реальном времени будем называть оптимальным регулятором.

Таким образом, с использованием принципа управления в реальном времени задача синтеза оптимального управления типа обратной связи сводится к построению алгоритма работы оптимального регулятора.

1.5 Теория управления по прогнозирующей модели

Model Predictive Control (MPC) — управление по прогнозирующей модели — подход к управлению линейными и нелинейными динамическими системами, в основе которого лежит решение в реальном времени последовательности так называемых *прогнозирующих* задач оптимального управления. Прогнозирующие задачи, в отличие от исходной, формулируются на конечном горизонте управления (для эффективного численного решения), учитывают текущие состояния исследуемого объекта управления и ограничения на траектории и управляющие воздействия, оценивают будущее поведение системы на основе целей управления на бесконечном горизонте, а также аппроксимируют эти цели [12].

В основе MPC лежат следующие принципы:

- Для предсказания и оптимизации будущего поведения системы используется математическая модель управляемого процесса в пространстве состояний.

- Для выбранной математической модели формулируется прогнозирующая задача оптимального управления, которая решается в каждый конкретный момент времени. В данной задаче:

  1. Конечный промежуток управления.
  2. Начальное состояние математической модели совпадает с измеренным текущим состоянием физического объекта управления.
  3. Критерий качества отражает цели управления: если целью является стабилизация объекта управления, то критерием качества выступает отклонение траектории объекта от положения равновесия.
  4. Учтены ограничения на траекторию и управляющие воздействия;

- Оптимальное управление прогнозирующей задачи (предсказанное управляющее воздействие) применяется к объекту в текущий момент времени и до тех пор пока не будет измерено следующее состояние объекта, затем оптимизация повторяется.

Поскольку в каждый момент времени в задаче оптимального управления учитывается текущее состояние, результирующее управление представляет собой обратную связь.

Популярность MPC в теоретических исследованиях и на практике обусловлена следующими свойствами, которыми не обладают другие методы теории управления:

- Критерий качества в прогнозирующей задаче позволяет учитывать экономические требования к процессу управления (например, минимизацию энергетических затрат);

- Учитываются жесткие ограничения на фазовые и управляющие переменные;

- Метод применим к нелинейным и многосвязным системам.

Задачи стабилизации и регулирования считаются основными и исторически первыми приложениями MPC. Рассмотрим результаты, полученные в теории MPC для задачи стабилизации [12].

Как было отмечено выше, основная идея MPC состоит в том, чтобы использовать математическую модель процесса в пространстве состояний, чтобы предсказывать и оптимизировать поведения динамической системы в дальнейшем. Будем считать, что модель, которая используется для предсказаний точно описывает процесс управления: на объект не действует возмущения и нет неучтенных различий между моделью и физическим объектом. Такие схемы MPC называются номинальными.

Будем исследовать нелинейную, дискретную, стационарную систему:

$$x(t+1) = f(x(t), u(t)),\ x(0) = x_0. \tag{1.15}$$

Здесь $x(t) \in X \subseteq \mathbb{R}^n$ — состояние системы в момент времени t, $u(t) \in U \subseteq \mathbb{R}^r$ — управляющее воздействие в момент t, $t \in I \geq 0$ — дискретное время. Предполагается, что функция $f : \mathbb{R}^n \times \mathbb{R}^r \to \mathbb{R}^n$ непрерывна.

Начальное состояние системы (1.15) задано:

$$x(0) = x_0 \in X.$$

На состояния и управляющие воздействия u накладываются ограничения следующего вида:

$$(x(t),\ u(t)) \in Z \subseteq X \times U,\ t \in \mathbb{I}_{\geq 0} \tag{1.16}$$

— смешанные ограничения. Будем считать, что множество Z компактно.

Цель (стабилизирующего) MPC — построить обратную связь $u(x)$, при которой замкнутая система

$$x(t+1) = g(x(t)) = f(x(t),\ u(x(t))),\ x(0) = x_0, \tag{1.17}$$

будет устойчива в некотором заданном положении равновесия, при этом переходный процесс не нарушает ограничения (1.16) $\forall t \in \mathbb{I}_{\geq 0}$.

Будем предполагать, что $f(0, 0) = 0$, тогда точка начало координат — положение равновесия для системы (1.17).

Определение 1.4 Тривиальное решение $x(t) = 0$ (1.17) называется устойчивым по Ляпунову, если $\forall \varepsilon > 0,\ \exists \delta(\varepsilon) > 0$, что неравенство $||x(t)|| \leq \epsilon$ выполнено, как только $||x_0|| \leq \delta(\varepsilon)$.

Определение 1.5 Тривиальное решение $x(t) = 0$ (1.17) называется асимптотически устойчивым по Ляпунову, если оно устойчиво и выполняется условие $\lim_{t\to\infty} ||x(t)|| = 0$.

Опишем базовый алгоритм MPC. Идея алгоритма состоит в том, чтобы в каждый момент $t \in \mathbb{I}_{\geq 0}$ оптимизировать дальнейшее поведение системы (1.15) на конечном горизонте $N \geq 2$ и использовать первое значение полученного оптимального управления в качестве значения обратной связи для момента t. Под "оптимизацией будущего поведения" понимается решение прогнозирующей задачи оптимального управления.

Понятно, что далее необходимо различать состояния объекта управления $x(t),\ t \in \mathbb{I}_{\geq 0}$, которые измеряются в каждом конкретном процессе управления, и состояния математической модели, которая используется для предсказаний и формулировки прогнозирующей задачи оптимального управления. Поэтому состояния математической модели будем будем обозначать $x(k|\ t),\ k \in \mathbb{I}_{[0,N-1]}$. Они изменяются согласно уравнению

$$x(k+1|\ t) = f(x(k|\ t),\ u(k|\ t)),\ x(0|\ t) = x(t),\ k \in \mathbb{I}_{[0,N-1]}. \tag{1.18}$$

Здесь аргумент t после черты подчеркивает зависимость от текущего момента, для которого проводится оптимизация. Начальное состояние — текущее состояние объекта управления $x(t)$.

Ограничения (1.16), записанные для состояний математической модели (1.18), имеют вид:

$$(x(k|\ t),\ u(k|\ t)) \in Z,\ k \in \mathbb{I}_{[0,N-1]}.$$

Помимо приведенных смешанных ограничений, в задачу, чаще всего, добавляются ограничения в терминальный момент времени. "Терминальные элементы" прогнозирующей задачи более подробно будут рассмотрены ниже после ее формулировки.

Оставшийся элемент прогнозирующей задачи оптимального управления — критерий качества. В задачах стабилизации критерий качества выбирается исследователем, практиком, и является, скорее, параметром настройки схемы MPC. Например, в задаче стабилизации критерий качества выбирается из соображений штрафа любого состояния $x \in X$, отклоняющегося от состояния равновесия x^*. Также часто штрафуются отклонения управления $u \in U$ от значения u^*. Критерий качества будет состоять из терминальной стоимости $V_f(x(N|\ t))$ и суммарной стоимости переходного процесса, т.е. это будет критерий качества типа Больца. Терминальная стоимость будет рассмотрена ниже, при обсуждении терминальных элементов задачи оптимального управления. Стоимость переходного процесса для дискретных систем задается суммой стоимостей за каждый этап:

$$\sum_{k=0}^{N-1} l(x(k|\ t), u(k|\ t)).$$

В литературе функция $l : \mathbb{R}^n \times \mathbb{R}^r \to \mathbb{R}$ называется стоимостью этапа. Считается, что она непрерывна, а также:

1. $l(x^*,\ u^*) = 0$, т.е. стоимость обращается в нуль в точке равновесия;

2. существует функция α_1 класса $\mathcal{K}_\infty$, что выполняется

$$l(x,\ u) \geq \alpha_1(|x - x^*|),\ \forall (x,\ u) \in Z.$$

Таким образом, прогнозирующая задача оптимального управления для момента времени t имеет вид:

$$\mathcal{P}(t): \qquad V(x(t)) = \min_{u(\cdot\ |\ t)} \sum_{k=0}^{N-1} l(x(k|\ t),\ u(k|\ t)) + V_f(x(N|\ t)), \tag{1.19}$$

при условиях

$$x(k+1|\ t) = f(x(k|\ t),\ u(k|\ t)),\ k \in \mathbb{I}_{[0,N-1]},$$

$$x(0|\ t) = x(t),$$

$$(x(k|\ t),\ u(k|\ t)) \in Z,\ k \in \mathbb{I}_{[0,N-1]},$$

$$x(N|\ t) \in X_f.$$

В задаче (1.19) обсуждавшиеся выше "терминальные элементы":

- терминальная стоимость $V_f(x(N|t))$ в критерии качества;
- терминальное ограничение $x(N|\ t) \in X_f$, где X_f — терминальное множество.

Именно условия на эти элементы обеспечивают устойчивость замкнутой системы несмотря на то, что решается задача с конечным горизонтом.

Далее используем следующие обозначения:

$u^0(\cdot\ |\ t) = \{u^0(0\ |\ t),\ \ldots,\ u^0(N-1\ |\ t)\}$ — оптимальное (программное) управление задачи $\mathcal{P}(t)$;

$x^0(\cdot\ |\ t) = \{x^0(0\ |\ t),\ \ldots,\ x^0(N\ |\ t)\}$ — соответствующая траектория;

X_N — множество всех состояний $x \in X$, для которых существует решение задачи (1.19) с $x(t) = x$.

Базовый алгоритм MPC состоит в следующем:

Для каждого $t \in \mathbb{I} \geq 0$

1. измерить состояние $x(t) \in X$ системы (1.15);

2. решить задачу (1.19) с начальным условием $x(0|t) = x(t)$, получить ее решение $u^0(\cdot|t)$;

3. подать на вход системы (1.15) управляющее воздействие:

$$u_{MPC}(t) := u^0(0|t). \tag{1.20}$$

Таким образом, в каждый момент $t \in \mathbb{I} \geq 0$ на систему подается управляющее воздействие (1.20), которое неявно зависит от текущего состояния $x(t)$.

Соответственно, замкнутая система имеет вид:

$$x(t+1) = f(x(t), u^0(0|t)),\ t \in \mathbb{I} \geq 0.$$

Асимтотическая устойчивость замкнутой системы обеспечивается правильным выбором терминальных элементов. Самый простой способ состоит в том, чтобы принять терминальное ограничение равенство

$$x(N|t) = x^*.$$

Другой популярный на практике подход описан в работе [12].

ГЛАВА 2

ПАРАМЕТРИЧЕСКОЕ ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ

Задачи математического программирования, такие как линейное, выпуклое, нелинейное программирование, помимо переменных оптимизации содержат коэффициенты — параметры — задающие, например, целевые функции или функциональные ограничения. Изучением влияния *малых возмущений* этих параметров на решение задачи (оптимальный план, оптимальное значение целевой функции) посвящен анализ чувствительности. С другой стороны, существует ряд прикладных задач, в которых интерес представляет характеризация решения для полного диапазона значений параметров. Раздел теории математического программирования, занимающийся исследованием таких задач называется *параметрическим программирование*. При этом термин параметрическое программирование относится к задачам, в которых исследуется влияние скалярного параметра, в то время как изучением задач с векторными параметрами занимается многопараметрическое программирование.

Как отмечалось во введении, многопараметрическое программирование — основной инструмент построения оптимальных и стабилизирующих обратных связей в данной дипломной работе. В связи с этим, в настоящей главе приводятся основные определения, теоремы и алгоритмы вычисления решений в задачах многопараметрического программирования.

2.1 Общие положения

Первый метод решения параметрических задач линейного программирования был предложен в работе [6] в 1955 г., а затем развит в работах [4] и [11]. Интерес к многопараметрическому программированию возрос с развитием теории систем и оптимального управления. Это связано с тем, что для динамических систем с дискретным временем задачи оптимального управления с ограничениями на управления и состояния и конечным горизонтом управления могут быть сформулированы как задачи математического программирования. Это будут задачи линейного или выпуклого программирования для линейных систем с линейными или выпуклыми функционалами и ограничениями или задачи нелинейного программирования. Как обсуждалось в главе 1, интерес в таких задачах представляет построение оптимальной обратной связи, как функции позиции — пары, в общем случае, состоящей из момента времени и состояния системы в этот момент. Тогда параметром, от которого зависит критерий качества и ограничения, является позиция динамического процесса.

Используя многопараметрическое программирование, можно охарактеризовать и вычислить решение задачи оптимального управления в явном виде как функцию позиции на некотором ограниченном множестве допустимых позиций.

В теории управления по прогнозирующей модели требуется, чтобы прогнозирующая задача оптимального управления решалась в реальном времени для вычисления текущего управляющего воздействия. Прогнозирующая задача зависит от текущего состояния объекта управления, который берется в качестве начального состояния в прогнозирующей задаче и выступает в качестве параметра. С помощью многопараметрического программирования решение в реальном времени (онлайн) может быть перенесено оффлайн, до начала процесса управления. Тогда в реальном времени все вычисления сводятся к вычислению решения задачи многопараметрического программирования для конкретного значения параметра, что значительно проще решения прогнозирующей задачи оптимального управления.

Многопараметрический анализ использует понятие критической области — подмножества параметров задачи параметрического программирования, в котором локальные условия оптимальности остаются неизменными. В этой главе сначала приведем основные результаты нелинейного многопараметрического программирования из работы [4], в частности, результаты, касающиеся непрерывной зависимости решений от параметров, определения невырожденности задач, понятия критической области. Затем опишем алгоритмы решения многопараметрических задач линейного программирования (mp-LP) и многопараметрических задач квадратичного программирования (mp-QP).

Основная идея алгоритмов, представленных в этой главе, состоит в том, чтобы построить критическую область в окрестности заданного значения параметра, используя необходимые и достаточные условия оптимальности, а затем рекурсивно расширить пространство параметров за пределы этой области. Все алгоритмы чрезвычайно просты в реализации, если доступны программные реализации численных методов решения задач нелинейного программирования.

2.2 Непрерывная зависимость решений от параметров в общих многопараметрических задачах

Рассмотрим задачу нелинейного программирования, зависящую от параметра x, который входит в целевую функцию и в ограничения:

$$\begin{aligned} &J^0(x) = \min_z f(z, x), \\ &g(z, x) \le 0, \end{aligned} \tag{2.1}$$

где $z \in M \subseteq \mathbb{R}^s$ — переменная оптимизации, $x \in X \subseteq \mathbb{R}^n$ — параметр, $f : \mathbb{R}^s \times \mathbb{R}^n \to \mathbb{R}$ — целевая функция, а функция $g : \mathbb{R}^s \times \mathbb{R}^n \to \mathbb{R}^{n_g}$ задает ограничения, $g_i(z, x)$ — i-ая компонента вектор-функции $g(x, z)$.

Малые вариации параметра x в задаче математического программирования (2.1) в зависимости от свойств функций f и g может приводить к малым вариациям оптимального решения как функции параметра x, но может и давать его значительные вариации. Цель последующего изложения — установить эти вариации.

При заданном значении параметра x обозначим:

- $R(x)$ — допустимое множество переменных z (множество планов) задачи (2.1) при фиксированном значении параметра:

$$R(x) = \{z \in M \mid g(z, x) \le 0\};$$

- K^* — множество допустимых значение параметра, т. е.

$$K^* = \{x \in \mathbb{R}^n \mid R(x) \ne \varnothing\};$$

- $J^0(x)$ — оптимальное значение задачи (2.1):

$$J^0(x) = \inf_z \{f(z, x) \mid z \in R(x)\};$$

- $Z^0(x)$ — множество оптимальных решений (оптимальных планов) задачи (2.1):

$$Z^0(x) = \{z \in R(x) \mid f(z, x) = J^0(x)\}.$$

Множество $Z^0(x)$ для краткости будем называть оптимальным множеством. Если $Z^0(x)$ является одноточечным множеством для всех x (задача имеет единственное решение), то $z^0(x) \triangleq Z^0(x)$ будем называть оптимальным планом.

Будем предполагать, что K^* замкнуто и $J^0(x)$ конечно при всех $x \in K^*$.

Для того, чтобы функция $J^0(x)$ и оптимальное множество $Z^0(x)$ обладали хорошими свойствами, важным будет условие непрерывности отображения $R : x \mapsto R(x) \subseteq M$. Дадим ряд определений.

Определение 2.1 Отображение $R(x)$ открыто в точке $\overline{x} \in K^*$, если для последовательности $\{x^k\} \subset K^*,\ x^k \to \overline{x}$ и точки $\overline{z} \in R(\overline{x})$ существуют целое число m и последовательность $\{z^k \in M\}$ такие, что $z^k \in R(x^k)$ при $k \geq m$ и $z^k \to \overline{z}$.

Определение 2.2 Отображение $R(x)$ замкнуто в точке $\overline{x} \in K^*$, если $\{x^k\} \subset K^*,\ x^k \to \overline{x},\ z^k \in R(x^k)$, и $z^k \to \overline{z}$ влечет $\overline{z} \in R(\overline{x})$.

Определение 2.3 Отображение $R(x)$ непрерывно в точке $\overline{x} \in K^*$, если оно открыто и замкнуто в точке $\overline{x}$. R непрерывно в K^*, если R непрерывно для любого $x \in K^*$.

На основе приведенных определений проверка непрерывности отображения представляется неосуществимой. Однако существуют достаточные условия непрерывности отображения R на основе свойств функций, задающих ограничения задачи (2.1).

Теорема 2.1 Пусть выполняются следующие условия: 1) M выпукло; 2) все функции $g_i(z,x)$ непрерывны на $M \times X$ и выпуклы по z для каждого фиксированного $x \in X$; 3) существует такое $\overline{z}$, что $g(\overline{z}, \overline{x}) < 0$. Тогда $R(x)$ — непрерывное отображение.

Доказательство теоремы 2.1 приведено в [3].

Теорема 2.2 Пусть выполняются следующие условия: 1) M выпукло; 2) все функции $g_i(z,x)$ непрерывны на $M \times X$ и выпуклы по z и x. Тогда $R(x)$ — непрерывное отображение.

Следующие две теоремы дают достаточные условия непрерывности оптимального значения значения и оптимального плана в задаче (2.1).

Теорема 2.3 Рассмотрим задачу (2.1). Если $R(x)$ — непрерывное отображение и функция $f(z,x)$ непрерывна, то $J^0(x)$, $x \in K^*$ непрерывна.

Теорема 2.4 Рассмотрим задачу (2.1). Пусть выполняются условия: 1) $R(x)$ — непрерывное отображение; 2) множество $R(x)$ является выпуклым для каждого $x \in K^*$, 2) функция $f(z,x)$ непрерывна и строго квазивыпукла по переменной z для каждого x. Тогда $J^0(x)$ и $z^0(x)$ — непрерывные функции.

Теоремы 2.1 и 2.2 можно объединить с теоремами 2.3 и 2.4, чтобы получить следующие следствия:

Следствие 2.1 Рассмотрим многопараметрическую задачу нелинейного программирования (2.1). Предположим, что

1) M — компактное выпуклое множество в $\mathbb{R}^s$;

2) f и g непрерывны на $M \times \mathbb{R}^n$;

3) каждая компонента g выпукла на $M \times K^*$.

Тогда функция $J^0(x)$, $x \in K^*$ непрерывна.

Следствие 2.2 Рассмотрим многопараметрическую задачу нелинейного программирования (2.1). Предположим, что

1) M — компактное выпуклое множество в $\mathbb{R}^s$;

2) f и g непрерывны на $M \times \mathbb{R}^n$;

3) каждая компонента g выпукла на M для каждого $x \in K^*$.

Тогда $J^0(x)$ непрерывна в точке x, если существует точка $\overline{z}$ такая, что $g(\overline{z}, x) < 0$.

2.3 Критические области в задаче многопараметрического линейного программирования

Рассмотрим многопараметрическую задачу линейного программирования (mp-LP)

$$\begin{aligned} &J^0(x) = \min_z J(z,x) = c'z, \\ &Gz \le W + Sx, \end{aligned} \tag{2.2}$$

где $z \in \mathbb{R}^s$ — переменные оптимизации, $x \in \mathbb{R}^n$ — вектор параметров, $J(z,x) \in \mathbb{R}$ — целевая функция и $G \in \mathbb{R}^{m \times s}$, $c \in \mathbb{R}^s$, $W \in \mathbb{R}^m$, и $S \in \mathbb{R}^{m \times n}$.

Для заданного многогранного множества параметров $K \subset R^n$

$$K \triangleq \{x \in \mathbb{R}^n : Tx \leq Z\}$$

обозначим через $K^* \subseteq K$ — множество параметров $x \in K$, при которых задача (2.2) допустима (имеет решение).

Цель последующего изложения — определить допустимую область параметров $K^* \subseteq K$, выражения для оптимального значения целевой функции и для одного из оптимальных планов $z^0(x) \in Z^0(x)$.

Дадим следующие определения прямой и двойственной невырожденности задачи (2.2). Определения приводятся в терминах активных ограничений на некотором плане $z(x)$. Напомним, что ограничение-неравенство $g_i(z, x)$ активно, если оно обращается в равенство: $g_i(z(x), x) = 0$.

Определение 2.4 Для любого заданного $x \in K^*$ задача (2.2) называется прямо невырожденной, если существует $z^0(x) \in Z^0(x)$ такое, что число активных ограничений на плане $z^0(x)$ больше, чем число переменных s.

Определение 2.5 Для любого данного $x \in K^*$ задача (2.2) называется двойственно невырожденной, если двойственная ей задача прямо невырождена.

Многопараметрический анализ основан на понятии критической области (critical region — CR).

В [4] критическая область определяется как подмножество пространства параметров, на котором некоторый фиксированный базис задачи линейного программирования является оптимальным. Алгоритм, предложенный в [4] для решения mp-LP, генерирует непересекающиеся критические области посредством построения и исследования графа базисов. На графе базисов вершины представляют собой оптимальные базисы данной многопараметрической задачи, и две вершины соединены ребром, если можно перейти от одного базиса к другому за одну итерацию (в этом случае базисы называются соседними).

В [13] определение критических областей связано не с базисами, а с набором активных ограничений (см. также [8, 9]).

Пусть $J \triangleq \{1, ..., m\}$ — множество индексов ограничений. Для любого $A \subseteq J$ обозначим через G_A и S_A — подматрицы матриц G и S, соответственно, содержащие строки с индексами из A, через G_j, S_j и W_j — j-ую строку G, S и W, соответственно.

Определение 2.6 Оптимальным разбиением J, связанным с параметром x, назовем пару $(A(x), NA(x))$, где

$$A(x) \triangleq \{j \in J : G_j z^0(x) - S_j x = W_j \ \forall z^0(x) \in Z^0(x)\},$$

$$NA(x) \triangleq \{j \in J : \exists z^0(x) \in Z^0(x),\ G_j z^0(x) - S_j x < W_j\}.$$

Понятно, что множества $A(x), NA(x)$ не пересекаются и их объединением является множество всех индексов строк J.

Для заданного $x^* \in K^*$ обозначим $(A, NA) \triangleq (A(x^*), NA(x^*))$, и далее определим множества

$$\begin{aligned} CR_A &\triangleq \{x \in K : A(x) = A\}, \\ \overline{CR}_A &\triangleq \{x \in K : A(x) \supseteq A\}. \end{aligned} \tag{2.3}$$

Множество CR_A является критической областью, связанной с некоторым фиксированным множеством активных ограничений A, т. е. это множество всех параметров x, таких что ограничения $i \in A$, активны на оптимальном решении $z^0(x)$ задачи (2.2). Ясно, что $\overline{CR}_A \supseteq CR_A$.

Теорема 2.5 Пусть $(A, NA) \triangleq (A(x^*), NA(x^*))$ для некоторого $x^* \in K$, и пусть $d = \dim(\mathrm{range} G_A \bigcap \mathrm{range} S_A)$. Если $d = 0$, то $CR_A = \{x^0\}$. Если $d > 0$, то

i) CR_A — открытый многогранник размерности d;

ii) $\overline{CR}_A$ — замыкание CR_A;

iii) каждая грань $\overline{CR}_A$ есть множество $CR_{A'}$ для некоторой подматрицы $A' \supseteq A$.

Замечание 2.1 Открытый многогранник — множество вида $\{x : Px < q\}$, его замыкание — $\{x : Px \le q\}$.

Из теоремы 2.5 и определения критических областей (2.3) следует, что множество K^* всегда *разбивается единственным образом*. Отметим, что это не так в подходе [4], где критическая область определяется на основе базиса решения задачи линейного программирования (см. также [7]). Цель дальнейшего изложения — построение всех полноразмерных критических областей, содержащихся в K^*, в соответствии с определением (2.3).

Напомним некоторые свойства функции $J^0(x) : \mathbb{R}^n \to \mathbb{R}$ и множества K^* из монографии [4].

Теорема 2.6 ([4]) *Пусть для фиксированного $x_0 \in K$ существует оптимальное решение $z^0(x_0)$ задачи (2.2). Тогда для всех $x \in K$ задача (2.2) имеет либо имеет решение, либо имеет пустое множество планов.*

Сформулированная теорема утверждает, что в задаче линейного программирования(2.2) либо нет решения (несовместны ограничения), либо решение существует. Третья ситуация — неограниченность целевой функции снизу — невозможна.

Теорема 2.7 ([4], стр. 179) *Пусть $K^* \subseteq K$ — множество всех параметров x, таких что задача линейного программирования (2.2) имеет оптимальное решение $z^0(x)$. Тогда K^* — замкнутый многогранник в $\mathbb{R}^n$.*

Следующая теорема формулирует свойства, которыми обладает решение в задаче многопараметрического линейного программирования.

Теорема 2.8 ([4], стр. 180) *Функция $J^0(x)$, $x \in K^*$, является выпуклой и кусочно-аффинной на K^* (в частности, аффинной в каждой критической области CR_{A_i}).*

Если для каждого $x \in K^*$ оптимальное решение $z^0(x)$ единственно, то функция $z^0(x)$, $x \in K^*$, является непрерывной и кусочно-аффинной. В противном случае (в случае неединственного решения) всегда можно определить непрерывную и кусочно-аффинную функцию решения $z^0(x) \in Z^0(x)$, $x \in K^*$.

В следующем разделе опишем алгоритм для определения множества K^*, его разбиения на полноразмерные критические области CR_{A_i}, построения кусочно-аффинных функций оптимального значения $J^0(x)$, $x \in K^*$, и оптимального решения $z^0(x)$, $x \in K^*$.

2.4 Алгоритм построения многопараметрического решения

Алгоритм состоит из двух основных этапов:

- Определить размерность $n' \leq n$ наименьшего аффинного подпространства $\mathcal{K}$, содержащего K^*. Если $n^{'} < n$, найти уравнения по x, которые определяют $\mathcal{K}$.

- Определить разбиение K^* на критические области CR_{A_i} и найти функции оптимального значения $J^0(x)$, $x \in K^*$, и оптимального решения $z^0(x)$, $x \in K^*$.

Первый этап — предварительный, его цель состоит в том, чтобы уменьшить количество параметров и получить полномерную допустимую область параметров. Это облегчает второй этап, который вычисляет многопараметрическое решение и является основой алгоритма mp-LP.

2.4.1 Определение аффинного подпространства $\mathcal{K}$

Для того, чтобы этап 2 работал с минимальной размерностью вектора параметров, первый этап алгоритма нацелен на поиск подпространства $\mathcal{K} \subseteq \mathbb{R}^n$, содержащего параметры x, при которых задача (2.2) допустима (имеет решение).

Первое простое, но важное наблюдение касается числа линейно независимых столбцов матрицы S (обозначим его r_S). Понятно, что если $r_S < n$, то $n - r_S$ параметров можно исключить. Поэтому далее без потери общности считаем, что столбцы матрицы S линейно независимы.

Существует еще один случай, когда размерность вектора параметров может быть понижена. Таким образом, прежде чем решать многопараметрическую задачу, нужен тест для проверки размерности n' наименьшего аффинного подпространства $\mathcal{K}$, содержащего K^*. Более того, результатом теста будет $n^{'} < n$, нужно получить уравнения, описывающие $\mathcal{K}$ в $\mathbb{R}^n$. Эти уравнения используются для изменения координат, чтобы уменьшить число параметров с n до $n^{'}$ и получить многогранник K^*, который имеет полную размерность в $\mathbb{R}^{n^{'}}$.

Вспомним задачу (2.2) и рассмотрим ее как задачу линейного программирования в пространстве $\mathbb{R}^{s+n}$

$$\begin{aligned} &\min_z J(z, x) = c'z, \\ &Gz - Sx \leq W. \end{aligned} \tag{2.4}$$

Понятно, что ограничения в (2.4) определяют многогранник $\mathcal{P}$ в $\mathbb{R}^{s+n}$. Следующая лемма утверждает, что проекция $\Pi_{\mathbb{R}^n}(\mathcal{P})$ многогранника $\mathcal{P}$ на пространство параметров $\mathbb{R}^n$ есть K^*.

Лемма 2.1 $x^* \in K^* \iff \exists z : Gz - Sx^* \leq W \iff x^* \in \Pi_{\mathbb{R}^n}(\mathcal{P})$.

Как следствие, размерность $n^{'}$ наименьшего аффинного подпространства, содержащего K^*, может быть определена путем вычисления размерности проекции $\Pi_{\mathbb{R}^n}(\mathcal{P})$.

Пусть задано следующее H-представление многогранника: $\mathcal{C} = \{\xi \in \mathbb{R}^s : B\xi \leq \upsilon\}$, т.е. многогранник $\mathcal{C}$, определяется набором полупространств $B_i\xi \leq \upsilon_i$.

Определение 2.7 Действительным неравенством многогранника $\mathcal{C}$ назовем такое неравенство $B_i\xi \leq \upsilon_i$, что $\exists\overline{\xi} \in \mathcal{C} : B_i\overline{\xi} < \upsilon_i$.

Учитывая H-представление многогранника $\mathcal{C}$ следующая простая процедура определяет множество I всех недействительных неравенств $\mathcal{C}$.

Algorithm 1:

Input : многогранник $\mathcal{C}$

Output: Множество I всех недействительных неравенств $\mathcal{C}$

$I \leftarrow \emptyset$
$M \leftarrow \{1, ..., m\}$
while $M \neq \emptyset$ **do**
    $j \leftarrow$ first element of $\mathcal{M}$
    $\mathcal{M} \leftarrow \mathcal{M} \backslash \{j\}$
    решаем следующую задачу линейного программирования:
    $B_i\xi \leq v_i, \forall i \in \mathcal{M}$
    if $B_i\xi = v_i$ **then**
        $I \leftarrow I \cup \{j\}$
    end
    for $h \in \mathcal{M}$ **do**
        if $B_h x^* < v_h$ **then**
            $\mathcal{M} \leftarrow \mathcal{M} \backslash \{h\}$
        end
    end
end

Следующий алгоритм описывает стандартную процедуру для определения размерности $n^{'} \leq n$ наименьшего аффинного подпространства $\mathcal{K}$, содержащего K^*, и, когда $n^{'} \leq n$, он находит уравнения, определяющие $\mathcal{K}$.

Algorithm 2:

Input : матрицы G, S, W

Output: Размерность $n^{'} \leq n$ наименьшего аффинного подпространства $\mathcal{K}$, которое содержит k^{*}, если $n^{'} \leq n$, такая что $\mathcal{K} = \{x \in \mathbb{R}^n \ Tx = Z\}$

Отбрасываем истинные неравенства из ограничений 12
if *никакого неравнества не осталось* **then**
  $\mathcal{K} \leftarrow \mathbb{R}^n$
else
  Разрешим $\mathcal{P}_a \triangleq \{(z, x) : G_a z - S_a x = W_a\}$ быть полученным аффинным подпространством, полученным путем сбора не истинных неравенств
  $\{u_1, ..., u_{k'}\}$ - базис ядра $G_a^{'}$
  if $k^{'} = 0$ **then**
    $\Pi_{\mathbb{R}^n}(\mathcal{P}_a)$ и K^{*} полноразмерны
    $\mathcal{K} \leftarrow \mathbb{R}^n$
  else
    $\mathcal{K} \leftarrow \{x \mid Tx = Z\}$, где

$$T = -\begin{bmatrix} u_1^{'} \\ \vdots \\ u_{k'}^{'} \end{bmatrix} S_a, \ Z = \begin{bmatrix} u_1^{'} \\ \vdots \\ u_{k'}^{'} \end{bmatrix} W_a$$

  end
end

С учетом приведенных алгоритмов, в дальнейшем, не теряя общности, будем предполагать, что множество K имеет размерность n в $\mathbb{R}^n$.

2.4.2 Определение критических областей

В этом разделе приведем детали алгоритма mp-LP, а именно процедуру построения критических областей CR_{A_i} в заданном многогранном множестве K. Далее будем считать, что все задачи линейного программирования являются прямо и двойственно невырожденными. Описание вырожденных случаев можно найти в работе [13].

В силу невырожденности задач, для каждого значения параметра $x \in K^{*}$ имеется единственное оптимальное решение $z^0(x)$, т.е. $Z^0(x) = \{z^0(x)\}$, $x \in K^{*}$.

В алгоритме используются:

- прямая допустимость для получения H-представления полиэдральных критических областей,
- условия дополняющей нежесткости для вычисления функции оптимального решения $z^0(x)$, $x \in K^*$,
- двойственная задача для получения функции оптимального значения $J^0(x)$, $x \in K^*$.

Двойственная задача для (2.2) имеет вид

$$\begin{aligned} &\max_y (W + Sx)'y, \\ &G'y = c, \\ &y \leq 0. \end{aligned} \tag{2.5}$$

Прямая и двойственная допустимость и условия дополняющей нежесткости для задач (2.2), (2.5) имеют вид

$$\begin{aligned} \text{прямая допустимость: } & Gz \leq W + Sx, \\ \text{двойственная допустимость: } & G'y = c,\ y \leq 0, \\ \text{условия доп. нежесткости: } & (G_j z - W_j - S_j x) y_i = 0,\ j \in J. \end{aligned}$$

Выберем произвольный вектор параметров $x_0 \in \mathcal{K}$ и решим прямую и двойственную задачи (2.2), (2.5) для $x = x_0$. Пусть z_0^0 и y_0^0 — решения прямой и двойственной задач, соответственно. Значение z_0^0 определяет следующее оптимальное разбиение

$$\begin{aligned} A(x_0) &\triangleq \{j \in J : G_j z_0^0 - S_j x_0 - W_j = 0\} \\ NA(x_0) &\triangleq \{j \in J : G_j z_0^0 - S_j x_0 - W_j < 0\} \end{aligned} \tag{2.6}$$

и, следовательно, критическую область $CR_{A(x_0)}$.

По предположению решение y_0^0 единственно, а по определению критической области значение y_0^0 остается оптимальным для всех $x \in CR_{A(x_0)}$. Тогда, согласно теореме двойственности, функция оптимального значения в $CR_{A(x_0)}$ определяется как

$$J^0(x) = (W + Sx)' y_0^0,\ x \in CR_{A(x_0)},$$

и является аффинной функцией x на $CR_{A(x_0)}$, как указано в теореме 2.8.

Более того, для оптимального разбиения (2.6) условие прямой допустимости можно переписать в виде:

$$G_A z^0(x) = W_A + S_A x, \tag{2.7}$$

$$G_{NA} z^0(x) < W_N A + S_{NA} x. \tag{2.8}$$

В отсутствие двойственной невырожденности прямое решение единственно, и система (2.7) может быть решена, чтобы получить решение $z^0(x)$. Фактически, уравнения (2.7) образуют систему из l равенств, где в отсутствие прямой невырожденности $l = s$ — число активных ограничений. Из (2.7) следует, что

$$z^0(x) = -G_A^{-1} S_A x + G_A^{-1} W_A = Ex + Q,$$

откуда следует линейность z^0 по x.

Из условия прямой допустимости (2.8) получаем следующее представление критической области $CR_{A(x_0)}$

$$G_{NA}(Ex + Q) < W_{NA} + S_{NA} x.$$

Ее замыкание $\overline{CR}_{A(x_0)}$ имеет вид $G_{NA}(Ex + Q) \leq W_{NA} + S_{NA} x$.

После определения критической области $\overline{CR}_{A(x_0)}$ необходимо исследовать оставшуюся часть пространства $R^{\text{rest}} = K \setminus \overline{CR}_{A(x_0)}$ и создать новые критические области.

Эффективный подход к разбиению остального пространства был предложен в [10] и формально доказан в [5]. Далее сформулируем теорему, которая обосновывает процедуру формирования остальных критических областей.

Теорема 2.9 Пусть $Y \subseteq \mathbb{R}^n$ — многогранник, $R_0 \triangleq \{x \in Y : Ax \leq b\}$ — полиэдральное подмножество Y, где $b \in R^{m \times 1}$, $R_0 \neq \emptyset$. Пусть

$$R_i = \left\{ x \in Y : \begin{array}{l} A^i x > b^i \\ A^j x \leq b^j, \ j < i \end{array} \right\}, \quad i = 1, ..., m,$$

где $b \in \mathbb{R}^{m \times 1}$ и пусть $R^{\text{rest}} \triangleq \bigcup_{i=1}^m R_i$.

Тогда

i) $R^{\text{rest}} \cup R_0 = Y$,

ii) $R_0 \cap R_i = \emptyset$, $R_i \cap R_j = \emptyset$, $\forall i \neq j$,

т.е. $\{R_0, R_1, \ldots, R_m\}$ — разбиение Y.

Доказательство. i) Нужно доказать, что точка $x \in Y$ либо принадлежит R_0, либо принадлежит R_i для некоторого i. Если $x \in R_0$, то результат очевиден. В противном случае, существует такой индекс i, что $A^i x > b^i$. Пусть $i^* = \min_{i \le m}\{i : A^i x > b^i\}$. Тогда $x \in R_{i^*}$, поскольку $A^{i^*}x > b^{i^*}$ и $A^j x \le b^j$, $\forall j < i^*$, по определению i^*.

ii) Пусть $x \in R_0$. Тогда не существует ни одного i такого, что $A^i x > b^i$, из чего следует, что $x \notin R_i$, $\forall i \le m$. Пусть $x \in R_i$ и возьмем $i > j$. Поскольку $x \in R_i$, по определению $R_i(i > j)$ имеем $A^j x \le b^j$, из чего следует, что $x \notin R_j$.

2.5 Multi-Parametric Toolbox

Multi-Parametric Toolbox (или MPT) — это набор инструментов и функций на основе Matlab с открытым исходным кодом для параметрической оптимизации и вычислительной геометрии. Инструментарий предлагает широкий спектр алгоритмов, скомпилированных в удобном и доступном для пользователя формате. Полученные в результате законы оптимального управления могут быть встроены в приложения в виде кода на C или развернуты на целевых платформах с использованием Real Time Workshop.

Существуют наборы инструментов, которые предлагают операции, связанные исключительно с вычислительной геометрией, то есть наборы инструментов GEOMETRY, CGLAB и Ellipsoidal Toolbox. Другие наборы инструментов помимо геометрических предлагают также алгоритмы для вычисления и реализации процедур управления, например, набор инструментов Hybrid, MOBY-DIC, RACT, PnPMPC и RoMulOC.

MPT также является едва ли не единственным пакетом, который сочетает методы вычислительной геометрии с современными методами управления. Многие из этих наборов инструментов, включая MPT, используют пакет оптимизации YALMIP [14], который обеспечивает язык высокого уровня для моделирования и формулирования задач конечномерной математической оптимизации.

Содержание MPT для задач управления можно разделить на четыре модуля:

- Моделирование динамических систем.
- Построение MPC-регуляторов для линейных систем на основе явных схем (на основе многопараметрического программирования).
- Анализ обратной связи, полученной MPC-регулятором.
- Развертывание регуляторов MPC на аппаратном уровне.

Каждая часть представляет собой один этап в разработке и реализации явного MPC. Модуль моделирования позволяет описывать системы с дискретным временем и линейной динамикой. Модуль управления позволяет формулировать и решать задачи оптимального управления для линейных систем с ограничениями. Модуль анализа предоставляет методы для исследования поведения и производительности замкнутого контура. Кроме того, MPT также содержит методы, позволяющие уменьшить сложность явных обратных связей MPC.

Рассмотрим самые важные возможности, которые предоставляет MPT.

Основная идея (см. главу 3) состоит в том, чтобы не решать прогнозирующую задачу оптимального управления $\mathcal{P}(\tau)$ в каждый момент времени для нового текущего состояния, а перенести вычисления оффлайн и выполнить их до начала процесса управления. Такая идея реализуема только для линейных систем с выпуклыми критериями качества на основе евклидовых, равномерных норм и использует теорию многопараметрического программирования. Соответственно, в MPT для создания замкнутого контура реализованы следующие два шага (рис. 2.1):

- Оффлайн: построение параметрического решения задачи оптимального управления.

- Онлайн: вычисление и применение полученной обратной связи.

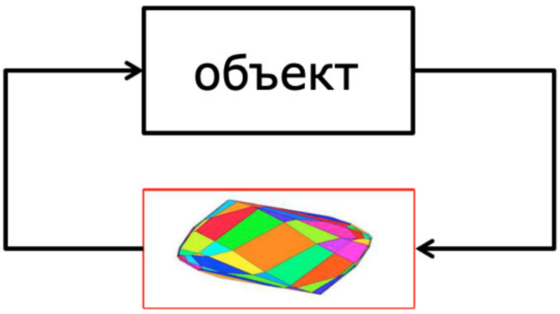


Рис. 2.1: Замкнутый контур управления: реализация в MPT

На рис. 2.1 классический регулятор, замыкающий контур управления использует многопараметрическое программирование, для реализации которого предлагается блок вычислительной геометрии. Основным объектом здесь являются многогранники в их H-представлении $\{x : Ax \leq b, Cx \leq d\}$ или V-представлении $\operatorname{conv}\{v_1, \ldots, v_n\}$. Для этого используются простые функции

$$P = Polyhedron('A',\ A,\ 'b',\ b,\ 'Ae',\ C,\ 'be',\ d),$$
$$Q = Polyhedron('V',\ V,\ 'R',\ R).$$

В частности, в MPT реализованы такие элементы работы с выпуклыми многогранниками, как

- возможность использовать неограниченные множества (рис. 2.2);
- осуществлять объединение выпуклых многогранников (рис. 2.3);
- выполнять следующие операции с множествами: сумма Минковского $S = P + Q$ (рис. 2.4), разность Понтрягина $T = U \backslash S$ (рис. 2.5), проекции, выпуклые оболочки и т.д.

Рис. 2.2: Неограниченные, низкоразмерные множества

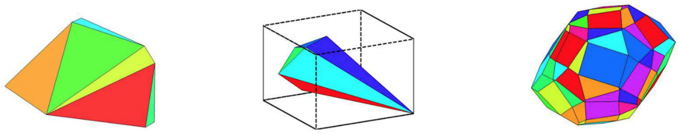


Рис. 2.3: Объединение многогранников

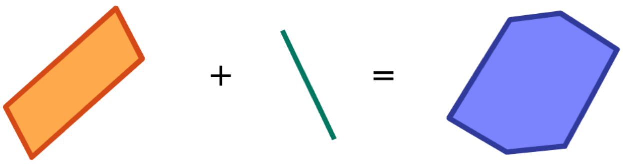


Рис. 2.4: Сумма Минковского

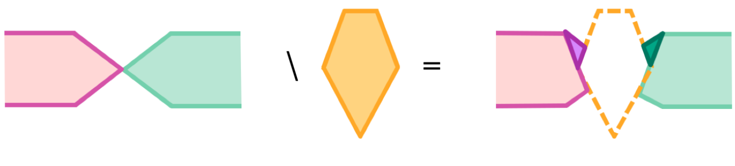


Рис. 2.5: Разность Понтрягина

В модуле моделирования динамической системы основной является функция $LTIsystem$ для задания элементов дискретных линейных стационарных систем управления. В случае системы

$$x_{k+1} = Ax_k + Bu_k, \quad y_k = Cx_k,$$

использование этой функции следующее

$$model = LTISystem('A',\ A,\ 'B',\ B,\ 'C',\ C).$$

Кроме того, модель можно снабдить ограничениями как на управления так и на состояния. Ниже приведены примеры включения в модель простого ограничения вида $|u| \leq 1$ на управления и терминального ограничения в виде многогранника:

$$model.u.min = -1;\ model.u.max = 1;$$

$$model.x.with('terminalSet'),$$

$$model.x.terminalSet = Polyhedron(A, b).$$

Заканчивается постановка задачи определением элементов критерия качества. Поскольку в критерии поддерживаются только суммы стационарных стоимостей этапа вида $||Qx||_2$, $||Qx||_\infty$, $||Qx||_1$, достаточно задать матрицу Q и указать конкретную используемую норму:

$$model.x.penalty = QuadFunction(Q),$$

$$model.x.penalty = OneNormFunction(Q),$$

$$model.x.penalty = InfNormFunction(Q).$$

После определения линейной модели и задания всех элементов прогнозирующей задачи применяются функции из модуля построения MPC-регулятора. Задается конечный горизонт управления N и применяется функция

$$ctrl = MPCController(model, N),$$

после чего для определения значения обратной связи $u(x)$ достаточно применить команду

$$u = ctrl.evaluate(x)$$

За симуляцию MPC-регулятора отвечают следующие функции

$$loop = ClosedLoop(ctrl, model),$$

$$data = loop.simulate(x_0, Nsim).$$

Реализованы также функции графического вывода, позволяющие в случае задач небольшой размерности представить решение — вид целевой функции, кусочно-линейный закон обратной связи и разбиение на критические области на графиках. На рис. 2.6 приведен пример построения обратной связи в задаче стабилизации маятника.

MPT может быть очень полезен в решении параметрических задач. Все это выполняется автономно и не требует большого объемого ресурсов из-за большого количетсва контроллеров, предсьавленых в MPT Toolbox. Ниже приведен пример решения параметрической задачи и показано, каким образом MPT выполняет возложенные на него функции. Результат выполнения программы показан на рисунке 2.7

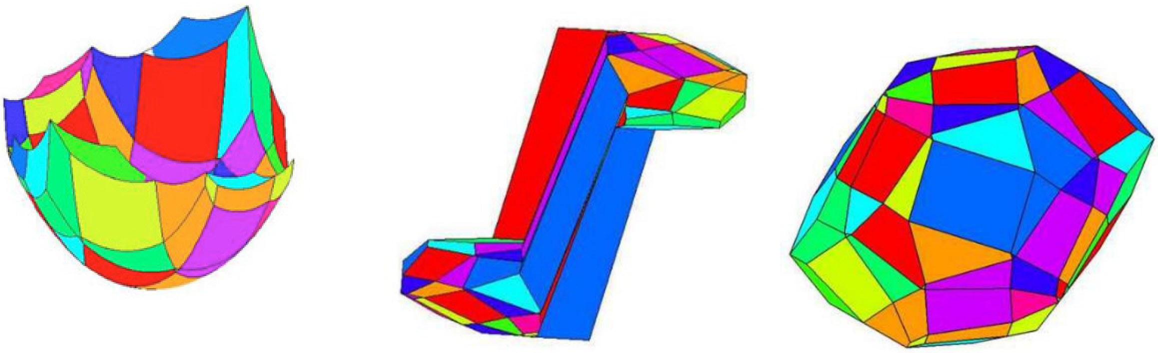


Рис. 2.6: Функция стоимости, обратная связь и критические области

$$\min 1/2u^T Hu + c^T uAu \leq b + E\xi,\ u \geq 0$$

$$\Downarrow$$

$$Hu + c + A^T\lambda - \upsilon = 0Au \leq b + E\xi,\ u \geq 0\lambda^T(Au - b - E\xi) = 0,\ \upsilon^T u = 0$$

$$\Downarrow$$

Найти w, z

$$w - Mz = q + G\xi w \geq 0,\ z \geq 0w^T z = 0$$

$$\Downarrow$$

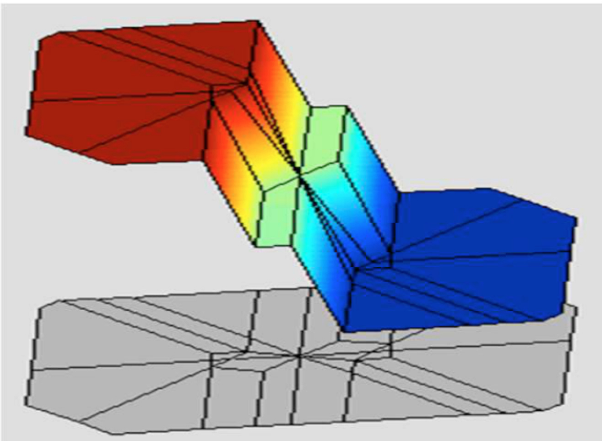


Рис. 2.7: Результат

Целью MPT является предоставление эффективных вычислительных средства для построения регуляторов для задач управления линейными системами в среде программирования Matlab. Соответствующее решение принимает форму кусочно-непрерывной обратной связи. В частности, пространство состояний разбивается на многогранные множества и для каждого из этих множеств закон оптимального управления задается как одна аффинная функция.

Для квадратичных задач, обратная связь также может быть получена для ограниченных линейных системы с применением методов многопараметрического программирования. Несмотря на то, что многопараметрические подходы основаны на автономном вычислении закона обратной связи, вычисление может быстро стать невозможным для больших проблем. Это связано не только высокой сложностью многопараметрических программ, но в основном из-за экспоненциального количества переходов между критическими областями, которые могут произойти, когда регулятор вычисляется в режиме динамического программирования.

Итак, Multi-Parametric Toolbox — это набор алгоритмов для моделирования, управления, анализа и развертывания MPC-регуляторов на основе Matlab. Он обладает мощной геометрической библиотекой, которая расширяет возможности применения набора инструментов за пределы оптимального управления для различных задач, возникающих в вычислительной геометрии.

ГЛАВА 3

ПРИМЕНЕНИЕ ПАРАМЕТРИЧЕСКОГО ПРОГРАММИРОВАНИЯ В ЗАДАЧАХ СИНТЕЗА И MPC

Объектом исследования дипломной работы являются задачи оптимального управления и задачи стабилизации, для которых требуется построить управления типа обратной связи. Инструментом построения обратной связи является многопараметрическое программирование. В настоящей главе покажем, как воспользоваться этим инструментом для построения стабилизирующей обратной связи или оптимальнйо обратной связи в явном виде.

Фактически, мы сформулируем задачи оптимального управления с конечным горизонтом в виде задач математического программирования, в которых управление является вектором оптимизации. Текущее состояние динамической системы входит в функцию стоимости и ограничения в качестве параметра, который влияет на решение задачи математического программирования.

Изучив структуру решения при изменении этого параметра, покажем, что решение задач оптимального управления может быть выражено в виде кусочно-афинной обратной связи. Кроме того, эта обратная связь будет непрерывна, а оптимальное значение задачи — выпукло и непрерывно.

3.1 Явный MPC

Управление по прогнозирующей модели с использованием многопараметрического программирования получило название явного MPC, поскольку обратная связь строится в явном виде. Явный MPC — очень популярный подход, который реализует идею, в которой вычисления переносятся со второго шага базового алгоритма MPC на подготовительный (еще до начала вычислений). Явные схемы на текущий момент времени могут быть использованы только для линейных динамических систем и линейных или квадратичных критериев качества.

В основе явных схем лежат следующие наблюдения: линейные или линейно-квадратичные задачи оптимального управления при численном решении сводятся к задачам линейного или квадратичного программирования, которые зависят от состояния $x = x(t)$, которое, в свою очередь, принимается как параметр. В итоге получаем задачу параметрического программирования, для которой к настоящему моменту были развиты специальные методы решения. Они позволяют нам получить явный закон управления $u(x)$.

Развитие подхода на нелинейные системы является не тривиальным, однако и в линейных случаях есть проблемы, касающиеся эффективности вычислений даже для задач средней размерности.

Ниже продемонстрируем основную идею явного MPC.

Рассматривается задача стабилизации линейной дискретной стационарной системы, для которой формулируется прогнозирующая задачи оптимального управления следующего вида:

$$J(x) = \min_{u(\cdot|t)} \sum_{t}^{t+N-1} L(x(k(t), u(k(t))) + F(x(t+N)t)), \tag{3.1}$$

при условиях

$$x(k+1|\ t) = Ax(k \mid t) + Bu(k \mid t),$$
$$x(t \mid t) = x(t),$$
$$C_x x(k \mid t) \leq d_x,\ t \leq k \leq t+N-1,$$
$$C_u u(k \mid t) \leq d_u,\ C_x x(t+N \mid t) = d_f,$$
$$C_f x(t+N|\ t) \leq d_f,$$

со следующей квадратичной функцией стоимости конкретного этапа и терминальной стоимостью:

$$L(x,u) = x^T Q x + u^T R u,$$
$$F(x) = x^T Q x,\ Q, R > 0,$$

где $C_x, C_u, C_f, d_x, d_u, d_f$ задают ограничения на траекторию, управления и терминальное состояние.

Перепишем (3.1) в виде задачи квадратичного программирования:

$$X := [x^T(t+1|\ t),\ \ldots,\ x^T(t+N|\ t)]^T,$$

$$U := [u^T(t|\ t),\ \ldots,\ u^T(t+N-1|\ t)]^T,$$
$$J(x(t),\ U) = x^T(t)Qx(t) + X^T\tilde{Q}X + U^T\tilde{R}U, \tag{3.2}$$
где $\tilde{Q} = diag(Q, \ldots, Q, P) \in \mathbb{R}^{n(N+1)\times n(N+1)}$, $\tilde{R} = diag(R, \ldots, R) \in \mathbb{R}^{mN\times mN}$.

Далее выражаем состояние системы в момент $t+k$ через $x(t)$ и u:

$$x(t+k \mid t) = A^k x(t) + \sum_{j=0}^{k-1} A^j B u(t+k-j-1 \mid t),\ k = 1, \ldots, N,$$

что можно также представить в виде

$$X = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix} x(t) + \begin{bmatrix} B & 0 & \vdots & 0 \\ AB & B & \vdots & 0 \\ \cdots & \cdots & \vdots & \cdots \\ A^{N-1}B & A^{N-1}B & \vdots & B \end{bmatrix} U. \tag{3.3}$$

Вводя обозначения

$$\Omega = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix},\ \Gamma = \begin{bmatrix} B & 0 & \vdots & 0 \\ AB & B & \vdots & 0 \\ \cdots & \cdots & \vdots & \cdots \\ A^{N-1}B & A^{N-1}B & \vdots & B \end{bmatrix},$$

можем записать

$$X = \Omega x(t) + \Gamma U.$$

С помощью (3.3) переписываем (3.2):

$$J(x(t), u) = \frac{1}{2} x^T(t) Y x(t) + \frac{1}{2} U^T H U + x^T(t) F U_x,$$

где

$$Y = 2(Q + \Omega^T \tilde{Q} \Omega)$$

$$H = 2(\tilde{R} + \Gamma^T \tilde{Q} \Gamma)$$

$$F = 2\Omega^T \tilde{Q} \Gamma$$

$$\begin{bmatrix} C_x & & \\ & \ddots & \\ & & C_x \end{bmatrix} (\Omega x(t) + \Gamma U) \leq \begin{bmatrix} d_x \\ \vdots \\ d_x \end{bmatrix}$$

Аналогично для u и $x(t+N)$

$$GU \leq W + Ex(t),$$

где $G = diag(C_x, \ldots, C_x),\ W = [d_x, \ldots, d_x]^T,\ E = diag(C_x, \ldots, C_x)\Omega.$

Задача (3.1) будет иметь следующий вид:

$$\min_U \frac{1}{2} u^T HU + x^T FU = \frac{1}{2} x^T Y x, \tag{3.4}$$

при условии

$$GU \leq W + Ex.$$

Сделаем замену переменных:

$$z := U + H^{-1} F^T x,$$

где H^{-1} — положительно определенная матрица.

Тогда задача (3.4) примет вид

$$J(x) = \min_z \frac{1}{2} z^T Hz + \frac{1}{2} x^T \tilde{Y} x, \tag{3.5}$$

при условиях

$$Gz \leq W + sx,$$

где $\tilde{Y} = Y - FH^{-1}F^T,\ S := E + GH^{-1}F^T$.

Задача (3.5) строго выпуклая с допустимым множеством с непустой внутренней частью, поэтому для нее выполняются условия регулярности Слейтера. Существует единственное оптимальное решение и оно характеризуется условиями Каруша-Куна-Такера.

Явный MPC использует методы параметрического программирования для задачи (3.5) с параметром x.

Известно, что множество допустимых значений параметра X (выбирается простое многогранное множество) в задаче (3.5) разбивается на непересекающиеся многогранники $X_i,\ i \in I$ (I — множество индексов), в каждом из которых определено решение $z(x)$, как линейная функция x. Последнее, в свою очередь влечет линейность закона управления $u(x) = K_i x,\ x \in X_i,\ i \in I$.

Таким образом задача (3.5) решается для всех состояний (значений параметра x), что дает явную функцию управления, как функцию состояния, т.е. обратную связь $u(x),\ x \in X$.

Алгоритм:

1. Выбрать $x_0 \in X$ (например $x_0 = 0$)

2. Решить задачу при $x = x_0 \Rightarrow z^*(x_0)$

3. Определить активные ограничения $\Rightarrow G^A, W^A, s^A$

4. Вычислить критическую область X_i по активным ограничениям и вычислить функцию управления для этой области

5. Перейти через границу CR^A, взять новое x_0 и $\to 2$

Теорема 3.1 Пусть имеется линейная система, линейные ограничения, функция стоимости квадратичная. Тогда полученный регулятор $u_{MPC}(x)$ является непрерывным и кусочно-линейным. Функция, предствляющая собой оптимальную стоимость $J^*(x)$ также является непрерывной выпуклой и кусочно-квадратичной.

3.2 Синтез оптимальной системы

В данном разделе рассмотрим задачу оптимального управления линейной системой со скалярным управлением

$$\dot{x} = Ax + bu,$$

со следующим критерием качества

$$J(u) = \int_0^{t_f} |u(t)|dt \to \min,$$

при условиях

$$x(0) = x_0,\ x(t_f) = 0, \quad |u(t)| \le L,$$

где $L > 0$ — амплитуда управления.

Семейство, в которое погружается задача для определения обратной связи, имеет вид

$$J(u) = \int_\tau^{t_f} |u(t)|dt \to \min, \tag{3.6}$$

$$\dot{x} = Ax + bu,\ x(\tau) = z,$$

$$x(t_f) = 0, \quad |u(t)| \le L, t \in [\tau, t_f].$$

Здесь z выступает в качестве параметра.

Чтобы избавиться от модуля в подынтегральном выражении представим $u(t)$ в виде:

$$u(t) = u_1(t) - u_2(t),$$

тогда $|u(t)| = u_1(t) + u_2(t)$, где $0 \le u_1(t) \le 1,\ 0 \le u_2(t) \le 1$.

Таким образом, в классе дискретных оптимальных управлений получаем криерий качества:

$$\int_{\tau}^{t_f} u_1(t) + u_2(t)dt = \sum_{s \in T_h(\tau)} h(u_1(s) + u_2(s)) \to \min .$$

Тогда в результате проделанных преобразований привели задачу (3.6) к следующему виду:

$$\begin{aligned} &\sum_{s \in T_h(\tau)} h(u_1(s) + u_2(s)) \to \min, \\ &\dot{x} = Ax + bu_1 - bu_2, \\ &x(\tau) = z, \quad x(t_f) = 0, \\ &0 \leq u_1(s) \leq L, \ 0 \leq u_1(s) \leq L, \ s \in T_h(\tau). \end{aligned}$$

Далее, задачу в классе дискретных управляющих воздействий можно свести к дискретной задаче оптимального управления. Для этого по формуле Коши запишем соотношение:

$$x(s+h) = F(s+h,s)x(s) + \int_{s}^{s+h} F(s+h,\theta)b(\theta)(u_1(\theta) - u_2(\theta))d\theta =$$

$$F(s+h,s)x(s) + \int_{s}^{s+h} F(s+h,\theta)b(\theta)d\theta(u_1(s) - u_2(s)).$$

Введем обозначения:

$$A_h(s) = F(s+h,s),$$

$$B_h(s) = \int_{s}^{s+h} F(s+h,\theta)b(\theta)d\theta.$$

В результате $x(s+h)$ будет представлен в следующем виде:

$$x(s+h) = A_h(s)x(s) + B_h(s)(u_1(s) - u_2(s)), \ s = \overline{k, tf-h}.$$

Таким образом, получаем, что исходная задача (3.6) эквивалентна следующей задаче оптимального управления дискретной системой:

$$\begin{aligned}
&\sum_{s\in T_h(\tau)} h(u_1(s)+u_2(s)) \to \min,\\
&x(s+h) = A_h(s)x(s) + B_h(s)(u_1(s)-u_2(s)),\ s=\overline{k,tf-h},\\
&x(\tau)=z,\ x(t_f)=0,\\
&0\le u_1(s)\le 1,\ 0\le u_2(s)\le 1,\ s=\overline{k,tf-h}.
\end{aligned} \tag{3.7}$$

Примем $\tau = kh$ и распишем получившиеся матрицы в блочном виде:

$$C = \begin{bmatrix} 0 \\ \cdots \\ 0 \\ -h \\ \cdots \\ -h \end{bmatrix} d_* = \begin{bmatrix} -\infty \\ \cdots \\ -\infty \\ 0 \\ \cdots \\ 0 \end{bmatrix} d^* = \begin{bmatrix} +\infty \\ \cdots \\ +\infty \\ 1 \\ \cdots \\ 1 \end{bmatrix},$$

где размерности векторов x — $(N+1-k)n$ и u — $2r(N-k)$.

Матрицы b и A имеют следующий вид:

$$b = \begin{bmatrix} z \\ 0 \\ \cdots \\ 0 \end{bmatrix},$$

$$A = \begin{bmatrix}
E_{n\times n} & 0_{n\times n} & \cdots & 0_{n\times n} & 0_{n\times r}\cdots 0_{n\times r} & 0_{n\times r}\cdots 0_{n\times r} \\
0_{n\times n} & 0_{n\times n} & \cdots & E_{n\times n} & 0_{n\times r}\cdots 0_{n\times r} & 0_{n\times r}\cdots 0_{n\times r} \\
-A_h(\tau) & E_{n\times n} & \cdots & 0_{n\times n} & -B_h(\tau)\cdots 0_{n\times r} & B_h(\tau)\cdots 0_{n\times r} \\
\cdots & & & & & \\
0_{n\times n} & 0_{n\times n} & ..-A_h(t_f-h) & E_{n\times n} & ..-B_h(t_f-h) & ..B_h(t_f-h)
\end{bmatrix},$$

где $n(2+N-k)$ — количество строк, $(N+1-k)n+2r(N-k)$ — количество столбцов.

Задача (3.7) является также и задачей линейного программирования, переменными оптимизации являются x и u, параметром — $z\in\mathbb{R}^n$.

Число переменных оптимизации в задаче линейного программирования можно сократить, выразив траекторию через управление. Получаемая задача носит название функциональной формы.

Запишем по формуле Коши терминальное состояние:

$$x(t_f) = F(t_f, t_0)x_0 + \int_{t_0}^{t_f} F(t_f, \theta)b(u_1(\theta) - u_2(\theta))d\theta.$$

В силу дискретности управляющего воздействия, интеграл на промежутке $T = [t_0, t_f]$ представим в виде суммы интегралов на промежутках $[s, s{+}h]$, $s \in T_h$:

$$x(t_f) = F(t_f, t_0)x_0 + \sum_{s=t_0}^{t_f-h} \int_{s}^{s+h} F(t_f, \theta)b(u_1(\theta) - u_2(\theta))d\theta$$
$$= F(t_f, t_0)x_0 + \sum_{s\in T_h} \int_{s}^{s+h} F(t_f, \theta)bd\theta(u_1(s) - u_2(s)).$$

Теперь представим терминальные ограничения в виде:

$$x(t_f) = F(t_f, t_0)x_0 + \sum_{s\in T_h} \int_{s}^{s+h} F(t_f, \theta)b\; d\theta(u_1(s) - u_2(s)) = 0.$$

Введем обозначение:

$$d_h(s) = \int_{s}^{s+h} F(t_f, \theta)b\; d\theta,$$

$$G(t_0) = -F(t_f, t_0), \quad G(\tau) = -F(t_f, \tau).$$

Исходная задача в классе дискретных управляющих воздействий будет иметь вид:

$$\sum_{s\in T_h} h(u_1(s) + u_2(s)) \to \min,$$
$$\sum_{s\in T_h} d_h(s)(u_1(s) - u_2(s)) = G(t_0)x_0,$$
$$0 \le u_i(s) \le L,\; i = 1, 2, \quad s \in T_h,$$

а задача для позиции (τ, z):

$$\sum_{s\in T_h(\tau)} h(u_1(s) + u_2(s)) \to \min,$$
$$\sum_{s\in T_h(\tau)} d_h(s)(u_1(s) - u_2(s)) = G(\tau)z,$$
$$0 \le u_i(s) \le L,\; i = 1, 2,\; s \in T_h(\tau).$$

Полученные задачи — функциональная форма задачи оптимального управления, при этом параметр был вынесен справа явно.

Соответствующая задача, как задача mp-LP:

$$\begin{aligned}
&\mathbf{1}^T U \to \min,\\
&DU = G(\tau)z,\\
&0 \le U \le L\mathbf{1},
\end{aligned}$$

где

$$D = [d_h(\tau)\ \ -d_h(\tau)\ \ \ldots\ \ d_h(t_f-h)\ \ -d_h(t_f-h)],$$
$$U = [u_1(\tau)\ u_2(\tau)\ \ \ldots\ \ u_1(t_f-h)\ u_2(t_f-h)]^T,$$

$\mathbf{1}$ — вектор из единиц соответствующей размерности.

ГЛАВА 4

ЧИСЛЕННЫЕ ЭКСПЕРИМЕНТЫ

В настоящей главе проведем ряд численных экспериментов по построению MPC-регуляторов в задаче стабилизации линейной системы с ограничениями согласно явной схеме с применением MPT и по построению синтеза оптимальных обратных связей в линейной задаче оптимального управления.

4.1 Применение параметрического программирования в MPC

Рассмотрим дискретную линейную систему управления:

$$x(t+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(t), \tag{4.1}$$

с ограничениями $-5 \leq x(t) \leq 5$ и $-1 \leq u(t) \leq 1$, $t = 0, 1, \ldots$.

Для динамической системы (4.1), положение равновесия которой, очевидно, находится в начале координат, рассмотрим задачу стабилизации. Для решения задачи стабилизации применим явную схему MPC, в которой прогнозирующая задача — линейно-квадратичная с горизонтом прогнозирования $N = 7$, отклонение траектории от устойчивой оценивается квадратами евклидовых норм:

$$\sum_{k=0}^{N-1} \left[||x(k|\ t)||^2 + ||u(k|\ t)||^2 \right].$$

Для данного примера задача многопараметрического программирования была решена с использованием панели инструментов MPT.

Матрицы для задачи будут следующие:

```
A = [ 1, 1; 0,  1];
B = [ 1; 0.5];
```

Задаем линейную модель с дискретным временем:

```
sys = ss(A,B,[],[],1);
model = LTISystem(sys);
```

Зададим ограничения на управления и состояния:

```
model.x.min = [-5; -5];
model.x.max = [5; 5];
model.y.min = [-10; -10];
model.y.max = [10; 10];
model.u.min = -1;
model.u.max = 1;
```

параметры критерия качества

```
R = 1;
model.u.penalty = QuadFunction(R);
model.x.penalty = QuadFunction(eye(2));
```

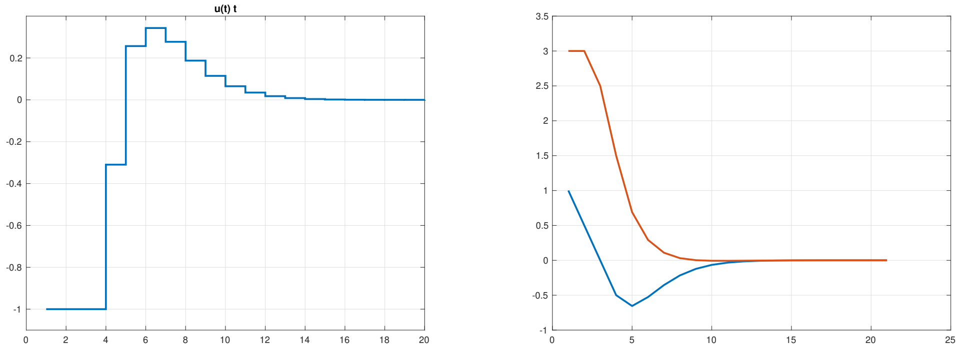
и терминальные значения:

```
P = model.LQRPenalty;
Tset = model.LQRSet;
```

Наконец, формулируем задачу MPC:

```
ctrl = MPCController(model,7);
```

Результаты моделирования замкнутой системы управляющее показаны на рис. 4.1. На рис. 4.1 a) представлена реализация обратной связи (управляющее воздействие, поданное на объект в конкретном процессе) для начального состояния $x_0 = (3, 1)$. На рис. 4.2 изображено разбиение допустимой области на критические области и траектории замкнутой системы для нескольких начальных значений. Из рисунка видно, что задача стабилизации успешно решена.



(a) Управляющее воздействие (b) Траектории X

Рис. 4.1: Результаты реализации для $x = (3, 1)$

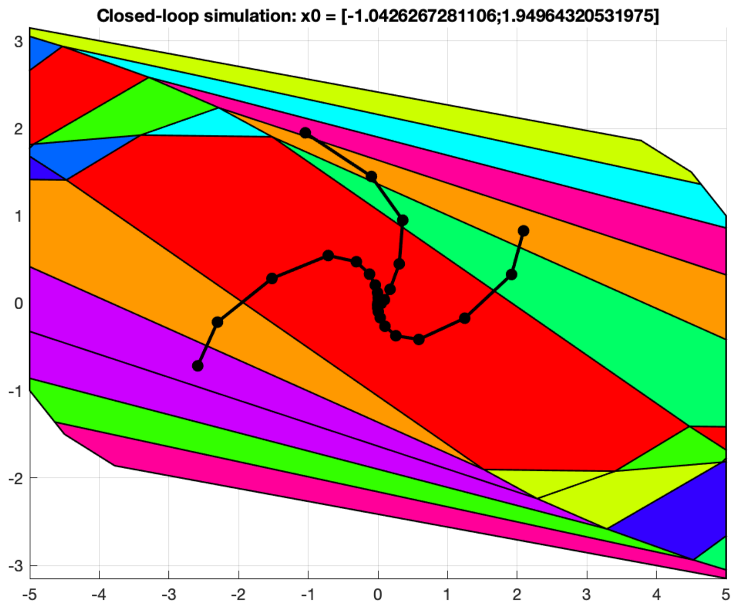


Рис. 4.2: Критические области и траектории замкнутой системы для нескольких начальных значений

4.2 Применение параметрического программирования в задаче синтеза оптимальной линейной системы

В качестве примера синтеза рассмотрим задачу оптимизации колебательной системы второго порядка, критерием качества в которой является минимизация полного импульса управляющего воздействия:

$$J(u) = \int_0^{t_f} |u(t)| dt \to \min,$$

$$\dot{x_1} = x_2, \dot{x_2} = -x_1 + u,$$

с ограничениями $-10 \le x \le 10$ и $-1 \le u(t) \le 1,\ t = 0, 1, \cdots.$

Параметрам присвоим значения: $t_0 = 0, t_f = 10$, число моментов квантования положим равным $N = 20$, тогда период квантования $h = 0.5$.

Для использования MPT, необходимо сформулировать соответствующую задачу для дискретизированной системы. Для этого воспользуемся следующими функциями Matlab

```
sys = ss(A, B, [], []);
sysd = c2d(sys,tf/N,'zoh');
```

В результате будут получены следующие матрицы

$$A_d = \begin{bmatrix} 0.8776 & 0.4794 \\ -0.4794 & 0.8776 \end{bmatrix} B_d = \begin{bmatrix} 0.1224 \\ 0.4794 \end{bmatrix}$$

описывающие дискретную систему $x(t+1) = A_d x(t) + B_d u(t)$, $x(0) = 0$, $t = 0, 1, 2, \ldots$.

В MPT нужно задать все параметры задачи, чтобы передать их в $mptconstructMatrices$, которая представляет собой функцию, на вход которой приходят параметры *probStruc*t и *sysStruct*. Структура системы *sysStruct* будет состоять из матриц A, B, $C = (0, 0)$, $D = 0$, ограничений на x (x_min и x_max) и ограничений на u (u_min и u_max). Структура задачи *probStruct* в свою очередь будет состоять из выбранной нормы, числа N, а также Q, P_N, R и тд. В рассматриваемой задаче вид критерия таков, что $Q = 0$, $P_N = 0$, $R = 1$, норма равномерная.

Чтобы отобразить полную структуру задачи в Matlab поступим так, как показано ниже:

```
sysStruct.A= Ad;
sysStruct.B= Bd;
sysStruct.C= [0 0];
sysStruct.D= 0;

sysStruct.xmin = [-10; -10];
sysStruct.xmax = [10; 10];

sysStruct.umin = -1;
sysStruct.umax = 1;
```

```
nx = size(A, 2);
probStruct.norm=inf;
probStruct.Q=[0 0; 0 0];
probStruct.P_N=[0 0; 0 0];
probStruct.R=1;
probStruct.N=N;
probStruct.subopt_lev=0;
H = [eye(nx); -eye(nx)];
K = eps*ones(nx*2,1);
probStruct.Tconstraint=2;
probStruct.Tset = polytope(H, K);
```

После выполнения такого кода будем иметь все параметры рассматриваемой задачи, сгенерированные в структуру, которую можно решить, используя известные процедуры.

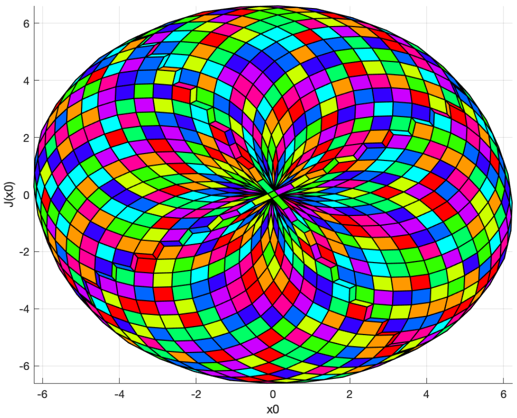
Далее нужно на основании $mptconstructMatrices$ посмотреть класс Opt, который инкапсулирует информацию и решения для задач LP/QP/pLP/pQP/LCP/pLCP. Для конкретного момента времени строится u на основании заданного x_τ и подается на систему. Процедура выполняется в каждый момент из T_h.

Будем реализовывать описанную выше операцию в цикле на каждом шаге и на половине пути добавим возмущение w.

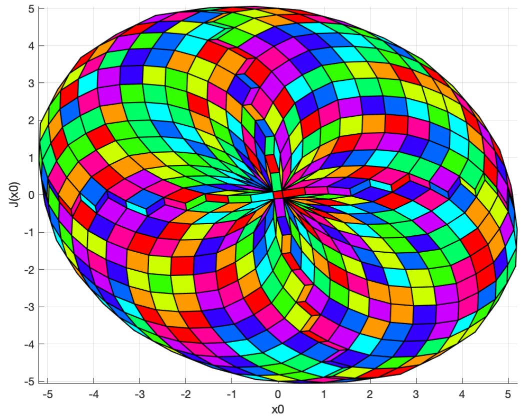
```
for tau = 0:NN-1
    probStruct.N=N;
    Matrices = mpt_constructMatrices(sysStruct,probStruct);
    plp = Opt(Matrices);
    solution = plp.solve();
    u = solution.xopt.feval(xtau, 'primal');
    if N > NN/2
        xtau = Ad * xtau + Bd * u(1) + w;
    else
        xtau = Ad * xtau + Bd * u(1);
    end

    figure;
    solution.xopt.fplot('obj');
    xlabel('x0');
    ylabel('J(x0)');
    N = N-1;
```

```
    X = [X xtau];
    U = [U u(1)];
end
```

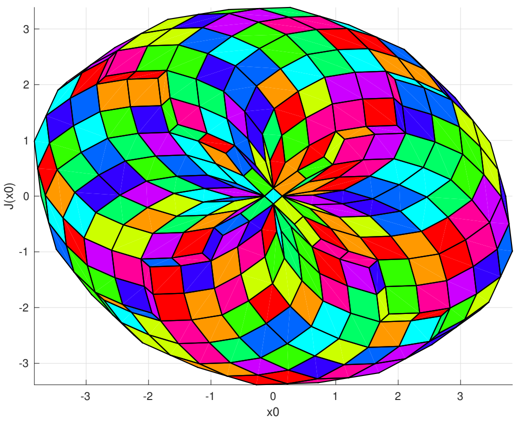
Из рисунка 4.3 видно, что количество критических областей уменьшается с увеличением τ. На рисунке 4.4 a) представлены траектории и управляющее воздействие для $x_0 = (0, 5)$. На рисунке 4.4 b) можем видеть фазовые траектории для $x_0 = (0, 5)$.



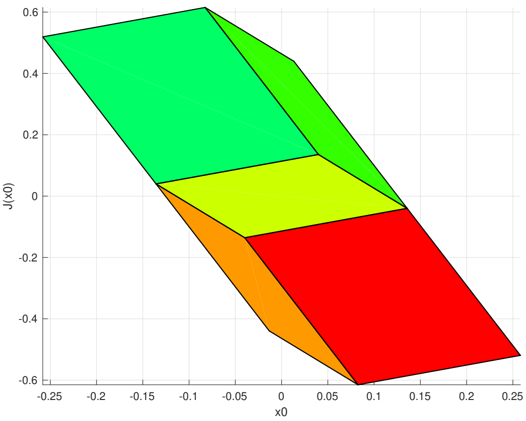
(a) N = 20, $\tau = 0$



(b) N = 15, $\tau = 2.5$

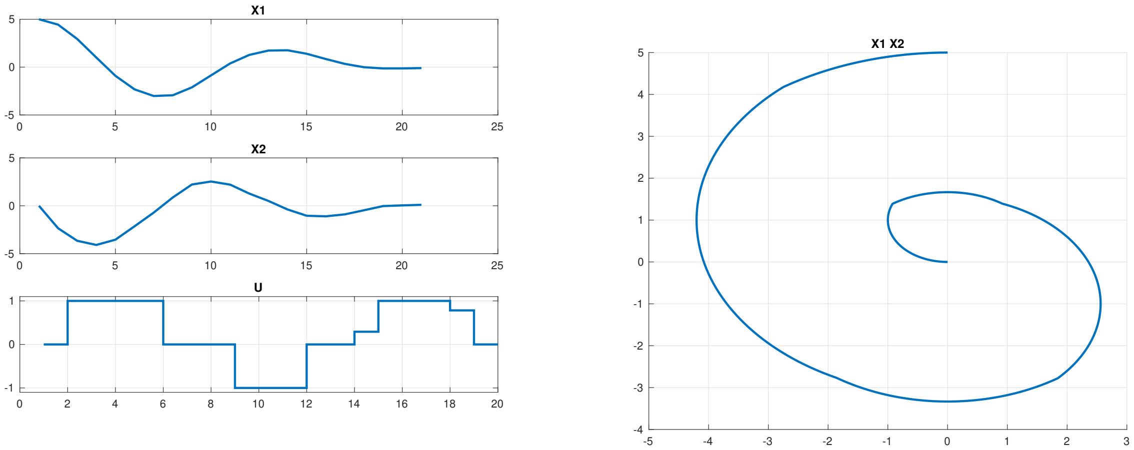


(c) N = 10, $\tau = 5$



(d) N = 1, $\tau = 9.5$

Рис. 4.3: Критические области при различных значениях τ и количеством оставшихся точек равных N



(a) Траектории и управление для $x_0 = (0, 5)$

(b) Фазовые траектории для $x_0 = (0, 5)$

Рис. 4.4: Траектории, управление и фазовые траектории для $x_0 = (0, 5)$

Далее перейдем к решению задачи, которая была рассмотрена выше без использования пакета MPT, затем сравним полученные результаты.

Задача будет решаться стандартной процедурой *linprog* пакета Matlab:

```
u0 =
  linprog([ch', ch'],[],[],[dh,-dh],
    g0,zeros(2*N,1),L*ones(2*N,1)),
```

где c_h в данной задаче будет вектор размерности N, состоящий из h, а g_0 будет вычисляются по формуле:

$$g_0 = g - HF(t_f, t_0)x_0.$$

Зададим это в программе с помощью следующих команд:

```
ch = h*ones(1,N);
g0 = g-H*F(tf-t0)*x0.
```

g_0 будет вектором:

$$g_0 = \begin{bmatrix} -2.7201 \\ 4.1954 \end{bmatrix}.$$

Значения d_h, в свою очередь, будет заполняться с помощью цикла, высчитывая интеграл $\int_t^{t+h} HF(t_f - t)b$ на каждом шаге:

```
for i = 1:N
    dh(:,i) = integral(d,t0+h*(i-1),t0+h*i,'ArrayValued',true);
end
```

где d — подинтегральная функция:

```
d = @(t)H*F(tf-t)*b;
```

. Матрица F задается функцией:

```
F = @(t)expm(A*t);
```

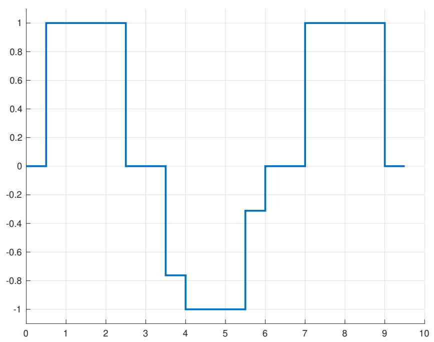
и состоит из следующих значений:

$$F = \begin{bmatrix} 0.8776 & 0.4794 \\ -0.4794 & 0.8776 \end{bmatrix}.$$

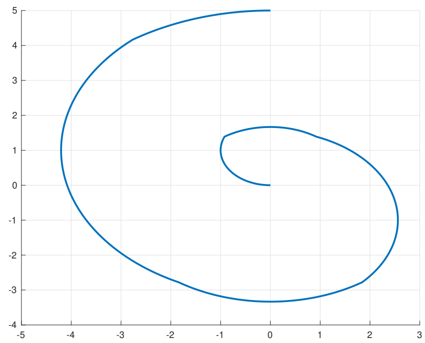
Наконец, x будем считать на каждом шаге, добавляя возмущение $w = 0.05$, когда $i > N/2$:

```
x(:,i+1) = F(h)*x2(:,i) +
    integral(@(t)u2(i)*F(t0+(i+1)*h-t)*b + w,
    t0 + i*h, t0+(i+1)*h,'ArrayValued',true);
```

График оптимальной программы $u^0(t)$ и фазовый портрет оптимальной траектории $x^0(t)$ на плоскости x_1, x_2 показаны на рисунке 4.5



(a) Оптимальная программа



(b) Фазовый портрет оптимальной траектории

Рис. 4.5: Результаты

Таким образом можем заметить, что полученные результаты совпадают с методом, который использует в своей основе возможности МРТ для решения подобного рода задач, описанный выше. Также стоит заметить, что при решении задач в МРТ вся работа проводится до начала процесса, а онлайн требуется только нахождение области. При решении задачи вторым способом, необходимо решать задачу оптимального управления онлайн.

ЗАКЛЮЧЕНИЕ

В дипломной работае рассмотрена задача синтеза оптимальной системы второго порядка. Для исследуемой задачи предложены пути решения, включающие в себя использоваение MPT для Matlab. Проведен анализ методов параметрического программирования. Результаты проиллюстрированы на графиках, представленных выше. Также в работе были изучены основные результаты теории оптимального управления и подход к построению реализаций оптимальных обратных связей.

Также предложены методы синтеза оптимальных обратных связей в линейной задаче оптимального управления на основе идей параметрического программирования, по аналогии с подходами, использующимися в явных схемах управления по прогнозирующей модели.

В работе представлена реализация управления по прогнозирующей модели, явные схемы MPC на основе параметрического линейного программирования и реализованы пути решения на конкретных примерах. В ходе работы произведено сравнение результатов, полученных путем использования MPT для Matlab, с результатами, полученными более стандартным методом, используя процедуру linprog для Matlab.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1 Bellman, R. The Dawn of Dynamic Programming / R. Bellman. – Dover Publications. – 2003.

2 Hazewinkel, M. Pontryagin maximum principle. Encyclopedia of Mathematics / M. Hazewinkel // Springer Science + Business Media B.V. – 2001.

3 Hogan, W. M. Point-to-set maps in mathematical programming / W. M. Hogan // SIAM Review. – 1973. – Vol. 15, №3. – P. 591-603.

4 Gal, T. Multiparametric linear programming / T. Gal, J. Nedoma // Management Science. – 1972. – P.406-442.

5 Bemporad A. The explicit linear quadratic regulator for constrained systems / A. Bemporad, M. Morari, V. Dua, and E.N. Pistikopoulos // Automatica. – 2002. – Vol. 38, № 1. – P. 3–20.

6 Gal. T. Parametric Programming /T. Gal. // de Gruyter. – Berlin, 1995.

7 Berkelaar A.B. The optimal set and optimal partition approach to linear and quadratic programming / A.B. Berkelaar, K. Roos, and T. Terlaky // Advances in Sensitivity Analysis and Parametric Programming / T. Gal and H.J. Greenberg editors. – 1997. – CH. 6.

8 Adler I. I. A geometric view of parametric linear programming / Adler and R.D.C. Monteiro // Algorithmica. – 1992. – Vol. 8, №2. – P. 161–176.

9 Filippi C. On the geometry of optimal partition sets in multiparametric linear programming / C. Filippi. // Department of Pure and Applied Mathematics, University of Padova. – Italy, 1997.

10 Dua V. An algorithm for the solution of multiparametric mixed integer linear programming problems / V. Dua and E.N. Pistikopoulos.

11 Bemporad A. The explicit linear quadratic regulator for constrained systems / A. Bemporad, M. Morari, V. Dua, and E.N. Pistikopoulos // Automatica. – 2002. – Vol. 38, №1. – P 3–20.

12 Chen H. A Quasi-Infinite Horizon Nonlinear Model Predictive Control Scheme with Guaranteed Stability / H. Chen, F. Allgower // Automatica. – 1998.

13 Borelli, F. Constrained Optimal Control of Linear and Hybrid Systems / F. Borelli // Lecture Notes in Control and Information Sciences. - Springer, 2003. - Vol. 290. – P. 293.

14 Löfberg J. Automatic robust convex programming / J. L'ofberg // Optimization methods and software – 2012. – Vol. 27, №1. – P. 115-129.