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

ДЕЦЕНТРАЛИЗОВАННОЕ ОПТИМАЛЬНОЕ УПРАВЛЕНИЕ

Курсовая работа

Лозовский Иван Иванович
студента 4 курса,
специальность «прикладная математика»

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

Минск, 2020

ОГЛАВЛЕНИЕ

ГЛАВА 1

ОСНОВНЫЕ ПОНЯТИЯ И ОБЗОР ЛИТЕРАТУРЫ

В настоящей главе формируются основные понятия, используемые в курсовой работе: приводится классификация задач оптимального управления, разбор их составляющих, определяется объект исследования; даются точные определения программного и позиционного решения; описывается алгоритм работы оптимального регулятора; рассматриваются прямые методы решения задачи оптимального управления, в частности последовательный и параллельный подходы.

1.1 Задачи оптимального управления

Задача оптимального управления формируется из пяти составляющих: временного интервала, математической модели, класса управлений и ограничений на них, ограничений на фазовую траекторию и критерия качества.

1) Временной интервал. По временному интервалу задачи оптимального управления разделяются на непрерывные, рассматриваемые на некотором промежутке времени $T = [t_0,\ t_f]$, и дискретные, где используются дискретные моменты времени $T_h = \{t_0,\ t_0 + h, \ldots, t_f - h\},\ h = \dfrac{t_f - t_0}{N},\ N \in \mathbb{N}$, то есть, например, если $t \in [s,\ s + h[,\ s \in T_h$, то дискретное управление $u(t) = u(s)$. Выделяют задачи с фиксированным и нефиксированном временем окончания динамического процесса, а также задачи на бесконечном интервале.

2) Математическая модель. Динамический процесс обычно моделируется дифференциальными

$$\dot{x}(t) = f(x(t), u(t), t),\ t \in T,$$

или разностными уравнениями

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

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

3) Класс управлений и ограничения на них. Для непрерывного процесса управления четко указывается класс функций, из которого выбираются управления. Кроме класса доступных управлений задается множество $U \subset \mathbb{R}^r$ — множество допустимых значений управления. Как правило U — компакт в $\mathbb{R}^r$.

Определение 1.1 Кусочно-непрерывная (дискретная, измеримая и т.д.) 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

$$X(t_i) \in X_i,\ i = 1 \dots m,$$

на всем промежутке управления — фазовые ограничения

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

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

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

5) Критерий качества. Множество допустимых управлений, как правило, содержит более одного элемента, поэтому возникает необходимость сравнивать их между собой. Для этого вводится функционал $J(u)$, называемый критерием качества, и выбирается операция минимизации или максимизации этого функционала, результат которой определяет наилучшее(оптимальное) управление. В теории оптимального управления различают четыре типа критериев качества: Майера, Больца, Лагранжа, задачи быстродействия. Все 4 критерия качества эквивалентны между собой.

Для примера выпишем критерий качества типа Майера (териминальный критерий):

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

Определение 1.3 Допустимое управление $u^0(\cdot)$ называется оптимальным, если на нем критерий качества достигает экстремального значения.

1.2 Программные и позиционные решения

Объектом исследований в настоящей работе будут непрерывные задачи оптимального управления линейными нестационарными системами с линейным терминальным ограничением и критерием качества:

$$J(u) = c'x(t_f) \to \min, \tag{1.1}$$

$$\dot{x} = A(t)x + B(t)u,\ x(t_0) = x_0,$$

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

$$u(t) \in U,\ t \in T = [t_0, t_f],$$

где $A(t)$ — непрерываная $n \times n$-матричная функция, и $B(t)$ — непрерываная $n \times r$-матричная функция.

Задача (1.1) будет исследоваться в классе дискретных управляющих воздействий

$$u(t) \equiv u(\tau), \quad t \in [\tau,\ \tau + h[, \quad \tau \in T_h = \{t_0,\ t_0 - h, \ldots,\ t_f - h\},$$

где $h = \dfrac{t_f - t_0}{N}$ — период квантавания, $N \in \mathbb{N}$ — заданная мощность множества T_h.

Определение 1.4 Программа $u^0(t),\ t \in T$, называется программным решением задачи (1.1) (оптимальной программой), если на соответствующей ей траектории $x^0(t),\ t \in T$, выполняется равенство

$$c'x^0(t_f) = \min_u c'x(t_f).$$

Приведем задачу (1.1) к набору задач, зависящих от скаляра $\tau \in T_h = \{t_0,\ t_0 - h, \dots,\ t_f\}$ и n-вектора z:

$$J(u) = c'x(t_f) \to \min, \tag{1.2}$$

$$\dot{x} = A(t)x + B(t)u,\ x(\tau) = z,$$

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

$$u(t) \in U,\ t \in T(\tau) = [\tau, t_f],$$

Пусть $u^0(t|\tau, z),\ t \in T(\tau),$ — оптимальная программа задачи (1.2) для позиции (τ, z); X_τ — множество состояний z, для которых в момент τ существуют программные решения.

Определение 1.5 Функция

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

называется позиционным решением задачи (1.2) (оптимальной обратной связью).

Управление называется программным, если оно регулируется программно, строго, без динамического наблюдения за состоянием объекта и контроля воздействия на него, то есть базируясь только на априорных оценках. В случае позиционного управления управляющие воздействия представляют собой функции от позиции объекта, которые содержат всю доступную на текущий момент информацию. Они также не корректируются в процессе управления.

Программное управление редко применяется на практике, так как со временем, из-за изначальной неточности математической модели и построения обратных связей, а также из-за действия в процессе управления неизвестных возмущений, накапливается общая погрешность вычислений.

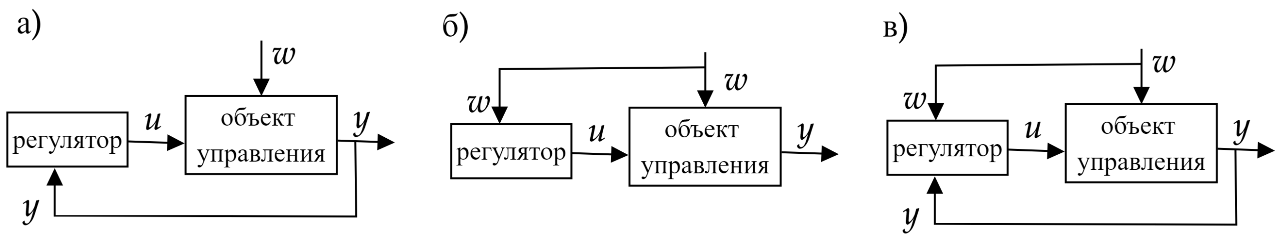

Рис. 1.1: а) обратная связь; б) прямая связь; в) комбинированная связь

Программное и позиционное управления следуют одному из трех принципов управления: по разомкнутому контуру, по замкнутому контуру, в реальном времени. Программные управления исполняются на разомкнутом контуре, а позиционные — на замкнутом и в реальном времени. При создании систем управления по принципу замкнутого контура используются связи 3-х типов: прямые (по входу), обратные (по выходу) и комбинированные (Рис. 1.1). По сути связи — функции, преобразующие наблюдаемые входные и выходные сигналы в управляющие воздействия.

В системах реального времени связи не используются. Нужные для управления их текущие значения вычисляются по ходу каждого процесса управления вычислительными устройствами.

Замкнутые системы управления и системы управления в реальном времени называют автоматическими и автоматизированными, соответственно.

Проблему синтеза оптимальных систем в рамках принципа управления по замкнутому контуру не удается решить из-за проклятия размерности ни с помощью принципа максимума, ни с помощью динамического программирования Беллмана — втрого фундаментального метода теории оптимального управления. Исключение составляет линейно-квадратичная задача Летова-Калмана. В силу этого позиционное решение задачи получается в ви- де простейшей (линейной) обратной связи.

Одним из способов избежания проклятия размерности является переход к синтезу оптимальных систем, следуя современному принципу оптимального управления в реальном времени.

1.3 Управление в реальном времени

Пусть $x^*(\tau)$ — измеренное в конкретном процессе управления состояние объекта управления. Оно отличается от состояния $x(\tau)$ математической модели (1.1) в силу неучтенных в принятой модели возмущений, неточностей математического моделирования, невязки линеаризации нелинейной модели и других факторов.

Введем $\Delta(\tau) \equiv \Delta(\tau,\ x^*(\tau))$ — время отыскания оптимальной обратной связи $u^0(\tau,\ x^*(\tau))$.

Определение 1.6 Функцию $u^*(t),\ t \in T$:

$$u^*(t) \equiv u^0(\tau,\ x^*(\tau)) = u^0(\tau|\tau, x^*(\tau)), \quad t \in [\tau+\Delta(\tau), \tau+h+\Delta(\tau+h)[, \quad \tau \in T_h,$$

назовем реализацией оптимальной обратной связи $u^0(\tau,\ x^*(\tau))$, $z \in X_\tau$, $\tau \in$

T_h, в конкретном процессе управления.

Тогда можно сказать, что в момент $\tau \in T_h$ определяется состояние объекта, а в момент $\tau+\Delta(\tau)$ будет найдено оптимальное для момента τ управление, которое подается на вход объекта управления.

Определение 1.7 Если в каждый момент времени $\tau \in T_h$ вычисление $u^*(\tau)$ производится за время $\Delta(\tau) < h$, то описанная выше схема управления объектом называется управлением в реальном времени.

Определение 1.8 Оптимальным регулятором, реализующим оптимальную обратную связь, называется устройство, способное вычислять $u^*(\tau),\ \tau \in T_h$, за время $\Delta(\tau) < h$.

Далее, в главе 2, будет приведен конкретный алгоритм работы оптимального регулятора.

1.4 Численные методы решения задач оптимального управления

Различают несколько подходов к решению непрерывных задач оптимального управления. Отметим динамическое программирование [2], непрямые методы, основанные на применении принципа максимума [1] и прямые методы решения. В настоящей работе будут применяться последние.

Прямые методы сводят непрерывную динамическую систему к системе с дискретным временем, после чего применяются численные методы нелинейной оптимизации (или линейного, квадратичного программирования).

Рассмотрим прямые методы на примере методов Single Shooting и Multiple Shooting, которые в свою очередь реализуют последовательный и параллельный подходы решения задач оптимального управления, соответственно.

Все Shooting методы содержат в себе модули для решения ОДУ, что позволяет исключить динамическую систему в непрерывном времени. Это осуществляется через замену функции управления $u(t)$ полиномами, кусочно-постоянной функцией или сплайном.

Обозначим конечное множество параметров управления вектором q, а итоговую функцию управления как $u(t;\ q)$.

Наиболее распространенная форма управления — кусочно-постоянные управления, для которых выбирается фиксированная сетка $0 = t_0 < t_1 <$

$... < t_N = t_f$ и N параметров $q_i \in \mathbb{R}^{n_u}$, $i = 0, ..., N-1$. При этом полагается, что $u(t;\ q) \equiv q_i, \quad t \in [t_i,\ t_{i+1}]$. Если сетка равномерная, то рассматриваемое управление — дискретное.

Таким образом размерность вектора $q = (q_0,\ \dots\ , q_{N-1})$ — $N \times n_u$.

В методе Single Shooting, являющемся последовательным подходом, $x(t)$, $[0, t_f]$, находится с помощью численного метода решения ОДУ, в котором начальное условием задается через x_0 и используются значения $u(t;\ q)$. Итоговую траекторию обозначим как $x(t;\ q)$, $t \in [0, t_f]$.

Тогда исходная задача примет вид:

$$c' x(t_N;\ q) \to \min_q,$$

$$x(t_0, q) = x_0,$$

$$x(t_N;\ q) \in X_f,$$

$$q_i \in U,\ i = 0, 1, \dots, N-1.$$

Полученная задача — задача математического программирования. Если X_f, U — многогранники, то это задача линейного программирования.

В методе Multiple Shooting, являющемся параллельным подходом, аналогично Single Shooting методу управление дискретизируется на сетке:

$$u(t;\ q) \equiv q_i, \quad t \in [t_i,\ t_{i+1}].$$

Однако в дальнейшем ОДУ решается отдельно для каждого интервала $[t_i,\ t_{i+1}]$, с заданными на них искусственными начальными значениями состояния s_i:

$$\dot{x}_i(t;\ s_i,\ q_i) = f(x_i(t;\ s_i,\ q_i),\ q_i),$$

$$t \in [t_i,\ t_{i+1}],$$

$$s_0 = x_0,$$

$$s_{i+1} = x_i(t_{i+1};\ s_i,\ q_i).$$

В итоге получм оптимизационную задачу вида:

$$c' s_N \to \min_{q,s},$$

$$s_0 - x_0 = 0,$$

$$x_i(t_{i+1};\ s_i,\ q_i) - s_{i+1} = 0,$$

$$q_i \in U,\ i = 0, 1, \ldots, N-1.$$

Таким образом, определен объект исследования настоящей работы — это линейная задача терминального управления. На примере построенной задачи введены понятия позиционного и программного решения, описан принцип работы оптимального регулятора. На примере данной задачи показано, как реализуются последовательный и параллельный подходы решения непрерывных задач оптимального управления и описаны методы Single Shooting и Multiple Shooting.

ГЛАВА 2

ЦЕНТРАЛИЗОВАННОЕ ОПТИМАЛЬНОЕ УПРАВЛЕНИЕ

В настоящей главе рассматривается задача оптимального управления группой линейных взаимосвязанных систем. Учитываются динамические связи между системами, т.е. состояния соседних систем влияют на динамику каждой отдельной системы, входя в правую часть дифференциального уравнения. Также учитываются статические связи, в частности, общее терминальное ограничение на состояния всех систем в терминальный момент времени. В настоящей работе рассматривается случай централизованного оптимального управления в реальном времени.

2.1 Задача оптимального управления группой динамических систем

Определим T и T_h как в главе 1:

$$T = [t_0,\ t_f],\ T_h = \{t_0,\ t_0 - h, \dots,\ t_f - h\},\ h = \frac{t_f - t_0}{N},\ N \in \mathbb{N}.$$

Пусть $I = \{1,\ 2,\ \dots,\ q\}$, $I_i = I \backslash i$; $A_{ij}(t) \in \mathbb{R}^{n_i \times n_j}$, $B_{ij}(t) \in \mathbb{R}^{n_i \times r_j}$, $t \in T$, $i, j \in I$, — кусочно-непрерывные матричные функции; $A_i(t) = A_{ii}(t)$, $B_i(t) = B_{ii}(t)$, $t \in T$, $i \in I$.

Составим задачу оптимального управления для группы q взаимосвязанных линейных нестационарных объектов управления.

Будем считать [4], что на промежутке T модель i-го ($i \in I$) объекта имеет вид:

$$\dot{x}_i = A_i(t)x_i + \sum_{j \in I_i} A_{ij}(t)x_j + B_i(t)u_i + \sum_{j \in I_i} B_{ij}(t)u_j, \quad x_i(t_0) = x_{i,0}, \tag{2.1}$$

где $x_i = x_i(t) \in \mathbb{R}^{n_i}$ — состояние i-ой математической модели; $u_i = u_i(t) \in U_i = \{u \in \mathbb{R}^{r_i} : u_{i*} \leq u \leq u_i^*\}$ — дискретное управляющие воздействие i-ой математической модели с периодом квантования h ($u_{i*}, u_i^* \in \mathbb{R}^{r_i}$ — заданные векторы). Будем также считать, что $n = \sum_{i \in I_i} n_i$, $r = \sum_{i \in I_i} r_i$. Таким образом $u \in \mathbb{R}^r$, $x \in \mathbb{R}^n$.

Функция $A_i(t), t \in T$, характеризует собственную динамику i-й модели, а $B_i(t), t \in T$, определяет её входное устройство.

Функции $A_{ij}(t), t \in T, i \in I_j$, служат для описания влияния на i-ую модель остальных моделей, а функции $B_{ij}(t), t \in T, i \in I_j$, — для описания влияния на на i-ую модель управляющих воздействий остальных моделей.

Таким образом, объекты в рассматриваемой группе содержат динамические взаимосвязи со всеми остальными объектами.

Кроме динамических взаимосвязей будем исследовать случай статических взаимосвязей, когда связь наложена в какой-то конкретный момент времени. Таким моментом выберем терминальный момент t_f.

Определим терминальное множество S (общее для всех объектов) в виде

$$x(t_f) \in S = \left\{ \begin{pmatrix} x_1 \\ \vdots \\ x_q \end{pmatrix} : \sum_{i \in I} H_i x_i = g \right\}, \tag{2.2}$$

где $H_i \in \mathbb{R}^{m \times n_i}$, $\mathrm{rank}(H_i) = m \leq n_i, i \in I, g \in \mathbb{R}^m$.

В дальнейшем пусть $H = (H_1, ..., H_q)$, т.е (2.2) можно записать в виде

$$x(t_f) \in S = \{x : Hx = g\}, \tag{2.3}$$

Целью управления является минимизация линейного терминального критерия качества:

$$\sum_{i \in I} c_i' x_i(t_f) \to \min_u, \tag{2.4}$$

где $c_i \in \mathbb{R}^{n_i}$ — заданные векторы.

Пусть $c = \begin{pmatrix} c_1 \\ \vdots \\ c_q \end{pmatrix}$, тогда (2.4) можно записать в виде:

$$c' x \to \min_u. \tag{2.5}$$

Таким образом задача оптимального управления для группы q взаимосвязанных линейных нестационарных объектов управления имеет вид:

$$\sum_{i \in I} c_i' x_i(t_f) \to \min_u, \tag{2.6}$$

$$\dot{x}_i = A_i(t)x_i + \sum_{j\in I_i} A_{ij}(t)x_j + B_i(t)u_i + \sum_{j\in I_i} B_{ij}(t)u_j, \quad x_i(t_0) = x_{i,0},\ i \in I,$$

$$\sum_{i\in I} H_i x_i(t_f) = g,$$

$$u_i(t) \in U_i, \quad t \in T, \quad i \in I.$$

Группой динамических объектов (2.1) можно управлять централизованно и децентрализованно [3].

В *централизованном* случае (см. также главу 1) имеется общий центр управления, которой на каждом промежутке времени $[\tau, \tau + h[, \tau \in T_h$ по точной информации о состоянии $x^*(\tau) = (x_i^*(\tau), i \in I)$ группы вырабатывает управляющие воздействия $u^*(t) = (u_i^*(t), i \in I), t \in [\tau, \tau + h[, \tau \in T_h$, которое используется до тех пор, пока не будет измерено и обработано следующее состояние $x^*(\tau + h)$.

В *децентрализованном* случае каждый объект управления имеет собственный регулятор. Этот локальный регулятор для каждого i-го объекта по точному состоянию $x_i^*(\tau), \tau \in T_h$ своего объекта и состояниям $x_j^*(\tau - h), \tau \in T_h, j \in I_i$, остальных объектов в предыдущий момент времени строит локальное управляющее воздействие $u_i^*(t), t \in [\tau, \tau + h[, \tau \in T_h$, которое используется до тех пор, пока не будет измерено и обработано следующее состояние $x_i^*(\tau + h)$.

В настоящей главе рассматривается случай централизованного управления.

При централизованном управлении динамическая модель (2.1) рассматривается как одна большая система:

$$\dot{x} = A(t)x + B(t)u, \quad x(t_0) = x_0, \tag{2.7}$$

где $x = \begin{pmatrix} x_1 \\ \vdots \\ x_q \end{pmatrix}, u = \begin{pmatrix} u_1 \\ \vdots \\ u_q \end{pmatrix}, x_0 = \begin{pmatrix} x_{1,0} \\ \vdots \\ x_{q,0} \end{pmatrix}$; $A,\ B$ — соответствующие блочные матрицы.

Система (2.7) имеет единственный регулятор, который в режиме реального времени на основе измеренного точного текущего состояния $x^*(\tau) \in \mathbb{R}^n$ вырабатывает управляющий сигнал $u^*(\tau) \in \mathbb{R}^r$.

Для этого оптимальный регулятор для каждого момента $\tau \in T_h$ решает ([3], с. 1713) задачу $P(\tau)$:

$$c'x(t_f) \to \min_u,$$

$$\dot{x} = A(t)x + B(t)u,$$

$$x(\tau) = x^*(\tau),$$

$$Hx(t_f) = g,$$

$$u \in U, \quad t \in [\tau, t_f],$$

где $U = U_1 \times U_2 \times ... \times U_q$.

В классе дискретных управлений задача $P(\tau)$ сводится к задаче линейного программирования. При сведении можно пользоваться последовательным или параллельным подходами (см. главу 1).

Рассмотрим параллельный подход. Для этого запишем формулу Коши для системы в непрерывном времени:

$$x(t) = F(t, t_0)x_0 + \int_{t_0}^{t} F(t, \theta)B(\theta)u(\theta)\, d\theta. \tag{2.8}$$

Пусть $t_0 = s, s \in T_h$. Применим данную формулу для дискретного управления и подставим $t = s + h$:

$$x(s + h) = A_h(s)x(s) + B_h(s)u(s),$$

где $A_h(s) = F(s + h, s),\ B_h(s) = \int\limits_{s}^{s+h} F(s + h, \theta)B(\theta)\, d\theta.$

В итоге задача (2.6) примет вид:

$$c'x(t_f) \to \min_u,$$

$$x(s + h) = A_h(s)x(s) + B_h(s)u(s),\ s \in T_h,$$

$$x(t_0) = x_0,$$

$$Hx(t_f) = g,$$

$$u_* \le u(s) \le u^*,\ s \in T_h,$$

где неизвестными являются $x(s),\ s \in T_h \cup t_f$ и $u(s),\ s \in T_h$.

Таким образом, при использовании параллельного подхода имеем всего $(N + 1)n + Nr$ неизвестных и $Nn + m$ основных ограничений.

Рассмотрим последовательный подход. Подставим в формулу Коши (2.8) $t = t_f$, а получившуюся величину $x(t_f)$ подставим в исходную задачу (2.6). Тогда (2.6) имеет вид:

$$\sum_{s \in T_h} c_h(s)u(s) \to \min, \tag{2.9}$$

$$\sum_{s\in T_h} d_h(s)u(s) = \tilde{g},$$

$$u_* \le u(s) \le u^*, \ s\in T_h$$

где

$$c_h(s) = \int\limits_{s}^{s+h} c'F(t_f,\theta)B(\theta)\,d\theta, \ c_h(s)\in\mathbb{R}^r,$$

$$d_h(s) = \int\limits_{s}^{s+h} HF(t_f,\theta)B(\theta)\,d\theta, \ d_h(s)\in\mathbb{R}^{m\times r};$$

$$\tilde{g} = g - HF(t_f,t_0)x_0.$$

В итоге при использовании последовательного подхода имеем всего Nr неизвестных и m основных ограничений.

2.2 Построение централизованной обратной связи в реальном времени

Рассмотрим централизованный случай управления группой динамических объектов. То есть пусть имеется общий центр управления, которой на каждом промежутке времени $[\tau,\tau+h[,\tau\in T_h$ по точной информации о состоянии $x^*(\tau) = \begin{pmatrix} x_1^*(\tau)\\ \vdots \\ x_q^*(\tau)\end{pmatrix}$ группы вырабатывает управляющие воздействия $u^*(t) = \begin{pmatrix} u_1^*(t)\\ \vdots \\ u_q^*(t)\end{pmatrix}, t\in[\tau,\tau+h[,\tau\in T_h$, которое используется до тех пор, пока не будет измерено и обработано следующее состояние $x^*(\tau+h)$.

Алгоритм построения централизованной обратной связи $u^*(\tau,x^*(\tau)),\tau\in T_h$, имеет вид:

1. Положить $\tau = t_0, \ x^*(\tau) = x_0$.
2. Найти оптимальную программу $u^*(t|\tau,x^*(\tau)), t\in[\tau,t_f]$.
3. Задать значение обратной связи для позиции $(\tau,x^*(\tau))$:

$$u^*(\tau,x^*(\tau)) = u^*(\tau|\tau,x^*(\tau)).$$

4. В момент $\tau+h$ измерить $x^*(\tau+h)$, положить $\tau := \tau+h$, при $\tau < t_f$

вернуться к шагу 2.

Таким образом, построили задачу оптимального управления группой линейных взаимосвязанных систем, показали два подхода для её сведения к задаче линейного программирования, описали централизованный и децентрализованный способы управления, а также разобрали алгоритм построения обратной связи в реальном времени для централизованного случая.

ГЛАВА 3

РЕЗУЛЬТАТЫ ЧИСЛЕННЫХ ЭКСПЕРИМЕНТОВ

В качестве примера, иллюстрирующего алгоритм построения обратной связи в реальном времени для централизованного случая, будем рассматривать задачу оптимального управления для объекта, состоящего из трех взаимосвязанных систем и описываемого линейными уравнениями с аддитивными возмущениями w:

$$\begin{aligned} \ddot{z}_1 &= -2kz_1 + kz_2 + u_1 + w_1, \\ \ddot{z}_2 &= -2kz_2 + kz_1 + kz_3 + u_2 + w_2, \\ \ddot{z}_3 &= -2kz_3 + kz_2 + u_3 + w_3. \end{aligned} \tag{3.1}$$

Управляющее воздействия ограничены по модулю $|u_i(t)| \leq L, t \in [t_0, t_f], i \in I = \{1,\ 2,\ 3\}$, и в момент времени t_f требуется перевести объект (3.1) на терминальное множество, задаваемое ограничениями $|z_i(t_f)| \leq d_1,\ |\dot{z}_i(t_f)| \leq d_2, i \in I$, а критерий качества имеет вид:

$$\int_{t_0}^{t_f} \sum_{i \in I} |u_i(t)|\, dt \to min.$$

Параметры задачи выберем следующим образом: $k = 10,\ L = 1,\ t_0 = 0,\ t_f = 4.5;\ d_i = 0.1,\ \dot{z}_i(0) = 1,\ z_i(0) = 1,\ i \in I$. Пусть также $|T_h| = 100$, $w(t) = \begin{pmatrix} 0.7\, sin\, 2t \\ 0.1\, cos\, t \\ -0.05\, cos\, 3t \end{pmatrix}$.

3.1 Сведение задачи к задаче линейного программирования

В отличие от разобранного ранее случая, в данной задаче добавилось возмущение $w(t)$, однако это не сильно влияет на алгоритм построения решения, разве что на построение траектории.

Произведем замену:

$$x_1 = z_1, \qquad x_3 = z_2, \qquad x_5 = z_3,$$
$$x_2 = \dot{z}_1, \qquad x_4 = \dot{z}_2, \qquad x_6 = \dot{z}_3.$$

Введем переменные $n = 6$ – число равенств замены; $r = 3$ – число взаимосвязанных систем; $m = 2n$ – число ограничений на $x_i,\ i = 1...n$. Тогда видно, что задача имеет вид (см главу 2):

$$\int\limits_{t_0}^{t_f} \sum_{i \in I} |u_i(t)|\, dt \to min, \tag{3.2}$$

$$\dot{x}(t) = Ax(t) + Bu(t) + Mw(t),$$

$$x(0) = x_0,\ t \in [t_0, t_f],\ Hx(t_f) \leq g,$$

где

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -2k & 0 & k & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ k & 0 & -2k & 0 & k & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & k & 0 & -2k & 0 \end{pmatrix},\ B = M = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix},\ x_0 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix},$$

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix},\ g = \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \end{pmatrix}.$$

Используя последовательный подход, сведем задачу (3.2) к задаче линейного программирования, аналогично тому, как сводили задачу (2.6) к задаче (2.9). Если после этого применить замену $u(s) = z(s) - v(s)$, $s \in T_h$, то в итоге задачу (3.2) можно записать следующим образом:

$$\sum_{s \in T_h} c' \begin{pmatrix} z(s) \\ v(s) \end{pmatrix} \to min, \tag{3.3}$$

$$\sum_{s \in T_h} (d_h(s), -d_h(s)) \begin{pmatrix} z(s) \\ v(s) \end{pmatrix} \le \tilde{g},$$

где $c = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^{2r}$; $d_h(s)$ и $\tilde{g}$ задаются аналогично тому, как они задавались во второй главе: $d_h(s) = \int\limits_s^{s+h} HF(t_f, \theta)B(\theta)\, d\theta$, $d_h(s) \in \mathbb{R}^{m \times r}$; $\tilde{g} = g - HF(t_f, t_0)x_0, \tilde{g} \in \mathbb{R}^m$.

3.2 Построение обратной связи и соответствующей траектории

Построение программного решения — первый шаг для построения обратной связи.

Для записи левой части ограничений задачи (3.3) в матричном виде в функции

```
function Ale = Form_LP()

    for_d = @(t)H*F(t_f-t)*b;
    get_d_h = ...
        @(s)integral(for_d,s,s+h,'ArrayValued', true);
    d_h_values = zeros(m, r, N_initial);

    for i = 1:N_initial
        d_h_values(:,:,i) = get_d_h(t_0 + i*h-h);
    end

    Ale = [];
    for i = 1:N_initial
        Ale = [Ale  d_h_values(:,:,i) ...
            -d_h_values(:,:,i)];
```

```
14        end
15    end
```

формируется матрица `Ale` (строки 11-14), которая будет иметь вид:

$$\begin{pmatrix} d_h(t_0) & -d_h(t_0) & \ldots & d_h(t_f) & -d_h(t_f) \end{pmatrix}. \tag{3.4}$$

Матрица `Ale` будет являться одним из параметров стандартной процедуры linprog для решения задач линейного программирования.

В функции

```
function u = P(tau, z)

    N = round((t_f - tau)/h);%new N for new  tau

    g_wave = g - H * F(t_f - tau) * z;

    c = ones(1, 2*r*N);

    ub = L*ones(2*r*N, 1);

    startBlockNumber = N_initial - N + 1; % ...
      1...N_initial

    Opt=optimset('TolFun',1e-9,'TolX',1e-9);

    z_and_v = ...
      linprog(c,Ale(:,2*r*startBlockNumber - 5 : ...
      2*r*N_initial),g_wave,[],[],zeros(2*r*N, ...
      1),ub,Opt);

    zv = reshape(z_and_v, 2*r, N);
    u  = zv(1:r,:) - zv(r+1:2*r,:); % r /times N
end
```

определяется оптимальная программа для позиции `(tau, z)`. Входным параметром является позиция `(tau, z)`.

Результатом стандартной процедуры linprog будет вектор `z_and_v`, который можно записать в виде:

$$\begin{pmatrix} z(\tau) \\ v(\tau) \\ z(\tau+h) \\ v(\tau+h) \\ \vdots \\ z(t_f) \\ v(t_f) \end{pmatrix}.$$

Вектор преобразуется в позиционное решение строками кода под номером 17 и 18.

То есть для получения программного решения позиции `(tau, z)` достаточно один раз вызвать функцию `function u = P(tau, z)`. Оно вместе с соотвествующей траекторией строится для сравнения с оптимальной обратной связью в части кода, записанной ниже:

```
U_0 = P(t_0, x_0);

X_0 = trajectory(x_0, t_0, t_f, U_0);
```

Функция

```
function x = trajectory(x0, t_begin, t_end, u)
    N = round((t_end - t_begin)/h);
    x = zeros(n, N);
    x(:,1) = x0;
    for j = 1:N
        curr = t_begin + (j-1)*h;
        next = t_begin + j*h;
        x(:, j+1) = F(h) * x(:,j) + ...
           integral(@(t) F(next - t)*b,curr, next, ...
           'ArrayValued', ...
           true,'RelTol',0,'AbsTol',1e-12)*u(:, j) ...
           + integral(@(t) F(curr - t)* M * w(t), ...
           curr, next,'ArrayValued', ...
           true,'RelTol',0,'AbsTol',1e-12);
    end
end
```

принимает параметры для построения траектории: `x0` — начальное состояние, `t_begin`, `t_end`, `u` — начало, конец управление и само управление. Она возвращает траекторию для указанного отрезка времени.

Оптимальная обратная связь ищется в соответсвии с описанным ранее (в главе 2) алгоритмом следующим образом:

```
X = zeros(n, N_initial);

X(:, 1) = x_0;

U = zeros(r, N_initial);

for k = 1:N_initial
    tau = t_0 + (k - 1)*h;

    u_res = P(tau, X(:, k));

    U(:, k) = u_res(:, 1);

    step_trajectory = trajectory(X(:, k), tau, ...
        tau + h, U(:, k));

    X(:, k + 1) = step_trajectory(:,2);
end
```

Стоит заметить, что в функции `function u = P(tau, z)` для отыскания позиционного решения для позиции `(tau, z)` в процедуру linprog вноситься часть уже построенной (функцией `function Ale = Form_LP()`) матрицы `Ale`. Для позиции `(t_0, x_0)` в linprog подается матрица вида (3.4), однако для следующей позиции `(t_0 + h, X(:, 2))` в linprog вносится матрица без первых $2r$ столбцов:

$$\begin{pmatrix} d_h(t_1) & -d_h(t_1) & \dots & d_h(t_f) & -d_h(t_f) \end{pmatrix}.$$

3.3 Результаты. Сравнение с программным решением

Для оценки результатов перенесем полученные траектории на фазовые графики. Для r объектов управления получится r фазовых графиков (про-

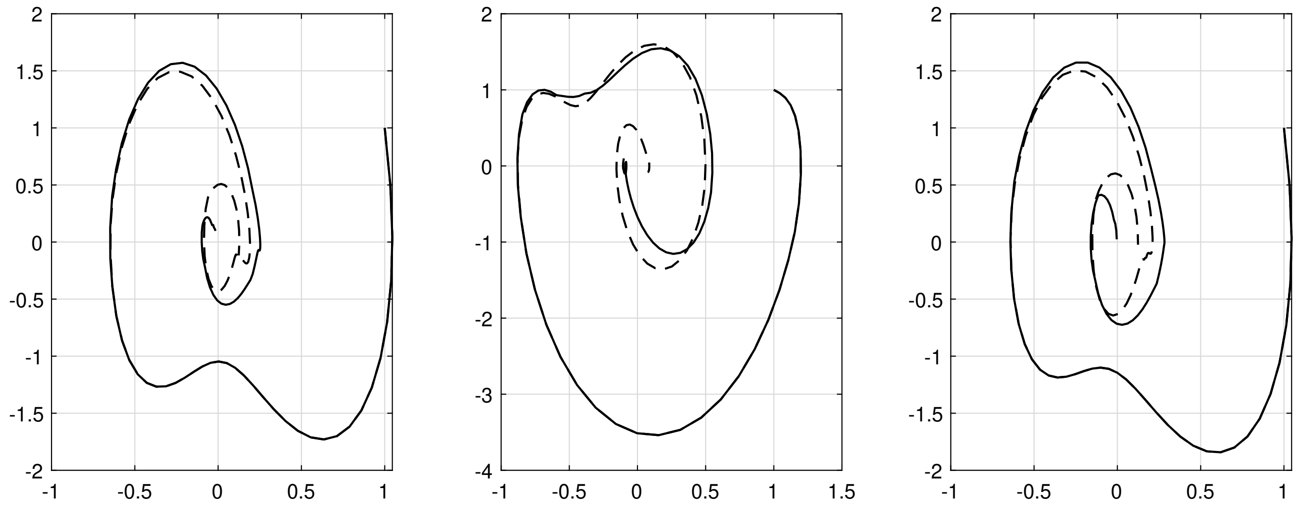

Рис. 3.1: Фазовые графики

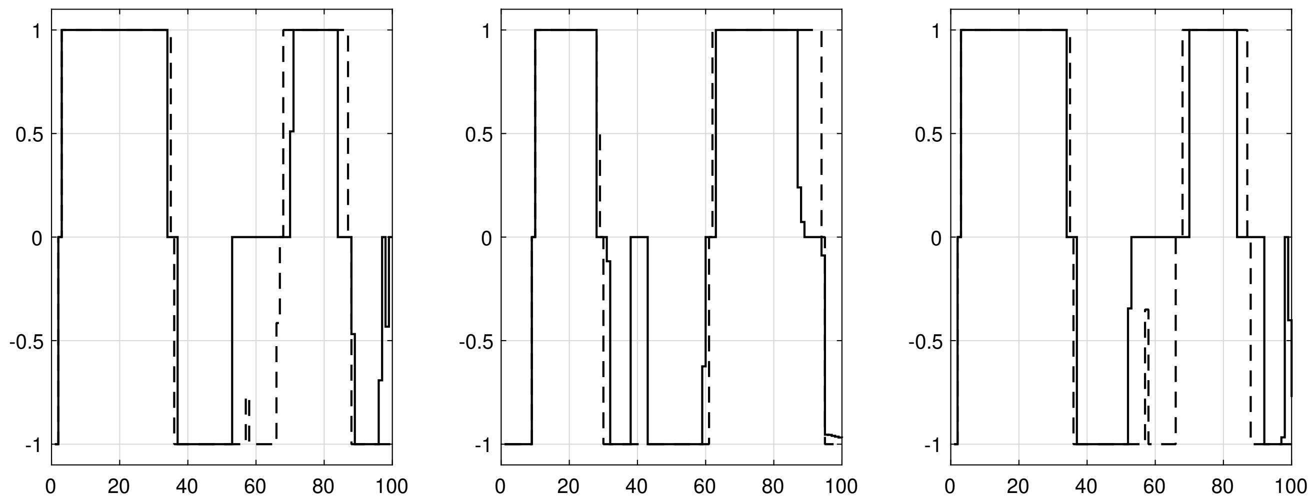

Рис. 3.2: Соответствующие фазовым графикам управления

граммное решение отмечено курсивом):

На фазовых графиках видно, что траектории не попадают на терминальное множество. Это происходит из-за наличия неучтенных возмущений w, вносимых в систему на протяжении времени управления. При построении оптимальных обратных связей эти возмущения учитываются посредством уточнения состояния системы для каждого момента $\tau \in T_h$.

Таким образом разобрали задачу оптимального управления для объекта, состоящего из трех взаимосвязанных систем, свели данную задачу (3.1) к задаче линейного программирования, построили программное решение и обратную связь, вывели фазовые графики и сравнили решения.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1 Понтрягин Л.С., Болтянский В.Г., Гамкрелидзе Р.В., Мищенко Е.Ф. Математическая теория оптимальных процессов. М.: Физматгиз, 1961. 392 с.

2 Беллман, Р. Динамическое программирование / Р. Беллман. – М.:Инностранная литература, 1960. – 400 с.

3 Асимптотически субоптимальное управление динамическими системами со слабыми взаимосвязями/Дмитрук Н.М., Калинин А.И. // Журнал вычислительной математики и математической физики, 2016, том 56, № 10, с. 1711–1724.

4 Оптимальное децентрализованное управление группой динамических объектов/Габасов Р., Дмитрук Н.М., Кириллова Ф.М. // Журнал вычислительной математики и математической физики, 2008, том 48, М 4, с. 593-609.