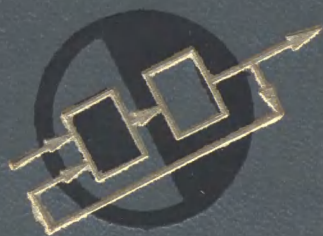


Д. ТАБАК, Б. КУО · ОПТИМАЛЬНОЕ УПРАВЛЕНИЕ И МАТЕМАТИЧЕСКОЕ ПРОГРАММИРОВАНИЕ



Д. ТАБАК, Б. КУО

ОПТИМАЛЬНОЕ
УПРАВЛЕНИЕ
И МАТЕМАТИЧЕСКОЕ
ПРОГРАММИРОВАНИЕ





ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ТЕХНИЧЕСКОЙ КИБЕРНЕТИКИ

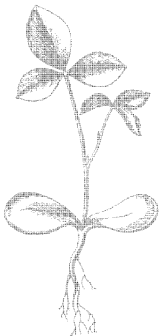
**ИЗДАТЕЛЬСТВО «НАУКА»
ГЛАВНАЯ РЕДАКЦИЯ
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ
МОСКВА 1975**

Д. ТАБАК, Б. КУО

ОПТИМАЛЬНОЕ УПРАВЛЕНИЕ И МАТЕМАТИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

ПЕРЕВОД С АНГЛИЙСКОГО
Л. А. МЕЕРОВИЧА

ПОД РЕДАКЦИЕЙ
Я. З. ЦЫПКИНА



ИЗДАТЕЛЬСТВО «НАУКА»
ГЛАВНАЯ РЕДАКЦИЯ
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ
МОСКВА 1975

6Ф6.5

Т 12

УДК 62-50

Оптимальное управление и математическое программирование. Д. Табак, Б. Куо, перев. с англ. Главная редакция физико-математической литературы издательства «Наука», М., 1975, 280 стр.

Книга посвящена систематическому изложению методов математического программирования применительно к задачам оптимального управления. В ней рассмотрены способы оптимизации линейных и нелинейных систем, как непрерывных, так и дискретных, а также систем с распределенными параметрами и стохастических. Основное внимание уделяется численным приемам построения решений, иллюстрируемых большим количеством примеров.

Табл. 25. Илл. 57. Библ. 285 назв.

DANIEL TABAK

Associate Professor of Automatic Control
Rensselaer Polytechnic Institute of Connecticut Hartford

BENJAMIN C. KUO

Professor of Electrical Engineering
University of Illinois Urbana

OPTIMAL CONTROL BY MATHEMATICAL PROGRAMMING

PRENTICE-HALL, INC. ENGLEWOOD CLIFFS, NEW JERSEY, 1971

Т $\frac{30502-094}{053(02)-75}$ 180-75

© Перевод на русский язык.
Главная редакция физико-
математической литературы
издательства «Наука», 1975.

От редактора перевода	7
Из предисловия авторов	9
Глава 1. Введение	11
Литература	19
Глава 2. Математическое программирование	22
2.1. Общие понятия (22). 2.2. Формулировка задачи математического программирования (22). 2.3. Пример задачи математического программирования (24). 2.4. Классификация задач математического программирования (25). 2.5. Выпуклость (29). 2.6. Теорема Куна — Таккера (32). 2.7. Двойственность (35).	
Литература	39
Глава 3. Численное решение задач математического программирования	41
3.1. Линейное программирование (41). 3.2. Квадратичное программирование (53). 3.3. Нелинейное программирование (58). 3.4. Методы минимизации без ограничений (71).	
Литература	76
Глава 4. Оптимальное управление и математическое программирование	79
4.1. Введение (79). 4.2. Формулировка задачи (80). 4.3. Математическое программирование и оптимальное управление (82). 4.4. Соображения о вычислениях (86).	
Литература	90
Глава 5. Непрерывные нелинейные системы	93
5.1. Введение (93). 5.2. Формулировка задачи (94). 5.3. Линейная система с заданными временными интервалами (98). 5.4. Задача минимального расхода топлива (103). 5.5. Задача встречи при минимальном расходе топлива (105). 5.6. Задача управления, оптимального по быстродействию (113). 5.7. Использование обобщенного программирования (125). 5.8. Применение для вычисления оптимального управления ракетным ядерным реактором (136).	
Литература	144
Глава 6. Непрерывные нелинейные системы	146
6.1. Введение (146). 6.2. Управление процессом отравления ксеноном в ядерных реакторах (146). 6.3. Аппроксимация более высокого порядка; задача оптимизации траектории (152). 6.4. Вычисление с помощью ЦВМ оптимальных параметров нелиней-	

	ных систем управления (155). 6.5. Метод внутреннего штрафа (163). 6.6. Решение двухточечных граничных задач (164).	
	Литература	166
Глава 7.	Дискретные линейные системы	168
	7.1. Введение (168). 7.2. Примеры задач минимизации (169). 7.3. Расчет цифрового регулятора (176). 7.4. Системы, управляемые в дискретные моменты при неизвестных интервалах (184). 7.5. Импульсные системы с квантованным управлением (193). 7.6. Заключительные замечания (206).	
	Литература	206
Глава 8.	Дискретные нелинейные системы	208
	8.1. Введение (208). 8.2. Пример: нелинейная система второго порядка (211). 8.3. Нелинейная система управления антенной (215). 8.4. Нелинейная система с цифровым управлением (218). 8.5. Расчет цифрового регулятора для нелинейной импульсной системы (226). 8.6. Итерационное решение (233).	
	Литература	236
Глава 9.	Стохастические системы	238
	9.1. Введение (238). 9.2. Комбинированный метод Монте-Карло и математического программирования (238). 9.3. Статистический расчет импульсных систем управления (241). 9.4. Применение к задачам оптимальной оценки (250). 9.5. Идентификация (254).	
	Литература	262
Глава 10.	Системы с распределенными параметрами	264
	10.1. Введение (264). 10.2. Задача оптимизации для уравнения теплопроводности (264). 10.3. Импульсные многомерные системы с распределенными параметрами (267).	
	Литература	274
	Именной указатель	276
	Предметный указатель	278

Проблема оптимальности уже продолжительное время находится в центре внимания специалистов по управлению. Период формулировок этой проблемы и различных подходов к ее строгому решению постепенно проходит. Большую популярность завоевали принцип максимума, метод динамического программирования, метод математического программирования и ряд других методов, которые вошли в золотой фонд теории оптимального управления.

Аналитическое решение разнообразных задач оптимального управления на основе упомянутых выше методов возможно лишь в крайне простых случаях, которые слишком далеки от запросов современной практики. Поэтому важное значение приобретают алгоритмические или, если угодно, численные методы оптимизации. Этому вопросу уделяется большое внимание в математических и технических журналах. Стали появляться и монографии. Большая часть этих работ посвящена различным математическим вопросам обоснования тех или иных алгоритмических методов.

В предлагаемой советскому читателю книге известных американских специалистов доцента Коннектикутского политехнического института Дэниэла Табака и профессора Иллинойского университета Бенджамена Куо основной упор сделан на приложении математического программирования к решению различных задач оптимального управления.

В ней приводятся краткие сведения по математическому программированию (гл. 2) и по методам численного решения задач линейного, квадратичного и нелинейного программирования (гл. 3). Обсуждается связь задач оптимального управления и задач математического программирования (гл. 4). Подробно излагается применение математического программирования к решению задач оптимального управления в линейных (гл. 5) и

нелинейных (гл. 6) непрерывных системах, в линейных (гл. 7) и нелинейных (гл. 8) дискретных системах, в стохастических системах (гл. 9) и в системах с распределенными параметрами (гл. 10).

Книга написана сжато. Она содержит большое число примеров, изучение которых несомненно будет полезно для специалистов, сталкивающихся с подобными задачами. Ряд полезных сведений и советов облегчают применение методов оптимизации.

В переводе книги устранены неточности и ошибки имевшиеся в оригинале. Библиография дополнена ссылками на отечественные работы.

Специалисты в области математического программирования не найдут в ней чего-либо нового, но все же тот, кто хочет познакомиться с конкретными приемами решения прикладных задач, возможно, найдет в ней ответы на многие интересующие его вопросы.

Книга будет полезна не только научным работникам, но и аспирантам и студентам, специализирующимся в области автоматического управления.

*Москва,
октябрь 1973*

Я. Цыпкин

Оптимальное управление, которое уже нашло большое число различных применений в космонавтике и в химической промышленности, в ядерных реакторах и в транспортных системах и много других, является одной из наиболее активно разрабатываемых областей современной техники. В течение последних десяти лет появился целый ряд учебников и монографий, посвященных оптимальному управлению. Одной из основных практических задач в этой области является численное решение задач оптимального управления. В течение последнего времени был предложен ряд методов решения задач этого типа.

После второй мировой войны было создано математическое программирование — мощный аппарат решения конечномерных задач оптимизации. Вначале оно нашло применение главным образом в задачах исследования операций. Только в последние годы были предприняты попытки применить математическое программирование для численного решения задач оптимального управления. На многих практических примерах было показано, что математическое программирование позволяет получить численное решение там, где другие методы оказываются бессильными.

Результаты работ, посвященных применению математического программирования к задачам оптимального управления, разбросаны по различным журналам, диссертациям и отчетам. Цель данной книги — собрать этот материал и изложить его с единой точки зрения.

Ядро книги возникло из диссертации Д. Табака, выполненной под руководством Б. Куо в Иллинойском университете. Позже это исследование было продолжено Табаком в Корпорации исследований и разработок Уолф. Значительная часть книги основана на результатах других исследователей, работающих в этой области.

Авторы обязаны следующим лицам, работающим в области применения математического программирования в оптимальном управлении, за разрешение включить в эту книгу полученные ими результаты: проф. Фегли (K. A. Fegley), доктору Порцелли (G. Porcelli), проф. Розену (J. B. Rosen), доктору Фасу (A. F. Fath), подполковнику Веспи (C. M. Waespy), проф. Молеру (R. R. Mohler), проф. Лэсдону (L. Lasdon), доктору Джизмаджану (C. S. Jizmagian), м-ру Гаррисону (R. W. Harrison), доктору Кэлизу (A. J. Calise), проф. Пьеру (D. A. Pierre), доктору Лорчирачункулу (V. Lorchirachoonkul), проф. Пирсону (B. L. Pierson), проф. Киму (M. Kim).

Весьма ценную помощь и поддержку оказало руководство Корпорации Уолф особенно м-р Артур Деннис (Arthur R. Dennis) и доктор Уильям Уэлс (William T. Wells).

Авторам приятно поблагодарить проф. Сэйджа (A. P. Sage) и проф. Хиггинса (T. J. Higgins), которые любезно взяли на себя труд прочесть рукопись и дали ценные советы.

Особенно благодарны авторы своим женам Пнине Табак и Маргарет Куо за их терпение и понимание.

*Б. Куо
Д. Табак*

В течение последних десяти лет серьезное внимание уделяется исследованиям в области оптимального управления. Результатом этого явилось большое число статей и ряд книг [1—8] *), посвященных методам оптимизации в теории управления. Методы проб и ошибок в расчетах систем управления вытесняются научными методами, основанными на прикладной математике и использующими быстродействующие ЦВМ. Для раннего этапа исследований систем управления характерными были такие вопросы, как анализ устойчивости, частотные характеристики, корневой годограф, фазовая плоскость, и т. п. Результаты подобных исследований широко известны и получили практическое применение. Тем не менее они пригодны только для не слишком сложных задач, возникающих при расчетах одномерных систем, систем с постоянными параметрами и систем без ограничений на переменные.

С наступлением космической эры техника управления столкнулась с проблемой проектирования самых разнообразных систем, к которым предъявляются жесткие требования. Большой интерес вызвали также расчеты социальных систем, ядерных реакторов и транспортных систем, которые потребовали значительного совершенствования теории и техники управления.

Вот некоторые характерные черты и типы этих современных систем: большое число входов и выходов (многомерность), ограничения, наложенные на фазовые переменные и управления, стохастические системы, неизвестные или переменные параметры, большие системы и системы с запаздыванием.

Исходным положением современной теории управления является система уравнений состояния, описывающих поведение динамической системы.

) См. также [20—36*]. (Прим. ред).

В непрерывной системе уравнением состояния является система обыкновенных дифференциальных уравнений первого порядка

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t), t], \quad (1.1)$$

где $\mathbf{x}(t)$ есть $(n \times 1)$ -мерный вектор, представляющий состояние системы (фазовые переменные), $\mathbf{u}(t)$ — $(r \times 1)$ -мерный вектор входных сигналов (управление) и \mathbf{f} — векторная функция. Для системы с запаздыванием переменных состояния или входов уравнения состояния представляют собой систему дифференциально-разностных уравнений следующего вида:

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}[\mathbf{x}(t - T), \mathbf{u}(t - T), t], \quad (1.2)$$

где T — запаздывание. Для импульсных систем уравнения состояния — это система разностных уравнений первого порядка

$$\mathbf{x}(k + 1) = \mathbf{f}_k[\mathbf{x}(k), \mathbf{u}(k)], \quad (1.3)$$

где k обозначает k -й момент времени.

Обычно задача оптимального управления формулируется следующим образом. Даны:

1) уравнения состояния;

2) система граничных условий, наложенных на переменные состояния в начальный и в конечный моменты времени;

3) система ограничений, наложенных на переменные состояния и управления.

Требуется найти такое допустимое управление, при котором показатель качества (или целевая функция) будет минимальной или максимальной.

Граничные условия, наложенные на переменные состояния, выражаются в виде $\mathbf{x}(t_0) = \mathbf{x}_0$ и $\mathbf{x}(t_f) \in S$, где t_0 и t_f являются соответственно начальными и конечными моментами времени, а S обозначает целевое множество. В то время как t_0 всегда фиксировано, конечный момент времени t_f не обязательно задан.

Часто оказывается, что физические свойства компонентов системы делают необходимыми ограничения, наложенные на переменные состояния и управления. Так,

например, амплитудное ограничение управления $\mathbf{u}(t)$ может быть выражено неравенством $\|\mathbf{u}(t)\| \leq M$.

Обычно показатель качества выражают скалярной величиной вида

$$J = G(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} F(\mathbf{x}(t), \mathbf{u}(t), t) dt, \quad (1.4)$$

где G и F — скалярные функции.

При расчете систем оптимального управления используются следующие основные теоретические методы:

- 1) вариационное исчисление [9],
- 2) принцип максимума (или минимума) [1, 2],
- 3) динамическое программирование [10, 11].

Если в задаче оптимального управления ограничения на переменные состояния и управления отсутствуют, то для ее решения может быть применен классический метод вариационного исчисления, разработанный математиками сотни лет тому назад. В зависимости от вида показателя качества обычно различают три класса задач оптимизации в вариационном исчислении, а именно задачу Лагранжа, задачу Майера и задачу Больца. Так, если задача заключается в отыскании экстремума функционала

$$J = \int_{t_0}^{t_s} F(\mathbf{x}(t), \mathbf{u}(t), t) dt, \quad (1.5)$$

то мы имеем задачу Лагранжа. Если показатель качества имеет вид

$$J = G(\mathbf{x}(t_f), t_f), \quad (1.6)$$

то задача классифицируется как задача Майера. Задача об оптимальном быстродействии и задача терминального управления являются хорошо известными примерами задачи Майера. Так, если в уравнении (1.6) $G = t_f$, то при минимизации G мы имеем задачу об оптимальном быстродействии. Однако, соответственно определяя F и G , мы можем определить задачу либо как задачу Майера, либо как задачу Лагранжа. Например, если в (1.5) взять $F \equiv 1$, то мы вновь получим задачу на быстродействие.

Применение вариационного исчисления для решения задачи оптимального управления приводит к известному

уравнению Эйлера — Лагранжа, которое должно быть решено при заданных граничных условиях для получения оптимальной траектории и оптимального управления. Основная трудность, возникающая при попытках получить численное решение уравнения Эйлера — Лагранжа, определяется граничными условиями, заданными в двух конечных точках. Это приводит к двухточечной граничной задаче, аналитическое решение которой, за исключением нескольких очень простых случаев, связано с большими трудностями. Получение численного решения двухточечной граничной задачи также достаточно сложно.

В 1956 г. Понтрягин и его сотрудники Болтянский и Гамкрелидзе разработали принцип максимума (или минимума) для непрерывных систем, дающий необходимые условия оптимальности для задач с ограничениями на управление. При применении принципа максимума в задачу должны быть введены переменные, аналогичные множителям Лагранжа. Эти переменные обычно называют переменными сопряженной задачи и обозначают $p_1(t)$, $p_2(t)$, ..., $p_n(t)$. В векторной форме $\mathbf{p}(t)$ обозначает векторную переменную сопряженной системы. Скалярная функция H , которая в общем случае является функцией \mathbf{x} , \mathbf{p} , \mathbf{u} и t , называется *функцией Гамильтона* задачи и определяется следующим образом:

$$H(\mathbf{x}, \mathbf{p}, \mathbf{u}, t) = F(\mathbf{x}, \mathbf{u}, t) + \sum_{i=1}^n p_i(t) f_i(\mathbf{x}, \mathbf{u}, t). \quad (1.7)$$

Принцип максимума Понтрягина для частной задачи оптимального управления кратко можно изложить следующим образом:

Дана система, описываемая уравнением

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t), t] \quad (1.8)$$

и граничными условиями

$$\mathbf{x}(t_0) = \mathbf{x}_0$$

и

$$\mathbf{x}(t_f) \in S \quad (S \text{ — целевое множество}).$$

В данном случае предполагается, что t_f фиксировано. Необходимо найти допустимое управление $\mathbf{u}(t)$, которое

минимизирует показатель качества

$$J = \int_{t_0}^{t_f} F(\mathbf{x}, \mathbf{u}, t) dt. \quad (1.9)$$

Для простоты ограничимся задачей Лагранжа. Вектор управления подчинен некоторому ограничению, наложенному на $\mathbf{u}(t)$, типа $\mathbf{u}(t) \in U$ для всех t .

Пусть $\mathbf{u}^0(t)$ есть оптимальное управление и $\mathbf{x}^0(t)$ — оптимальная траектория. Тогда существует вектор сопряженного состояния $\mathbf{p}^0(t)$ такой, что

$$\frac{dx_i^0(t)}{dt} = \frac{\partial H}{\partial p_i^0(t)} \quad \text{на оптимальной траектории} \quad (1.10)$$

$$\frac{dp_i^0(t)}{dt} = - \frac{\partial H}{\partial x_i^0(t)} \quad \text{на оптимальной траектории,} \quad (1.11)$$

где H — гамильтониан, определенный в (1.7). При этом оптимальное управление удовлетворяет неравенству

$$H(\mathbf{x}^0, \mathbf{u}^0, \mathbf{p}^0, t) \geq H(\mathbf{x}^0, \mathbf{u}, \mathbf{p}^0, t) \quad (1.12)$$

для всех $\mathbf{u} \in U$. Кроме того, должно быть выполнено условие трансверсальности $\mathbf{p}^0(t_f)$ по отношению к целевому множеству S при $\mathbf{x}^0(t_f)$. Другими словами, $\mathbf{p}^0(t_f)$ должно быть ортогонально плоскости, касательной к S в $\mathbf{x}^0(t_f)$. С более общей формулировкой принципа максимума (или минимума) для различных других случаев читатель может познакомиться, обратившись к источникам [1, 2].

Рассмотрев только приведенную выше формулировку частной задачи оптимального управления, можно видеть, что начальные условия и условия трансверсальности вновь приводят к двухточечной краевой задаче. Эта формулировка еще более усложняется в задачах с ограничениями состояния в форме неравенств.

Метод динамического программирования вначале был разработан Беллманом [10] для дискретных систем. Основой метода является принцип оптимальности, который формулируется следующим образом:

Оптимальная стратегия обладает тем свойством, что, каковы бы ни были начальное состояние и принятое начальное решение, последующие решения должны со-

ставить оптимальную стратегию относительно состояния, возникшего в результате первоначального решения.

Пусть, например, рассматривается дискретный аналог задачи оптимального управления:

$$\text{минимизировать } J = \sum_{k=1}^N F(\mathbf{x}(k), \mathbf{u}(k-1), k) \quad (1.13)$$

при ограничении

$$\mathbf{u}(k) \in U \quad (k=0, 1, \dots, N-1) \quad (1.14)$$

и уравнении состояния

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k), k) \quad (k=0, 1, \dots, N-1), \\ \mathbf{x}(0) &= \mathbf{x}_0. \end{aligned} \quad (1.15)$$

Введем функцию

$$S_i(\mathbf{x}) = \min_{\mathbf{u}(i-1), \mathbf{u}(i), \dots, \mathbf{u}(N-1)} \sum_{k=i}^N F(\mathbf{x}(k), \mathbf{u}(k-1), k), \quad (1.16)$$

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k), k) \quad (k=i-1, i, \dots, N-1), \\ \mathbf{x}(i-1) &= \mathbf{x}. \end{aligned} \quad (1.17)$$

Эта функция удовлетворяет следующему рекуррентному соотношению:

$$S_i(\mathbf{x}) = \min_{\mathbf{u}(i-1)} \{S_{i+1}(\mathbf{x}(i)) + F(\mathbf{x}(i), \mathbf{u}(i-1), i)\}, \quad (1.18)$$

где

$$\mathbf{x}(i) = \mathbf{f}(\mathbf{x}, \mathbf{u}(i-1), i). \quad (1.19)$$

Полученные рекуррентные соотношения в принципе могут быть решены последовательно: по $S_N(\mathbf{x}) \equiv 0$ находится $S_{N-1}(\mathbf{x})$, затем $S_{N-2}(\mathbf{x})$ и т. д., а $S_1(\mathbf{x})$ дает решение исходной задачи:

$$S_1(\mathbf{x}_0) = \min J. \quad (1.20)$$

Для численного решения рекуррентных уравнений (1.18) функцию $S_i(\mathbf{x})$ можно, например, запоминать на сетке из конечного числа точек.

Главный недостаток динамического программирования заключается в чрезмерных требованиях к объему памяти ЦВМ [10, 11]. Так, например, если вектор со-

стояния x имеет размерность n и каждая переменная состояния в допустимой области представлена M дискретными значениями, то необходимо от шага к шагу запоминать таблицу из M^n значений величины $S_i(x)$. В четырехмерной задаче при $M=100$ нужна таблица объемом в 10^8 данных. При нескольких переменных управления поиск оптимального значения становится значительно более трудной задачей. Делалось много попыток ослабить требования к памяти, предъявляемые алгоритмом динамического программирования. Одним из наиболее значительных достижений в этой области является алгоритм приращений состояния, предложенный Ларсоном [11]. Применение метода приращений состояния дает значительное уменьшение требуемого объема памяти. Однако задача решена лишь в некоторой мере. Решение задач оптимального управления с помощью динамического программирования в системах более высокого порядка, чем четвертый, и при нескольких переменных управления все еще связано с большими трудностями, а во многих случаях оказывается вовсе невозможным.

В течение последних десяти лет большое внимание уделялось отысканию эффективных решений двухточечной граничной задачи. Здесь даны ссылки лишь на небольшую часть имеющейся литературы [6, гл. 13—15; 12—18]. Авторами указанных работ и многими другими были решены задачи многих классов. Однако в решении двухточечной граничной задачи все еще существуют большие трудности. Это в особенности относится к следующим классам двухточечных граничных задач:

1. Задачи с нелинейными ограничениями следующего типа:

$$g(x, u) \leq 0, \quad (1.21)$$

где g — нелинейная векторная функция.

2. Серьезные проблемы возникают при решении двухточечной задачи даже без ограничений для нелинейных многомерных систем.

3. Задачи, относящиеся к импульсным системам с ограничениями, наложенными на интервалы дискретизации.

4. Импульсные системы с неравными и априорно неизвестными интервалами дискретизации.

Приведенный перечень далек от полноты.

Эта книга посвящена подходу, отличающемуся от рассмотренных выше. Здесь используются методы математического программирования, основные понятия и алгоритмы которого описаны в гл. 2 и 3. Задача математического программирования состоит в отыскании экстремума функции многих переменных при ограничениях в форме системы неравенств и равенств. Теория и алгоритмы математического программирования исследуются и разрабатываются непрерывно, начиная с 1940 г.

Разработанные методы нашли основное применение в области исследования операций. Интерес к применению методов математического программирования для численного решения задач оптимального управления возник только в 1960 г. Одной из первых работ, посвященных этому вопросу, была работа 1962 г. Заде и Уалена [19]. Вслед за ней появилось значительное число публикаций, посвященных этому вопросу, причем большинство было опубликовано в период 1967—1969 гг. Эти работы подробно рассматриваются в гл. 4—10. В основном в большинстве работ задача оптимального управления переформулируется в виде, типичном для задач математического программирования. Следующий шаг заключается в численном решении задачи с помощью одного из имеющихся алгоритмов, описанных в гл. 3. В некоторых случаях были созданы новые алгоритмы, предназначенные для рассматриваемых задач.

Из многих достоинств применения методов математического программирования назовем следующие:

1. Большинство алгоритмов математического программирования эффективно справляется с ограничивающими неравенствами. Многие сложные ограничения в форме неравенств, наложенные как на переменные состояния, так и на управления, или на интервалы дискретного времени, которые крайне усложняют двухточечную граничную задачу, относительно легко учитываются алгоритмами математического программирования.

2. Алгоритмы математического программирования при задачах со сравнимым числом переменных не требуют столь большого объема памяти ЦВМ, как динамическое программирование.

Отсюда не следует, что методы математического программирования являются универсальным средством для получения численных решений задач оптимального управления. Многие классы задач математического программирования (в частности, нелинейные, невыпуклые) связаны с очень большими вычислительными трудностями. Есть классы задач оптимального управления, для которых другие подходы оказываются более эффективными. Однако для большого класса задач оптимального управления математическое программирование является наиболее эффективным подходом, а в ряде случаев и единственным, фактически применимым на практике.

Применение алгоритмов математического программирования к задачам оптимального управления различных классов рассмотрено в гл. 5—10. Гл. 2 и 3 служат общим введением в основы теории и в вычислительные методы математического программирования. Связь между оптимальным управлением и математическим программированием, как и некоторые общие вопросы вычислений, обсуждаются в гл. 4. Отдельно рассматриваются следующие классы задач оптимального управления: линейные непрерывные системы в гл. 5, нелинейные непрерывные системы в гл. 6, линейные дискретные системы в гл. 7, нелинейные дискретные системы в гл. 8, стохастические системы в гл. 9 и системы с распределенными параметрами в гл. 10. Во всех этих гл. 5—10 в основном обсуждаются пути применения методов математического программирования к различным классам систем и в качестве примеров рассматриваются различные частные случаи.

Литература *)

1. M. Athans, P. L. Falb, Optimal Control: An Introduction to the Theory and its Applications, McGraw-Hill, New York, 1966. [Русский перевод: М. Атанс, П. Фалб, Оптимальное управление, «Машиностроение», 1968.]
2. Л. С. Понтрягин, В. Г. Болтянский, Р. В. Гамкрелидзе, Е. Ф. Мищенко, Математическая теория оптимальных процессов, Физматгиз, 1961.

*) Литература, отмеченная звездочкой, добавлена при переводе (Прим. ред.)

3. E. B. Lee, L. Markus, *Foundations of Optimal Control Theory*, Wiley, New York, 1967. [Русский перевод: Э. Б. Ли, Л. Маркус, *Основы теории оптимального управления*, «Наука», 1972.]
4. D. J. Wilde, C. S. Beightler, *Foundations of Optimization*, Prentice-Hall, Englewood Cliffs, N. J., 1967.
5. V. W. Evtleigh, *Adaptive Control and Optimization Techniques*, McGraw-Hill, New York, 1967.
6. A. P. Sage, *Optimum Systems Control*, Prentice-Hall, Englewood Cliffs, N. J., 1968.
7. G. Leitmann, *Optimization Techniques*, Academic Press, New York, 1962. [Русский перевод: *Методы оптимизации с приложениями к механике космического полета*, под ред. Дж. Лейтмана, «Наука», 1965.]
8. D. G. Luenberger, *Optimization by Vector Space Methods*, Wiley, New York, 1969.
9. И. М. Гельфанд, С. В. Фомин, *Вариационное исчисление*, Физматгиз, 1961.
10. R. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, N. J., 1957. [Русский перевод: Р. Беллман, *Динамическое программирование*, ИЛ, 1960.]
11. R. E. Larson, *State Increment Dynamic Programming*, American Elsevier, New York, 1968.
12. A. E. Bryson, W. F. Denham, *A Steepest Ascent Method for Solving Optimum Programming Problems*, *J. Appl. Mech.* **29**, pp. 247—257 (1962). [Русский перевод: А. Брайсон, В. Денхем, *Прикладная механика*, № 2, 1962.]
13. W. F. Denham and A. E. Bryson, *Optimal Programming Problems with Inequality Constraints II: Solution by Steepest-Ascent* *AIAA J.* **2**, pp. 25—34 (1964). [Русский перевод: В. Денхем, А. Брайсон, *Ракетная техника и космонавтика*, № 2, 1964.]
14. P. Kenneth, R. McGill, *Two-Point Boundary-Value-Problem Techniques*, *Advan. Control Systems* **3**, pp. 69—109 (1966).
15. D. Isaacs, *Algorithms for Sequential Optimization of Control Systems*, *Advan. Control Systems* **4**, pp. 1—71 (1966).
16. R. E. Koop, H. C. Moyer, *Trajectory Optimization Techniques*, *Advan. Control Systems* **4**, pp. 103—155 (1966).
17. D. K. Schamack, *An Initial Value Method for Trajectory Optimization Problems*, *Advan. Control Systems* **5**, pp. 51—132 (1967).
18. J. B. Plant, *Some Iterative Solutions in Optimal Control*, M. I. T. Press, Cambridge, Mass., 1968.
19. L. A. Zadeh, B. H. Whalen, *On Optimal Control and Linear Programming*, *IRE Trans. Automatic Control* **AC-7**, pp. 45—46 (1962).
- 20*. А. А. Фельдбаум, *Основы теории оптимальных автоматических систем*, Физматгиз, 1963.
- 21*. В. Г. Болтянский, *Математические методы оптимального управления*, «Наука», 1966.
- 22*. Н. Н. Моисеев, *Численные методы в теории оптимальных систем*, «Наука», 1971.
- 23*. Р. Беллман, С. Дрейфус, *Прикладные задачи динамического программирования*, «Наука», 1965.

- 24*. Л. Я. Ц л а ф, Вариационное исчисление и интегральные уравнения, «Наука», 1966.
- 25*. А. Б р а й с о н, Х о Ю - ш и, Прикладная теория оптимального управления, «Мир», 1972.
- 26*. Дж. Л е й т м а н, Введение в теорию оптимального управления, «Наука», 1968.
- 27*. К. У. М е р р и э м, Теория оптимизации и расчет систем управления с обратной связью, «Мир», 1967.
- 28*. Современная теория систем управления, под ред. К. Т. Леондеса, «Наука», 1970.
- 29*. К. С п и д и, Р. Б р а у н, Дж. Г у д в и н, Теория управления, «Мир», 1973.
- 30*. А. И. П р о п о й, Элементы теории оптимальных дискретных процессов, «Наука», 1973.
- 31*. Р. А р и с, Дискретное динамическое программирование, «Мир», 1969.
- 32*. Н. Н. К р а с о в с к и й, Теория управления движением, «Наука», 1968.
- 33*. В. Ф. К р о т о в, В. З. Б у к р е е в, В. И. Г у р м а н, Новые методы вариационного исчисления в динамике полета, «Машиностроение», 1969.
- 34*. А. М. Л е т о в, Динамика полета и управление, «Наука», 1969.
- 35*. Р. Г а б а с о в, Ф. М. К и р и л л о в а, Качественная теория оптимальных процессов, «Наука», 1971.
- 36*. А. А. Ф е л ь д б а у м, А. Г. Б у т к о в с к и й, Методы теории автоматического управления, «Наука», 1971.

2.1. Общие понятия

Предметом математического программирования [1—6, 21*—27*] является решение многомерных экстремальных задач. Рассматриваемые переменные составляют некоторое функциональное выражение, которое должно быть минимизировано или максимизировано. Соответствующее выражение обычно называют *целевой функцией* или *показателем качества*.

Обычно требуется, чтобы рассматриваемые переменные удовлетворяли дополнительным неравенствам или равенствам, называемым *ограничениями*. В общем случае как целевая функция, так и ограничения могут быть нелинейными функциями всех или некоторых из рассматриваемых переменных. В тех случаях, когда ограничения отсутствуют, задачу называют *максимизацией* (или *минимизацией*) *без ограничений*.

Любое множество значений рассматриваемых переменных, удовлетворяющее всем ограничениям задачи, называют *допустимым решением* или *планом*. Допустимое решение, которое максимизирует или минимизирует целевую функцию, является *оптимальным решением* (или *оптимальным планом*) задачи математического программирования.

2.2. Формулировка задачи математического программирования

Допустим, что рассматриваемая задача математического программирования содержит n переменных x_i ($i = 1, 2, \dots, n$). Эти n переменных образуют n -мерный *вектор переменных* x . Обозначим через $f(x)$ целевую функцию n переменных x_i ; $f(x)$ — скалярная, в общем случае нелинейная, функция всех или некоторых переменных x_i ($i = 1, 2, \dots, n$).

Ограничения в форме неравенств можно записать следующим образом:

$$g_i(\mathbf{x}) \geq 0, \quad i=1, \dots, q, \quad (2.1)$$

где $g_i(\mathbf{x})$ ($i=1, \dots, q$) представляют собой скалярные нелинейные функции некоторых или всех переменных x_i ($i=1, \dots, n$). Функции $g_i(\mathbf{x})$ можно считать компонентами векторной функции $\mathbf{g}(\mathbf{x})$ и переписать (2.1) в виде

$$\mathbf{g}(\mathbf{x}) \geq 0. \quad (2.2)$$

Аналогично, ограничения в форме равенств можно записать в скалярной форме

$$h_i(\mathbf{x}) = 0, \quad i = 1, \dots, p, \quad (2.3)$$

где $h_i(\mathbf{x})$ ($i=1, \dots, p$) — скалярные нелинейные функции некоторых или всех переменных x_i ($i=1, \dots, n$) или, в векторной форме,

$$\mathbf{h}(\mathbf{x}) = 0. \quad (2.4)$$

В формуле (2.2) 0 представляет собой q -мерный нулевой вектор, а в формуле (2.4) — p -мерный нулевой вектор.

Задачу математического программирования теперь можно сформулировать следующим образом:

$$\left. \begin{array}{l} \text{минимизировать (или максимизировать) } f(\mathbf{x}) \\ \text{при условии} \end{array} \right\} \begin{array}{l} \mathbf{g}(\mathbf{x}) \geq 0, \\ \mathbf{h}(\mathbf{x}) = 0 \end{array} \quad (2.5)$$

или, кратко,

$$\min \{f(\mathbf{x}) \mid g_i(\mathbf{x}) \geq 0, i=1, \dots, q; h_j(\mathbf{x}) = 0, j=1, \dots, p\}. \quad (2.6)$$

Отметим, что когда вместо определения экстремума мы ограничиваемся рассмотрением минимизации, задача не теряет общности. Допустим, например, что нужно *максимизировать функцию* $f_1(\mathbf{x})$. Это эквивалентно *минимизации* функции $f(\mathbf{x}) = -f_1(\mathbf{x})$.

Не ограничивается общность и при использовании ограничений в форме неравенств, определяемых формулой (2.1). Допустим, например, что имеются ограничения

$$g_{1i}(\mathbf{x}) \leq 0.$$

Определив новое множество функций

$$g_i = -g_{1i},$$

вновь получим ограничения в форме (2.1).

2.3. Пример задачи математического программирования

Для иллюстрации задачи математического программирования предлагается двумерная задача с четырьмя ограничениями в форме неравенств и одним ограничением в форме равенства. Задача следующая:

$$\text{максимизировать } f(x_1, x_2) = x_1 x_2 \quad (2.7)$$

$$\text{при условии } 0,8 - x_1 \geq 0, \quad (2.8)$$

$$0,8 - x_2 \geq 0, \quad (2.9)$$

$$x_1 \geq 0, \quad (2.10)$$

$$x_2 \geq 0, \quad (2.11)$$

$$x_1^2 + x_2^2 - 1 = 0. \quad (2.12)$$

Лучше всего можно пояснить задачу с помощью графического представления на плоскости x_1 — x_2 (рис. 2.1).

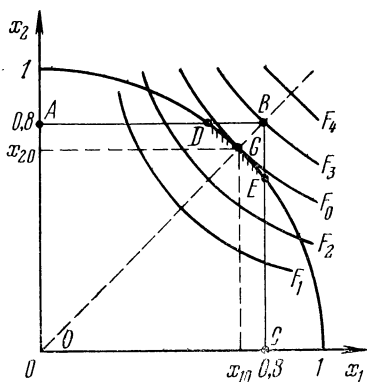


Рис. 2.1. Пример задачи математического программирования.

Формулы (2.10) и (2.11) ограничивают допустимую область первым квадрантом. Ограничения (2.8) и (2.9) устанавливают, что эта область лежит внутри квадрата $ABCO$.

Ограничение в форме равенства (2.12) устанавливает, что допустимая область должна лежать на единичной окружности с центром в начале координат. Таким образом, фактическая допустимая область задачи состоит

из замкнутой дуги DE единичной окружности, уравнение которой определено формулой (2.12).

Каждая из кривых F_1, \dots, F_4 представляет собой геометрическое место точек постоянного значения функции $f(x_1, x_2)$; при этом $F_1 < F_2 < \dots < F_4$. Легко видеть, что эти кривые представляют собой гиперболы, симметричные относительно прямой OB и описываемые уравнением

$$x_2 = \frac{F_i}{x_1}.$$

Обозначим через F_0 ту из этих кривых, которая в точке G касается дуги DE (G является также точкой пересечения DE и OB). Тогда в точке $G(x_{10}, x_{20})$ функция $f(x_1, x_2)$ достигает своего максимального значения $f(x_{10}, x_{20}) = F_0$ и при этом выполняются все ограничения, определяемые формулами от (2.8) до (2.12).

Таким образом, оптимальным решением рассматриваемой задачи является

$$x_{10} = x_{20} = \frac{1}{\sqrt{2}} \approx 0,707.$$

Оптимальное значение целевой функции равно

$$F_0 = x_{10}x_{20} = 0,5.$$

2.4. Классификация задач математического программирования

Рассмотрим задачу математического программирования, сформулированную в виде (2.6). Если $f(\mathbf{x})$, $g_i(\mathbf{x})$ ($i=1, \dots, q$) и $h_j(\mathbf{x})$ ($j=1, \dots, p$) являются линейными функциями x_i ($i=1, \dots, n$), то ее называют задачей линейного программирования [7—11, 28*—30*]. Следует подчеркнуть, что в задаче линейного программирования не только все ограничения, но и целевая функция являются линейными функциями x_i .

Целевая функция в задаче линейного программирования может быть представлена в виде

$$f(\mathbf{x}) = \sum_{i=1}^n c_i x_i = \mathbf{c}^T \mathbf{x}, \quad (2.13)$$

где \mathbf{c}^T — транспонированный n -мерный постоянный вектор \mathbf{c} . В общем случае, ограничения можно записать в

следующей форме:

$$Ax \begin{cases} \geq \\ = \\ \leq \end{cases} \mathbf{b}, \quad (2.14)$$

где \mathbf{b} — m -мерный постоянный вектор, m — общее число ограничений и A — постоянная матрица с размерностью $m \times n$. Более подробно формулировка ограничений в задаче линейного программирования будет рассмотрена в гл. 3. В ограничения обычно входят условия, определяющие знак некоторых переменных, как, например,

$$x_i \geq 0, \quad i = 1, \dots, n.$$

Эти условия не включаются в матрицу A и записываются отдельно.

В качестве примера рассмотрим следующую задачу линейного программирования (рис. 2.2):

$$\text{максимизировать } (x_1 + 4x_2) \quad (2.15)$$

$$\text{при условии } \left. \begin{array}{l} x_1 \leq 10, \\ x_2 \leq 10, \end{array} \right\} \quad (2.16)$$

$$\left. \begin{array}{l} x_1 + x_2 \leq 14, \\ x_1, x_2 \geq 0. \end{array} \right\} \quad (2.17)$$

Иначе можно записать

$$\max \{ \mathbf{c}^T \mathbf{x} \mid A\mathbf{x} \leq \mathbf{b}; x_1, x_2 \geq 0 \}, \quad (2.18)$$

где

$$\mathbf{c}^T = [1 \ 4], \quad A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 10 \\ 10 \\ 14 \end{bmatrix}.$$

Из рассмотрения ограничений (2.16) и (2.17) видно, что допустимая область, как это показано на рис. 2.2, лежит внутри многоугольника $ABCD$. Прямые L_1 , L_2 и L_3 соответствуют постоянным значениям целевой функции (2.15) в порядке их возрастания. Очевидно, что прямая L_2 , для которой $\mathbf{c}^T \mathbf{x} = 44$, соответствует максимальному значению целевой функции, при котором эта прямая имеет по крайней мере одну точку (B) общую с допустимой областью. Иначе говоря, точка B есть оптимальная точка. Следовательно, оптимальным

решением для этой задачи будет

$$x_{10} = 4$$

и

$$x_{20} = 10.$$

Численное решение задач линейного программирования детально будет рассмотрено в гл. 3.

Если целевая функция задачи математического программирования *квадратичная*, в то время как *все* ограничения линейны, мы имеем так называемую задачу *квадратичного программирования* [1—6, 12]. В общем случае целевая функция задачи квадратичного программирования может быть представлена следующим образом:

$$f(x) = c^T x + x^T D x, \quad (2.19)$$

где D — постоянная симметричная матрица с размерностью $n \times n$. Линейные ограничения задачи квадратичного программирования можно записать так же, как и для линейного программирования. Целевую функцию можно представить и в скалярной форме

$$f(x_1, \dots, x_n) = \sum_{i=1}^n c_i x_i + \sum_{i,j=1}^n x_i d_{ij} x_j, \quad (2.20)$$

где d_{ij} обозначают элементы матрицы D .

Так как $x_i x_j = x_j x_i$, то из (2.20) следует, что не ограничивая общности можно считать D симметричной матрицей.

Приведенный в 2.3 пример может служить иллюстрацией задачи квадратичного программирования, если исключить ограничение в форме равенства (2.12). Однако теперь, допустимая область ограничена квадратом $ABCO$

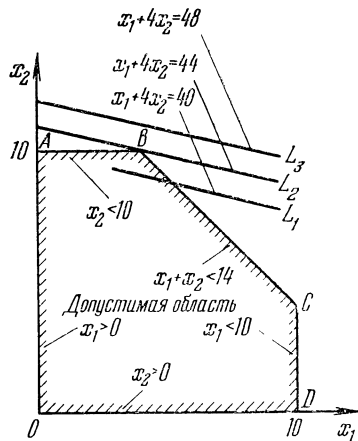


Рис. 2.2. Пример задачи линейного программирования.

(рис. 2.1) и точка B является оптимальной ($x_{10}=x_{20}=0,8$). Оптимальное значение целевой функции равно

$$f(x_{10}, x_{20}) = x_{10} x_{20} = (0,8)^2 = 0,64.$$

Некоторые численные методы решения задач квадратичного программирования будут рассмотрены в гл. 3.

Наиболее общим случаем задачи математического программирования является задача *нелинейного программирования* [1—6, 21*—27*]. К этой категории относится задача, если она имеет по крайней мере одно нелинейное ограничение или нелинейную целевую функцию. Естественно, что квадратичное программирование является частным случаем нелинейного программирования. Однако, вследствие линейности всех ее ограничений, ее относят к отдельному классу. Причина этого станет ясной при рассмотрении вычислительных методов в гл. 3. Приведенный в 2.3 пример является примером задачи нелинейного программирования.

Отдельный класс задач нелинейного программирования образует так называемая задача *геометрического программирования* [13, 14]. Общая форма целевой функции и ограничивающих неравенств имеет следующий вид:

$$\sum_{i=1}^m c_i \prod_{j=1}^n x_j^{a_{ij}},$$

где c_i — постоянные положительные коэффициенты, $i=1, \dots, m$, a_{ij} — произвольные вещественные числа, $j=1, \dots, n$.

Основная особенность геометрического программирования заключается в том, что оптимальная цена, связанная с каждым членом $c_i \prod_j x_j^{a_{ij}}$ целевой функции, может быть установлена отдельно, до отыскания оптимальных значений переменных x_j ($j=1, \dots, n$) [13, 14].

В некоторых приложениях линейного программирования переменные принимают лишь дискретные целочисленные значения. Этот вид задачи называется *целочисленным линейным программированием* [2, гл. 8; 15, 31*].

Большая часть работ, посвященных целочисленному программированию, ограничивается рассмотрением линейных задач. Поэтому некоторые авторы опускают слово линейное и говорят просто целочисленное программирование [15]. В ряде задач лишь некоторые из пере-

менных могут принимать только целочисленные значения, а остальные могут изменяться непрерывно. Задачи такого типа называют задачами *смешанного целочисленного программирования* [16].

Существуют задачи, параметры которых, образующие коэффициенты в ограничениях или в целевой функции, являются случайными переменными. Их называют задачами *стохастического программирования* [1, гл. X; 2, гл. 5; 7, гл. 25; 30*, гл. 6].

2.5. Выпуклость

Выпуклость [17, 32*] является одним из наиболее важных математических понятий, используемых в математическом программировании. Прежде чем перейти к изложению некоторых основных методов математического программирования, необходимо рассмотреть вопрос о выпуклости и привести несколько основных определений.

Как указывалось в предыдущих параграфах, каждая задача математического программирования обладает своей допустимой областью в пространстве состояний, которая состоит из всех точек, удовлетворяющих всем ограничениям задачи. Фактически она является множеством точек, представляющих допустимые решения задачи. Для задач математического программирования особый интерес представляют выпуклые множества.

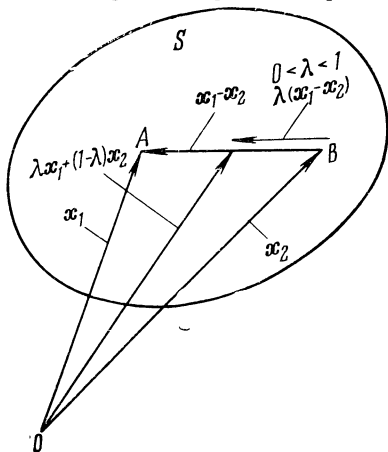


Рис. 2.3. Определение выпуклости.

Определение выпуклости. Множество называется *выпуклым*, если для любых двух точек, принадлежащих этому множеству, оно содержит соединяющий их отрезок.

Рассмотрим множество точек S (рис. 2.3) и две принадлежащие ему точки A и B , т. е. $A \in S$ и $B \in S$.

Точки A и B представляют собой соответственно концы векторов x_1 и x_2 . Для любого скаляра λ такого, что $0 \leq \lambda \leq 1$, вектор $\lambda(x_1 - x_2)$, как показано на рис. 2.3, будет коллинеарен с вектором $x_1 - x_2$. Очевидно, что точка P на отрезке AB определяется вектором $\lambda x_1 + (1 - \lambda)x_2$.

Теперь можно следующим образом определить выпуклое множество: точечное множество S называется *выпуклым*, если для любых двух точек $x_1, x_2 \in S$ все точки

$$[\lambda x_1 + (1 - \lambda)x_2] \in S, \quad 0 \leq \lambda \leq 1.$$

На рис. 2.4, а представлено несколько примеров выпуклых множеств, а на рис. 2.4, б несколько невыпуклых множеств.

Аналогично можно определить понятие *выпуклой функции*. Функция $f(x)$ является *выпуклой* на выпуклом

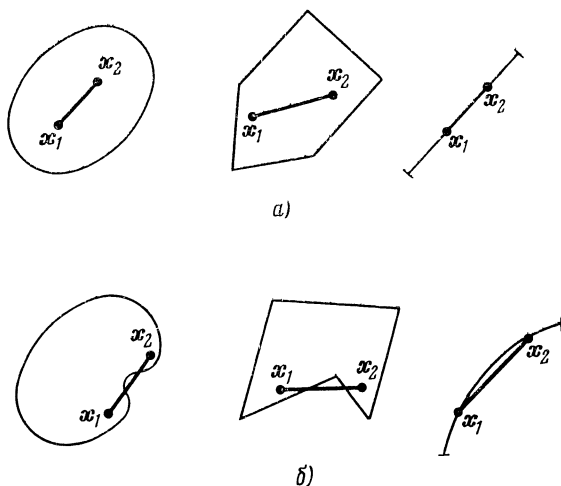


Рис. 2.4. Примеры выпуклых и невыпуклых множеств: а) выпуклые множества, б) невыпуклые множества.

множестве S , если для любых двух точек $x_1, x_2 \in S$ и для всех скаляров λ , $0 \leq \lambda \leq 1$ (рис. 2.5),

$$f[\lambda x_1 + (1 - \lambda)x_2] \leq \lambda f(x_1) + (1 - \lambda)f(x_2). \quad (2.21)$$

Если в (2.21) будет строгое неравенство (т. е. $<$ при $0 < \lambda < 1$), то функция $f(x)$ строго выпуклая. Если знак

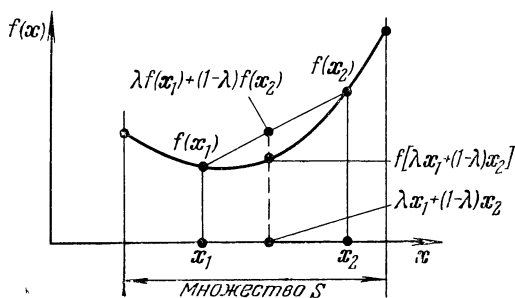
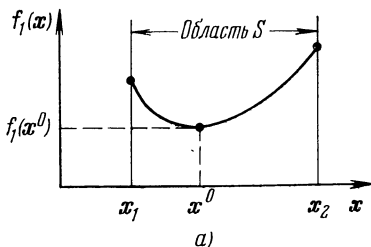
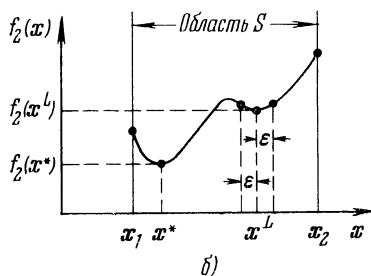


Рис. 2.5. Выпуклая функция.

неравенства изменить на обратный, то $f(x)$ есть *вогнутая функция*. Если при этом неравенство строгое (т. е. $>$), то $f(x)$ будет строго вогнутой. Из определений следует, что если $f(x)$ выпуклая (или строго выпуклая), то $-f(x)$ вогнутая (или строго вогнутая) и наоборот.



Значение понятия выпуклости для задач математического программирования станет более ясным в дальнейшем, особенно послечтения гл. 3. Одно из наиболее важных свойств выпуклых функций видно из следующего рассмотрения.



Рассмотрим функции $f_1(x)$ и $f_2(x)$, представленные соответственно на рис. 2.6, а и б. Функция $f_1(x)$ — выпуклая, а $f_2(x)$ — невыпуклая. Рассмотрим обе функции в области $\{S \mid x_1 \leq x \leq x_2\}$.

Рис. 2.6. Минимум выпуклой и невыпуклой функций: а) выпуклая $f_1(x)$, б) невыпуклая $f_2(x)$.

Рассмотрим обе функции в области $\{S \mid x_1 \leq x \leq x_2\}$.

Функция $f_1(x)$ имеет минимум $f_1(x^0)$ при $x^0 \in S$, так что

$$f_1(x^0) \leq f_1(x) \quad \text{для всех } x \in S.$$

Другими словами, выпуклая функция $f_1(x)$ имеет *глобальный* минимум в области S .

Аналогично, невыпуклая функция $f_2(x)$ обладает глобальным минимумом в точке $x^* \in S$, так что

$$f_2(x^*) \leq f_2(x) \quad \text{для всех } x \in S.$$

С другой стороны, в окрестности точки $x^L \in S$ удовлетворяется следующее условие:

$$f_2(x^L) \leq f_2(x^L \pm \varepsilon), \quad \varepsilon > 0, \quad (x^L \pm \varepsilon) \in S,$$

где ε — малое положительное число. Точка x^L называется *локальным минимумом*. Выпуклая функция имеет только глобальный минимум, в то время как невыпуклая функция может обладать любым числом локальных минимумов. Это свойство имеет особое значение для большинства алгоритмов численных решений задач минимизации.

2.6. Теорема Куна—Таккера

Известная теорема Куна—Таккера [18] является одной из наиболее важных теорем математического программирования. Она служит основой многих вычислительных алгоритмов. Исчерпывающее изложение теоремы Куна—Таккера можно найти, например, в [1, гл. 11; 2, гл. 6; и 3, гл. 3].

Допустим, что задача математического программирования сформулирована следующим образом:

$$\left. \begin{array}{l} \text{минимизировать } f(x) \\ \text{при условии } g_j(x) \leq 0, \quad j=1, \dots, m, \\ x_i \geq 0, \quad i=1, \dots, n, \end{array} \right\} \quad (2.22)$$

где $f(x)$ и $g_j(x)$ ($j=1, \dots, m$) — выпуклые функции n переменных $x^T = [x_1, x_2, \dots, x_n]$.

Введем совокупность из m переменных u_1, \dots, u_m , обычно называемых *множителями Лагранжа*, которые образуют вектор u . Новая функция — *функция Лагран-*

жа $L(x, u)$ — определяется следующим образом:

$$L(x, u) = f(x) + \sum_{j=1}^m u_j g_j(x). \quad (2.23)$$

Теперь можно сформулировать теорему Куна — Таккера.

Теорема Куна — Таккера. Пусть выполнено условие Слейтера: найдется точка x^* такая, что $x_i^* \geq 0$, $i=1, \dots, n$; $g_j(x^*) < 0$, $j=1, \dots, m$ (если ограничения $g_j(x)$ линейны, то это условие может быть отброшено). Вектор x^0 является решением задачи оптимизации (2.22) тогда и только тогда, когда существует такой вектор u^0 , что при всех $x_i \geq 0$ и $u_j \geq 0$

$$\left. \begin{aligned} x_i^0 &\geq 0, \quad i = 1, \dots, n, \\ u_j^0 &\geq 0, \quad j = 1, \dots, m, \\ L(x^0, u) &\leq L(x^0, u^0) \leq L(x, u^0). \end{aligned} \right\} \quad (2.24)$$

Другими словами, оптимальная точка (x^0, u^0) обладает следующим свойством: при фиксированном u^0 $L(x, u^0)$ имеет глобальный минимум в области $x_i \geq 0$ ($i=1, \dots, n$) при $x=x^0$. При фиксированном x^0 она имеет глобальный максимум в области $u_j \geq 0$ ($j=1, \dots, m$) при $u=u^0$. Экстремальная точка (x^0, u^0) с этими свойствами называется *седловой точкой*. Поэтому теорему Куна — Таккера часто называют *теоремой о седловой точке*.

Можно привести следующие эвристические соображения, подтверждающие теорему. Подставив (2.23) в (2.24), получим для всех $x_i \geq 0$ ($i=1, \dots, n$) и $u_j \geq 0$ ($j=1, \dots, m$):

$$\begin{aligned} f(x^0) + \sum_{j=1}^m u_j g_j(x^0) &\leq f(x^0) + \sum_{j=1}^m u_j^0 g_j(x^0) \leq f(x) + \\ &+ \sum_{j=1}^m u_j^0 g_j(x). \end{aligned} \quad (2.25)$$

Из левого неравенства формулы (2.25) следует, что

$$\sum_{j=1}^m u_j g_j(x^0) \leq \sum_{j=1}^m u_j^0 g_j(x^0). \quad (2.26)$$

Так как (2.26) справедливо при всех $u_j \geq 0$, то

$$g_j(\mathbf{x}^0) \leq 0 \quad (2.27)$$

и

$$\sum_{j=1}^m u_j^0 g_j(\mathbf{x}^0) = 0. \quad (2.28)$$

Из неравенства (2.27) следует, что оптимальная точка \mathbf{x}^0 является допустимой точкой. Учитывая (2.28), правое неравенство формулы (2.25) можно записать следующим образом:

$$f(\mathbf{x}^0) \leq f(\mathbf{x}) + \sum_{j=1}^m u_j^0 g_j(\mathbf{x}) \quad (2.29)$$

при всех $x_i \geq 0$.

Так как в соответствии с требованиями допустимости $u_j^0 \geq 0$ ($j=1, \dots, m$) и $g_j(\mathbf{x}) \leq 0$, то из (2.29) следует, что

$$f(\mathbf{x}^0) \leq f(\mathbf{x}) \quad (2.30)$$

при всех $x_i \geq 0$ ($i=1, \dots, n$) и $g_j(\mathbf{x}) \leq 0$ ($j=1, \dots, m$). Это значит, что $f(\mathbf{x}^0)$ действительно является минимумом $f(\mathbf{x})$ в допустимой области и \mathbf{x}^0 есть решение задачи (2.22). Строгое доказательство теоремы Куна — Таккера можно найти в оригинальной работе [18] и в других источниках, ссылки на которые были сделаны в начале этого параграфа.

Условия Куна — Таккера для случая дифференцируемых функций можно выразить также в следующей форме [2, гл. 6; 3, гл. 3]:

$$\left. \begin{aligned} \frac{\partial L}{\partial x_i} \Big|_{\mathbf{x}^0, \mathbf{u}^0} &\geq 0, \\ x_i^0 \frac{\partial L}{\partial x_i} \Big|_{\mathbf{x}^0, \mathbf{u}^0} &= 0, \\ x_i^0 &\geq 0, \quad i = 1, \dots, n, \end{aligned} \right\} \quad (2.31)$$

$$\left. \begin{aligned} \frac{\partial L}{\partial u_j} \Big|_{\mathbf{x}^0, \mathbf{u}^0} &\leq 0, \\ u_j^0 \frac{\partial L}{\partial u_j} \Big|_{\mathbf{x}^0, \mathbf{u}^0} &= 0, \\ u_j^0 &\geq 0, \quad j = 1, \dots, m. \end{aligned} \right\} \quad (2.32)$$

2.7. Двойственность

Понятие двойственности играет важную роль в математическом программировании.

Рассмотрим несколько иную формулировку задачи минимизации:

$$(P) = \min \{f(\mathbf{x}) \mid g_j(\mathbf{x}) \leq 0, j=1, \dots, m\}. \quad (2.33)$$

Эту задачу называют *прямой* (P) [19, 20]. Существует связанная с этой задачей задача максимизации, называемая *двойственной* (D) или сопряженной с прямой задачей. Двойственная задача формируется следующим образом:

$$(D) = \max \left\{ L(\mathbf{x}, \mathbf{u}) \mid \frac{\partial L(\mathbf{x}, \mathbf{u})}{\partial x_i} = 0, i = 1, \dots, n; \right. \\ \left. u_j \geq 0, j = 1, \dots, m \right\}, \quad (2.34)$$

где $L(\mathbf{x}, \mathbf{u})$ — функция Лагранжа, определенная формулой (2.23). Понятие двойственности устанавливает определенные соотношения между решениями прямой задачи и двойственной задачи. Теорема 2.1 определяет связь между целевыми функциями $f(\mathbf{x})$ и $L(\mathbf{x}, \mathbf{u})$ двух задач.

Теорема 2.1. Если $f(\mathbf{x})$ и $g_j(\mathbf{x})$ ($j=1, \dots, m$) являются выпуклыми дифференцируемыми функциями, \mathbf{y} есть любая допустимая точка прямой задачи (2.33) и если $(\mathbf{x}^f, \mathbf{u}^f)$ обозначает любую допустимую точку двойственной задачи (2.34), то

$$f(\mathbf{y}) \geq L(\mathbf{x}^f, \mathbf{u}^f). \quad (2.35)$$

Доказательство. Так как \mathbf{y} — допустимая точка, то

$$g_j(\mathbf{y}) \leq 0, \quad j=1, \dots, m,$$

и поэтому

$$f(\mathbf{y}) \geq f(\mathbf{y}) + \sum_{j=1}^m u_j^f g_j(\mathbf{y}). \quad (2.36)$$

Так как f и все g_j выпуклые, то можно записать следующие выражения, являющиеся, в сущности,

определениями выпуклой дифференцируемой функции:

$$f(\mathbf{y}) \geq f(\mathbf{x}^f) + (\mathbf{y} - \mathbf{x}^f)^T \nabla f(\mathbf{x}^f), \quad (2.37)$$

$$g_j(\mathbf{y}) \geq g_j(\mathbf{x}^f) + (\mathbf{y} - \mathbf{x}^f)^T \nabla g_j(\mathbf{x}^f), \quad j = 1, \dots, m. \quad (2.38)$$

Подставляя (2.37) и (2.38) в (2.36), получим

$$f(\mathbf{y}) + \sum_{j=1}^m u_j^f g_j(\mathbf{y}) \geq f(\mathbf{x}^f) + \sum_{j=1}^m u_j^f g_j(\mathbf{x}^f) + (\mathbf{y} - \mathbf{x}^f)^T \left[\nabla f(\mathbf{x}^f) + \sum_{j=1}^m u_j^f \nabla g_j(\mathbf{x}^f) \right]. \quad (2.39)$$

Так как точка $(\mathbf{x}^f, \mathbf{u}^f)$ есть допустимая точка двойственной задачи (D) , то

$$\nabla_x L(\mathbf{x}^f, \mathbf{u}^f) = \nabla f(\mathbf{x}^f) + \sum_{j=1}^m u_j^f \nabla g_j(\mathbf{x}^f) = 0. \quad (2.40)$$

Следует также заметить, что согласно определению

$$f(\mathbf{x}^f) + \sum_{j=1}^m u_j^f g_j(\mathbf{x}^f) = L(\mathbf{x}^f, \mathbf{u}^f). \quad (2.41)$$

Рассматривая (2.41) с учетом (2.36) и (2.39), имеем

$$f(\mathbf{y}) \geq L(\mathbf{x}^f, \mathbf{u}^f),$$

что и требовалось доказать.

Практическое значение теоремы 2.1 состоит в том, что если алгоритмы, решающие прямую задачу (P) , выработывают двойственные допустимые точки, то это позволяет получить оценку снизу для оптимального значения $f(\mathbf{y})$.

Прежде чем перейти к теореме 2.2, необходимо определить понятие условия регулярности первого порядка Куна — Таккера [18].

Определение. Пусть \mathbf{x}^f является допустимой точкой основной задачи (P) , а функции $g_j(\mathbf{x})_j (j=1, \dots, m)$ дифференцируемы. Тогда ограничения в точке \mathbf{x}^f удовлетворяют *условию регулярности первого порядка*, если всякий ненулевой вектор \mathbf{z} , для которого

$$\mathbf{z}^T \nabla g_j(\mathbf{x}^f) \leq 0 \quad \text{для всех } j \in \{j | g_j(\mathbf{x}^f) = 0\}, \quad (2.42)$$

касателен к некоторой гладкой дуге, выходящей из \mathbf{x}^f и принадлежащей допустимой области. Это условие

выполняется, если выполнено упомянутое выше условие Слейтера.

Теорема 2.2. *Если ограничения удовлетворяют условиям регулярности первого порядка в точке x^0 , которая является решением основной задачи (P), то существует решение двойственной задачи (D), а максимальное значение функции L равно минимальному значению функции f для допустимого x основной задачи (полное доказательство этой теоремы дано в [19]).*

Теорема 2.2 имеет большое практическое значение. Возможны случаи, когда численное решение задачи математического программирования связано с большими трудностями. В то же время, двойственная ей задача может быть легко решена. Теорема 2.2 утверждает, что при определенных условиях оптимальные значения целевых функций прямой и двойственной задач совпадают. Таким образом, может оказаться более удобным решить вначале двойственную задачу и, используя полученную информацию, решить прямую задачу.

В качестве частного случая сформулируем понятие двойственности для задачи линейного программирования [8, 28*, 29*]. Рассмотрим следующую прямую задачу линейного программирования:

$$\max \{c^T x \mid Ax \leq b; x_i \geq 0, i=1, \dots, n\}, \quad (2.43)$$

где все переменные те же, что и в формулах (2.13) и (2.14).

Задача, двойственная к этой, может быть сформулирована следующим образом:

$$\min \{b^T u \mid A^T u \geq c; u_j \geq 0, j=1, \dots, m\}. \quad (2.44)$$

Большое значение имеют следующие теоремы:

Теорема 2.3. *Если x — любое допустимое решение прямой задачи (2.43) и u — любое допустимое решение двойственной задачи (2.44), то*

$$c^T x \leq b^T u. \quad (2.45)$$

Доказательство. Так как x допустимо в прямой задаче

$$(Ax)_i \leq b_i, \quad i=1, \dots, m, \quad (2.46)$$

и так как $u_j \geq 0 (j=1, \dots, m)$ для любого допустимого

решения двойственной задачи, то

$$u_i (Ax)_i \leq u_i b_i. \quad (2.47)$$

Суммируя (2.47) по i , запишем

$$\sum_{i=1}^m u_i (Ax)_i \leq \sum_{i=1}^m u_i b_i \quad (2.48)$$

или

$$\mathbf{u}^T A \mathbf{x} \leq \mathbf{b}^T \mathbf{u}. \quad (2.49)$$

Аналогично, так как $x_i \geq 0$ ($i=1, \dots, n$), то из (2.44) следует, что

$$\mathbf{u}^T A \mathbf{x} \geq \mathbf{c}^T \mathbf{x}. \quad (2.50)$$

Сравнивая (2.49) и (2.50), имеем

$$\mathbf{c}^T \mathbf{x} \leq \mathbf{b}^T \mathbf{u},$$

что и требовалось доказать.

Теорема 2.4. Если \mathbf{x}^f есть допустимое решение прямой задачи и \mathbf{u}^f — допустимое решение двойственной задачи такое, что

$$\mathbf{c}^T \mathbf{x}^f = \mathbf{b}^T \mathbf{u}^f, \quad (2.51)$$

то \mathbf{x}^f есть оптимальное решение прямой задачи (2.43) и \mathbf{u}^f есть оптимальное решение двойственной задачи (2.44).

Доказательство. Из (2.45) и (2.51) следует, что для всякого допустимого \mathbf{x}

$$\mathbf{c}^T \mathbf{x} \leq \mathbf{b}^T \mathbf{u}^f = \mathbf{c}^T \mathbf{x}^f \quad (2.52)$$

и так как прямая задача — это задача максимизации, то \mathbf{x}^f есть оптимальное решение. Аналогично, для любого допустимого \mathbf{u}

$$\mathbf{b}^T \mathbf{u}^f = \mathbf{c}^T \mathbf{x}^f \leq \mathbf{b}^T \mathbf{u} \quad (2.53)$$

и так как двойственная задача — это задача минимизации, то \mathbf{u}^f есть ее оптимальное решение.

Теорема 2.5. Если одна из пары задач (2.43), (2.44) имеет оптимальное решение, то и другая имеет оптимальное решение.

Полное доказательство теоремы 2.5 см в [8, стр. 229—230]. Очевидно, что теоремы 2.3 и 2.4 являются частными случаями более общих теорем 2.1 и 2.2.

Литература

1. J. Abadie, ed., *Nonlinear Programming*, Wiley, New York, 1967.
2. G. Hadley, *Nonlinear and Dynamic Programming*, Addison-Wesley, Reading, Mass., 1964. [Русский перевод: Дж. Хедли, *Нелинейное и динамическое программирование*, «Мир», 1967.]
3. Г. П. Кюнци, В. Крелле, *Нелинейное программирование*, перев. с нем. под ред. Г. А. Соколова, «Советское Радио», 1965.
4. S. Vajda, *Mathematical Programming*, Addison-Wesley, Reading, Mass., 1961.
5. R. L. Graves, P. Wolfe, eds., *Recent Advances in Mathematical Programming*, McGraw-Hill, New York, 1963.
6. G. Zoutendijk, *Methods of Feasible Directions*, American Elsevier, New York, 1960. [Русский перевод: Г. Зойтендейк, *Методы возможных направлений*, ИЛ, 1963.]
7. G. B. Dantzig, *Linear Programming and Extensions*, Princeton University Press, Princeton, N. J., 1963. [Русский перевод: Дж. Данциг, *Линейное программирование, его применения и обобщения*, «Прогресс», 1966.]
8. G. Hadley, *Linear Programming*, Addison-Wesley, Reading, Mass., 1962.
9. M. Simonard, *Linear Programming*, Prentice-Hall, Englewood Cliffs, N. J., 1966.
10. S. I. Gass, *Linear Programming*, 3rd ed., McGraw-Hill, New York, 1969. [Русский перевод: С. Гасс, *Линейное программирование*, Физматгиз, 1961.]
11. R. W. Llewellyn, *Linear Programming*, Holt, Rinehart and Winston, New York, 1964.
12. J. C. G. Boot, *Quadratic Programming*, North-Holland, Amsterdam, 1964.
13. R. J. Duffin, E. L. Peterson and C. M. Zener, *Geometric Programming*, Wiley, New York, 1967. [Русский перевод: Р. Даффин, Э. Питерсон, К. Зенер, *Геометрическое программирование*, «Мир», 1972.]
14. D. J. Wilde and C. S. Beightler, *Foundation of Optimization*, Prentice-Hall, Englewood Cliffs, N. J., 1967.
15. M. L. Balinski, *Integer Programming: Methods, Uses, Computation*, *Management Sci.* 12, pp. 253—313 (1965).
16. J. F. Benders, *Partitioning Procedures for Solving Mixed-Variables Programming Problems*, *Numerische Math.* 4, pp. 238—252 (1962).
17. F. A. Valentine, *Convex Sets*, McGraw-Hill, New York, 1964.
18. H. W. Kuhn, A. W. Tucker, *Nonlinear Programming*, *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, Berkeley, 1951, pp. 481—492.
19. A. V. Fiacco, G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, Wiley, New York, 1968. [Русский перевод: А. Фиакко и Г. Мак-Кормик, *Нелинейное программирование. Методы последовательной безусловной минимизации*, «Мир», 1972.]

20. P. Wolfe, A Quality Theorem for Nonlinear Programming, *Quart. Appl. Math.* **19**, pp. 239—244 (1961).
- 21*. К. Дж. Эрроу, Л. Гурвиц, Х. Удзава, Исследования по линейному и нелинейному программированию, ИЛ, 1962.
- 22*. С. И. Зуховицкий, Л. И. Авдеева, Линейное и выпуклое программирование, «Наука», 1967.
- 23*. Дж. Б. Деннис, Математическое программирование и электрические цепи, ИЛ, 1961.
- 24*. С. Карлин, Математические методы в теории игр, программировании и экономике, «Мир», 1964.
- 25*. В. Ф. Демьянов, А. М. Рубинов, Приближенные методы решения экстремальных задач, ЛГУ, 1968.
- 26*. Ж. Сеа, Оптимизация. Теория и алгоритмы, «Мир», 1973.
- 27*. У. И. Зангвилл, Нелинейное программирование, «Советское радио», 1973.
- 28*. Д. Б. Юдин, Е. Г. Гольштейн, Задачи и методы линейного программирования, «Советское радио», 1964.
- 29*. Д. Б. Юдин, Е. Г. Гольштейн, Линейное программирование, Физматгиз, 1963.
- 30*. Е. Г. Гольштейн, Д. Б. Юдин, Новые направления в линейном программировании, «Советское радио», 1966.
- 31*. А. А. Корбут, Ю. Ю. Финкельштейн, Дискретное программирование, «Наука», 1971.
- 32*. Р. Рокфеллар, Выпуклый анализ, «Мир», 1973.

ЧИСЛЕННОЕ РЕШЕНИЕ ЗАДАЧ МАТЕМАТИЧЕСКОГО ПРОГРАММИРОВАНИЯ

3.1. Линейное программирование

Методы линейного программирования [1—5, см. также 28*—30* в гл. 2] нашли применение в ряде задач оптимального управления и, в частности, в задачах, относящихся к дискретным системам. В настоящем параграфе мы переформулируем задачу линейного программирования, приведя ее к канонической форме. Такая формулировка задачи позволяет удобно применять алгоритм симплексного метода.

Новая формулировка задачи

Как было указано в гл. 2, задача линейного программирования может быть сформулирована следующим образом. Дана целевая функция

$$f(\mathbf{x}) = \sum_{i=1}^N c_i x_i = \mathbf{c}^T \mathbf{x}. \quad (3.1)$$

Найти значения переменных $x_i \geq 0$, $i=1, 2, \dots, N$, удовлетворяющие следующим ограничениям в форме линейных неравенств и равенств:

$$\sum_{r=1}^N a_{ir} x_r \leq b_i, \quad i = 1, \dots, p, \quad (3.2)$$

$$\sum_{r=1}^N a_{jr} x_r = b_j, \quad j = p + 1, \dots, p + q, \quad (3.3)$$

$$\sum_{r=1}^N a_{kr} x_r \geq b_k, \quad k = p + q + 1, \dots, m. \quad (3.4)$$

Таким образом, задано q ограничений в форме равенств и $m - q$ ограничений в форме неравенств.

Так как в общем случае легче оперировать с равенствами чем с неравенствами, то неравенства (3.2) и (3.4) преобразуем в равенства путем введения *дополнительных переменных*.

Для ограничений вида (3.2) дополнительная переменная $x_{si} \geq 0$ вводится следующим образом:

$$\sum_{r=1}^N a_{ir}x_r + x_{si} = b_i, \quad i = 1, \dots, p. \quad (3.5)$$

Для ограничений (3.4) дополнительную переменную $x_{sj} \geq 0$ можно ввести следующим образом:

$$\sum_{r=1}^N a_{jr}x_r - x_{sj} = b_j, \quad j = p + q + 1, \dots, m. \quad (3.6)$$

Следует заметить, что в целевую функцию дополнительные переменные не входят и, как будет видно из дальнейшего, они не влияют на оптимальное решение. Таким образом, дополнительные переменные используются, в сущности, для того чтобы привести ограничения к следующей общей форме:

$$Ax = b, \quad (3.7)$$

где $x \geq 0$ теперь представляет $n = (N + m - q)$ -мерный вектор, включающий в себя все основные и дополнительные переменные, b — m -мерный постоянный вектор и A — постоянная матрица размерности $m \times n$. Таким образом, (3.7) представляет собой m линейных уравнений с n неизвестными. Предполагается, что $n > m$, т. е. что общее число неизвестных переменных больше числа ограничений в форме равенств. Если $m > n$, то уравнений больше чем неизвестных, и если решение существует, то некоторые уравнения являются зависимыми, а поэтому могут быть исключены. Если $m = n$, то мы имеем обычный случай системы линейных алгебраических уравнений, которая имеет единственное решение, если A — неособенная матрица.

Задача линейного программирования может быть теперь сформулирована в следующей канонической форме:

$$\max (\text{или } \min) \{c^T x \mid Ax = b; x_i \geq 0, i = 1, 2, \dots, n\}, \quad (3.8)$$

где

$$c^T = [c_1, c_2, \dots, c_N, 0, \dots, 0]$$

есть n -мерный вектор-строка и N — число основных переменных.

Симплекс-алгоритм

Так называемый симплекс-алгоритм, предложенный Данцигом [1], является одним из наиболее эффективных алгоритмов численного решения задач линейного программирования. В настоящее время существует несколько вариантов симплексного метода. В данном параграфе, однако, будет изложена только основная идея этого метода. Более детально с симплекс-алгоритмом читатель может познакомиться по книгам, посвященным линейному программированию [1—5].

Перепишав ограничивающие равенства (3.7) в скалярной форме, получим следующую систему m линейных уравнений с n переменными:

$$\left. \begin{aligned} a_{11}x_1 + \dots + a_{1n}x_n &= b_1, \\ \dots \dots \dots \dots \dots \dots \\ a_{m1}x_1 + \dots + a_{mn}x_n &= b_m. \end{aligned} \right\} \quad (3.9)$$

Предполагается, что $n > m$ и что коэффициенты a_{ij} ($i=1, \dots, m$; $j=1, \dots, n$) являются постоянными элементами матрицы A . Как было определено в гл. 2, любой вектор x , компоненты которого $x_j \geq 0$ ($j=1, \dots, n$) удовлетворяют всем равенствам (3.9), представляет собой *допустимое решение* задачи. В симплексном методе есть очень важная особая форма допустимого решения, а именно *базисное допустимое решение*.

Определение базисного решения. Пусть дана система m линейных уравнений вида (3.9) с n неизвестными и допустим, что матрица A , составленная из коэффициентов уравнений, имеет ранг m . *Базисом* называется набор из m переменных таких, что матрица, составленная из коэффициентов при этих переменных в m ограничениях, является неособенной. Эти m переменных называются *базисными переменными* (по отношению к данному базису), остальные называются *небазисными переменными*. Если положить все небазисные переменные равными нулю и решить ограничения относительно базисных переменных, то мы получим *базисное решение*, связанное с данным базисом.

Если базисное решение x является допустимым, т. е.

$$x_j \geq 0, \quad j=1, \dots, n, \quad (3.10)$$

то его называют *допустимым базисным решением*.

Рассмотрим теперь геометрическую структуру, определяемую формулой (3.9). Каждое из m уравнений определяет гиперплоскость в n -мерном пространстве. Очевидно, что гиперплоскость представляет собой выпуклое множество. Предположим, например, что точки \mathbf{x}_1 и \mathbf{x}_2 лежат на некоторой i -й гиперплоскости. Тогда

$$\mathbf{a}_i^T \mathbf{x}_1 = b_i \quad (3.11)$$

и

$$\mathbf{a}_i^T \mathbf{x}_2 = b_i. \quad (3.12)$$

Любая точка \mathbf{x} , расположенная на отрезке прямой, соединяющем две точки \mathbf{x}_1 и \mathbf{x}_2 , также лежит на той же гиперплоскости, так как

$$\mathbf{a}_i^T \mathbf{x} = \mathbf{a}_i^T [\lambda \mathbf{x}_2 + (1 - \lambda) \mathbf{x}_1] = \lambda b_i + (1 - \lambda) b_i = b_i. \quad (3.13)$$

Открытое и замкнутое полупространства, ограниченные гиперплоскостью, также являются выпуклыми множествами. Замкнутые полупространства, определяемые неравенствами $x_i \geq 0$, также являются выпуклыми. Опираясь на очевидный факт, что пересечение выпуклых множеств выпукло, мы можем тем самым утверждать, что множество допустимых решений в задаче (3.8) выпукло. Совершенно аналогично проверяется выпуклость допустимого множества и при другой форме задания ограничений (например, (3.2) — (3.4)).

Введем важное понятие крайних (или экстремальных) точек выпуклых множеств.

Определение крайней точки. Точка \mathbf{x} есть крайняя (экстремальная) точка выпуклого множества, если не существует принадлежащих данному множеству других точек \mathbf{x}_1 , \mathbf{x}_2 , $\mathbf{x}_1 \neq \mathbf{x}_2$, таких, что $\mathbf{x} = \lambda \mathbf{x}_2 + (1 - \lambda) \mathbf{x}_1$, $0 < \lambda < 1$.

Таким образом, крайняя точка не может лежать на отрезке, соединяющем любые другие две точки этого множества. Крайняя точка фактически является «угловой точкой» выпуклого множества. Так, в примере, представленном на рис. 2.2, крайними точками выпуклого множества (многоугольник $ABCO$) являются точки A , B , C , D и O .

Из рассмотрения рис. 2.2 следует, что если оптимальное решение единственно, то оно всегда есть край-

няя точка. Допустим, что L_1 соответствует допустимому решению. Можно всегда перемещать L_1 вдоль BC в области, пока не будет достигнута крайняя точка B . При перемещении L_1 за крайнюю точку решение уже не будет допустимым. Следовательно, B есть точка, в которой целевая функция достигает максимального значения, не нарушая какого-либо из ограничений.

Большое значение имеет следующая теорема 3.1.

Теорема 3.1. *Каждое допустимое базисное решение системы уравнений (3.9) есть крайняя (экстремальная) точка выпуклого множества (3.9), (3.10) и, обратно, каждая крайняя точка множества (3.9), (3.10) есть допустимое базисное решение [2, стр. 101—102].*

Можно сказать, что симплекс-алгоритм фактически вытекает из теоремы 3.1.

Симплекс-алгоритм определяет следующие шаги:

1. Найти любое допустимое базисное решение ограничивающих равенств (3.9).

2. Перейти от первого допустимого базисного решения к соседнему допустимому базисному решению в направлении, при котором значение целевой функции возрастает.

3. Продолжать переходить от одного допустимого базисного решения к другому, пока не будет достигнуто наибольшее значение целевой функции.

В системе из m уравнений с n неизвестными ($n > m$) общее число базисных решений равно числу сочетаний из m по n :

$$C_n^m = \frac{n!}{m!(n-m)!}. \quad (3.14)$$

Число допустимых базисных решений (иначе говоря, число крайних точек многогранника, выделяемого ограничениями) не превосходит этой величины. Как указывалось выше, крайние точки являются кандидатами в оптимальное решение задачи линейного программирования. Итерации симплекс-алгоритма состоят в переходе из одной крайней точки в другую в направлении возрастающего значения целевой функции. Легко видеть, что общее число возможных шагов ограничено сверху и, следовательно, оптимальное решение может быть достигнуто за конечное число шагов.

Симплекс-таблица

Рассмотрим теперь следующую задачу линейного программирования:

$$\max \left\{ f(\mathbf{x}) = \sum_{i=1}^N c_i x_i \mid \sum_{i=1}^N a_{ji} x_i \leq b_j; \right. \\ \left. b_j \geq 0, j = 1, \dots, m; \quad x_i \geq 0, i = 1, \dots, N \right\}. \quad (3.15)$$

Вводя m дополнительных переменных x_{N+1}, \dots, x_{N+m} , преобразуем ограничения в форме неравенств к виду

$$\sum_{i=1}^N a_{ji} x_i + x_{N+j} = b_j, \quad (3.16)$$

$x_{N+j} \geq 0, j = 1, \dots, m$. Или, в матричной форме,

$$A\mathbf{x} = \mathbf{b}, \quad (3.17)$$

где A — следующая матрица размерности $m \times n$ при $n = N + m$:

$$A = \left[\begin{array}{ccc|ccc} a_{11} & \dots & a_{1N} & 1 & 0 & \dots & 0 \\ & & & 0 & 1 & \dots & 0 \\ \dots & & \dots & \dots & \dots & \dots & \dots \\ a_{m1} & \dots & a_{mN} & 0 & 0 & \dots & 1 \end{array} \right] \quad (3.18)$$

$\left. \begin{array}{l} \text{---} \\ \text{---} \end{array} \right\} \begin{array}{l} N \text{ столбцов} \\ m \text{ столбцов} \end{array}$

и

$$\mathbf{x} = \left[\begin{array}{l} x_1 \\ x_2 \\ \vdots \\ x_N \\ \dots \\ x_{N+1} \\ \vdots \\ x_{N+m} \end{array} \right] \left. \begin{array}{l} \\ \\ \\ \end{array} \right\} \begin{array}{l} \text{основные переменные} \\ \\ \\ \end{array} \quad (3.19)$$

$$\left. \begin{array}{l} \\ \\ \\ \end{array} \right\} \begin{array}{l} \text{дополнительные переменные} \\ \\ \\ \end{array}$$

Рассмотрение (3.16) показывает, что решение

$$x_1 = x_2 = \dots = x_N = 0, \quad (3.20)$$

$$x_{N+j} = b_j, \quad j = 1, \dots, m, \quad (3.21)$$

является допустимым базисным решением. Поэтому с

него можно начать решение задачи линейного программирования. Согласно (3.16) можно записать для базисных переменных

$$x_{N+j} = b_j - \sum_{i=1}^N a_{ji}x_i \geq 0, \quad j = 1, \dots, m. \quad (3.22)$$

Вектор \mathbf{c} целевой функции дополняют нулевыми членами, соответствующими переменным x_{N+j} ($j=1, \dots, m$). Новый вектор \mathbf{c} будет равен

$$\mathbf{c}^T = [c_1, c_2, \dots, c_N, c_{N+1}, \dots, c_{N+m}], \quad (3.23)$$

где

$$c_{N+1} = c_{N+2} = \dots = c_{N+m} = 0. \quad (3.24)$$

Значение целевой функции при допустимом базисном решении (3.20), (3.21) равно

$$f_0(\mathbf{x}) = \sum_{j=1}^m c_{N+j}x_{N+j} = \sum_{j=1}^m c_{N+j}b_j = 0. \quad (3.25)$$

С учетом (3.24) и (3.25) можно записать целевую функцию в следующем виде:

$$f(\mathbf{x}) = \sum_{i=1}^N c_i x_i = f_0(\mathbf{x}) + \sum_{i=1}^N \left(c_i - \sum_{j=1}^m a_{ji}c_{N+j} \right) x_i, \quad (3.26)$$

или

$$f(\mathbf{x}) - f_0(\mathbf{x}) = \sum_{i=1}^N z_i x_i, \quad (3.27)$$

где

$$z_i = c_i - \sum_{j=1}^m a_{ji}c_{N+j} = c_i. \quad (3.28)$$

Заметим, что (3.27) представляет собой разность между целевой функцией и ее значением при допустимом базисном решении.

Табл. 3.1 представляет собой начальную таблицу симплексного метода. В таблице использовано обозначение

$$b_j = a_{j0}, \quad j = 1, \dots, m. \quad (3.29)$$

Следует напомнить, что в допустимом базисном решении значения N первых переменных x_1, \dots, x_N (небазисных) равны нулю, а базисных переменных x_{N+1}, \dots, x_{N+m} равны b_1, \dots, b_m .

Таблица 3.1

		Небазисные переменные					Базисные переменные			
		b	x_1	x_2	...	x_N	x_{N+1}	x_{N+2}	...	x_{N+m}
Базисные переменные	x_{N+1}	a_{10}	a_{11}	a_{12}	...	a_{1N}	1	0	...	0
	x_{N+2}	a_{20}	a_{21}	a_{22}	...	a_{2N}	0	1	...	0

	x_{N+m}	a_{m0}	a_{m1}	a_{m2}	...	a_{mN}	0	0	...	1
	$f(x)$	$f_0(x)$	z_1	z_2	...	z_N	0	0	...	0

Допустим теперь, что произведена следующая коррекция: значение одной из небазисных переменных x_i ($i=1, \dots, N$) взято равным единице вместо нуля, а остальные небазисные переменные по-прежнему равны нулю. Ненулевые базисные переменные изменяются так, что решение остается допустимым, т. е. все ограничения по-прежнему удовлетворяются. Тогда из (3.27) следует

$$f(x) - f(x_0) = z_i, \quad i=1, \dots, N. \quad (3.30)$$

Таким образом, z_i представляет собой величину, на которую изменяется целевая функция $f(x)$ по сравнению с ее значением при допустимом базисном решении, если производится вышеописанная коррекция. Если z_i положительно, то значение $f(x)$ в результате коррекции возрастет. Так как мы имеем задачу максимизации, то эта коррекция улучшает решение и приближает его к оптимальному. Если z_i отрицательно, то в результате изменений $f(x)$ уменьшится. Основываясь на этом, можно произвести следующее испытание таблицы.

Проверить все z_i ($i=1, \dots, N$). Если одно или несколько z_i положительны, то это означает, что текущее решение еще не является оптимальным и может быть улучшено описанной заменой. Если все z_i отрицательны, то достигнуто оптимальное решение, так как дальней-

шее улучшение невозможно. Если некоторые из z_i равны нулю, то существует несколько эквивалентных оптимальных решений. Читатель найдет более подробное рассмотрение этого особого случая в [11].

Если более одной z_i положительны, возникает вопрос — какое из соответствующих x_i надо заменить. При этом выборе одним из наиболее важных факторов является абсолютная величина $|z_i|$. Чем она больше, тем больше будет возрастать величина целевой функции. Поэтому надо стремиться выбрать

$$\max |z_i|, \quad z_i > 0. \quad (3.31)$$

Пусть выбранная для замены небазисная переменная есть x_l . Следующая задача заключается в том, чтобы определить, какую из базисных переменных надо извлечь из базиса и поместить вместо x_l в число небазисных переменных. Главное в этом случае быть уверенным в том, что новое решение допустимо, т. е. что после замены все ограничения задачи по-прежнему удовлетворяются.

Поскольку при замене по-прежнему $x_i = 0$ при $i \neq l$, $i = 1, \dots, N$, то из (3.22)

$$x_{N+j} = b_j - a_{jl}x_l, \quad j = 1, \dots, m. \quad (3.32)$$

При увеличении x_l те x_{N+j} , для которых $a_{jl} > 0$, будут убывать, и максимальное возможное значение x_l будет достигнуто, когда первое из таких x_{N+j} обратится в 0. Если для всех j будет

$$a_{jl} \leq 0, \quad (3.33)$$

то x_l можно увеличивать неограниченно и, поскольку в этом случае

$$f(\mathbf{x}) = f_0(\mathbf{x}) + z_l x_l \quad (3.34)$$

и $z_l > 0$, то $f(\mathbf{x})$ будет также неограниченно возрастать. Итак, если выполняется условие (3.33), то задача линейного программирования не имеет решения (целевая функция не ограничена). Поэтому будем предполагать, что

$$a_{jl} > 0 \quad (3.35)$$

для некоторых j . Как мы уже отмечали, максимальное значение x_l (а тем самым и максимальное значение $f(\mathbf{x})$) будет достигнуто, когда первое из x_{N+j} обратится

в 0. Эта переменная и выводится из базиса. Ее номер r в соответствии с (3.32), (3.35) определяется из условия

$$\frac{b_r}{a_{rl}} = \min \left\{ \frac{b_j}{a_{jl}}, a_{jl} > 0 \right\}. \quad (3.36)$$

Элемент a_{rl} симплексной таблицы, стоящий на пересечении r -й строки и l -го столбца, называется ведущим (или разрешающим) элементом. При этом r -я строка, соответствующая исключаемой из базиса переменной, называется ведущей строкой, а l -й столбец, соответствующий новой базисной переменной, — ведущим столбцом.

Далее новый набор базисных переменных должен быть выражен из уравнений (3.16) через небазисные переменные подобно уравнению (3.22). Коэффициенты в этом уравнении a'_{ji} представляют собой новую симплексную таблицу. Нетрудно видеть, что a'_{ji} пересчитываются из a_{ji} следующим образом:

$$a'_{ji} = a_{ji} - a_{ri} \frac{a_{jl}}{a_{rl}}, \quad i \neq l, \quad (3.37)$$

$$b'_j = a'_{j0} = a_{j0} - a_{r0} \frac{a_{jl}}{a_{rl}}, \quad j \neq r, \quad (3.38)$$

$$a'_{jl} = -\frac{a_{jl}}{a_{rl}}, \quad j \neq r, \quad (3.39)$$

$$a'_{rl} = \frac{1}{a_{rl}}, \quad (3.40)$$

$$b'_r = a'_{r0} = \frac{a_{r0}}{a_{rl}}. \quad (3.41)$$

Аналогичным образом пересчитываются величины z_i :

$$z'_i = z_i - z_l \frac{a_{ri}}{a_{rl}}, \quad i \neq l, \quad z'_l = -\frac{z_l}{a_{rl}}. \quad (3.42)$$

Новое допустимое базисное решение будет

$$x_i = 0, \quad i \neq l, \quad i = 1, \dots, N, \quad (3.43)$$

$$x_{N+r} = 0, \quad (3.44)$$

$$x_{N+j} = b_j - a_{jl} \frac{b_r}{a_{rl}}, \quad j \neq r, \quad j = 1, \dots, m, \quad (3.45)$$

$$x_{N+r} = \frac{b_r}{a_{rl}}. \quad (3.46)$$

Таблица 3.2

Новая таблица

	x_1	x_2	...	x_l	...	x_N	x_{N+1}	...	x_{N+r}	...	x_{N+m}
x_{N+1}	a'_{10}	a'_{11}	a'_{12}		0	a'_{1N}	1		a'_{1l}		0
x_{N+2}	a'_{20}	a'_{21}	a'_{22}		0	a'_{2N}	0		a'_{2l}		0
·	·	·	·		·	·	·		·		·
·	·	·	·		·	·	·		·		·
x_{N+r}	a'_{r0}	a'_{r1}	a'_{r2}		1	a'_{rN}	0		a'_{rl}		0
·	·	·	·		·	·	·		·		·
·	·	·	·		·	·	·		·		·
x_{N+m}	a'_{m0}	a'_{m1}	a'_{m2}		0	a'_{mN}	0		a'_{ml}		1
$z=f(x)$	$f'_0(x)$	z'_1	z'_2		0	z'_N	0		z'_l		0

Табл. 3.2 представляет собой новую симплексную таблицу. С ней прodelьваются те же преобразования, и процесс продолжается до тех пор, пока не будет найдено оптимальное решение (или пока мы не убедимся в его отсутствии).

В качестве иллюстрации ниже приведен детально решенный простой пример.

Пример. Рассмотрим пример, представленный формулами от (2.15) до (2.17) в параграфе 2.4 и решенный графически на рис. 2.2. Этот пример будет теперь решен с помощью описанных выше симплексных таблиц. Задача заключается в следующем:

$$\max\{x_1+4x_2 \mid x_1 \leq 10, x_2 \leq 10, x_1+x_2 \leq 14, x_1, x_2 \geq 0\}. \quad (3.47)$$

Введя дополнительные переменные, перепишем первые три неравенства в виде

$$\begin{aligned} x_1+x_2+x_3 &= 14, \\ x_1 &+ x_4 = 10, \\ x_2 &+ x_5 = 10. \end{aligned} \quad (3.48)$$

В данном случае

$$c^T = [1 \ 4], \quad b = \begin{bmatrix} 14 \\ 10 \\ 10 \end{bmatrix},$$

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Начальными базисными переменными являются x_3, x_4, x_5 . Начальная таблица будет следующей:

		x_1	x_2	Небазисные ← переменные
Базисные переменные →	x_3	14	1	1
	x_4	10	1	0
	x_5	10	0	<u>1</u>
$f(x)$		$f_0(x) = 0$	$z_1 = 1$	$z_2 = 4$

← Ведущий элемент

Так как столбцы таблицы, соответствующие базисным переменным, образуют единичную матрицу, то нет необходимости каждый раз ее воспроизводить.

Так как $c_3 = c_4 = c_5 = 0$, то

$$f_0(x) = \sum_{j=3}^5 c_j x_j = 0.$$

Пользуясь (3.28), вычисляем

$$z_1 = c_1 = 1 > 0,$$

$$z_2 = c_2 = 4 > 0.$$

Так как

$$|z_2| = 4 > |z_1| = 1,$$

то для включения в базис выбираем x_2 . Теперь

$$\frac{b_1}{a_{12}} = 14 > \frac{b_3}{b_{32}} = 10.$$

Таким образом, в качестве ведущей строки выбрана третья строка, т. е. x_5 будет изъята из базиса, а $a_{32} = 1$ является ведущим элемен-

том. В соответствии с формулами от (3.37) до (3.42) вычисляется новая таблица:

		x_1	x_3	
				← Небазисные переменные
Базисные переменные →	x_3	4	$\boxed{1}$	0
	x_4	10	1	0
	x_2	10	0	1
	$f(x)$	40	1	-4

Теперь только $z_1 = 1 > 0$, так что столбец 1 будет ведущим столбцом и x_1 будет включен в базис.

Так как

$$\frac{b'_1}{a_{11}} = 4 < \frac{b'_2}{a_{21}} = 10,$$

то x_3 выбирается для изъятия из базиса.

Очередная таблица приведена ниже

		x_3	x_5
x_1	4	1	0
x_4	6	0	0
x_2	10	0	1
$f(x)$	44	-1	-4

Поскольку z_1, z_2 неположительны, достигнуто оптимальное решение:

$$x_1 = 4, \quad x_2 = 10, \quad z_{\text{opt}} = 44.$$

Оно совпадает с решением, полученным на рис. 2.2.

3.2. Квадратичное программирование

Общая формулировка

Задача квадратичного программирования [7—11] в общем виде была представлена в гл. 2 формулами (2.19) и (2.20). Здесь мы сформулируем ее более детально. Не умаляя общности, можно рассмотреть лишь

задачу минимизации. Целевая функция в этой задаче равна

$$f(\mathbf{x}) = \mathbf{c}^T \mathbf{x} + \mathbf{x}^T D \mathbf{x}, \quad (3.49)$$

где \mathbf{x} — n -мерный переменный вектор, \mathbf{c} — n -мерный постоянный вектор, D — постоянная матрица размерности $n \times n$ (дополнительные свойства D будут установлены ниже).

На основании изложенного в начале параграфа 3.1 можно выразить линейные ограничения задачи в виде

$$A\mathbf{x} = \mathbf{b}, \quad x_i \geq 0, \quad i = 1, \dots, n, \quad (3.50)$$

где \mathbf{b} — m -мерный постоянный вектор, A — постоянная матрица размерности $m \times n$.

При обозначениях (3.50) вектор \mathbf{x} может содержать также дополнительные переменные, если некоторые из исходных ограничений являются неравенствами. Уравнение (3.50) можно записать также в следующем виде:

$$g(\mathbf{x}) = A\mathbf{x} - \mathbf{b} = 0. \quad (3.51)$$

Задачу квадратичного программирования можно записать в следующей сжатой форме:

$$\min \{ \mathbf{c}^T \mathbf{x} + \mathbf{x}^T D \mathbf{x} \mid g(\mathbf{x}) = 0; x_i \geq 0, i = 1, \dots, n \}. \quad (3.52)$$

Для решения этой задачи естественно было бы применить теорему Куна — Таккера, рассмотренную в параграфе 2.6. Одним из основных предположений теоремы Куна — Таккера является предположение о выпуклости целевой функции всех ограничений. Из (3.51) и (3.52) видно, что $\mathbf{c}^T \mathbf{x}$ и $g(\mathbf{x})$ линейны и, следовательно, выпуклы. Остается проверить только условия, которым должна удовлетворять матрица D , для того чтобы квадратичная форма $\mathbf{x}^T D \mathbf{x}$ была выпуклой. На этот вопрос дает ответ теорема 3.2.

Теорема 3.2. [7]. *Положительно полуопределенная квадратичная форма $\mathbf{x}^T D \mathbf{x}$ является выпуклой функцией.*

Таким образом, надо показать, что если

$$\mathbf{x}^T D \mathbf{x} \geq 0 \quad \text{при любом } \mathbf{x}, \quad (3.53)$$

го для любых двух точек \mathbf{x}_1 и \mathbf{x}_2

$$[\lambda \mathbf{x}_2 + (1 - \lambda) \mathbf{x}_1]^T D [\lambda \mathbf{x}_2 + (1 - \lambda) \mathbf{x}_1] \leq \\ \leq \lambda \mathbf{x}_2^T D \mathbf{x}_2 + (1 - \lambda) \mathbf{x}_1^T D \mathbf{x}_1 \quad (3.54)$$

при любом $0 \leq \lambda \leq 1$.

Доказательство. Левая часть неравенства (3.54) может быть записана в виде

$$[\mathbf{x}_1 + \lambda (\mathbf{x}_2 - \mathbf{x}_1)]^T D [\mathbf{x}_1 + \lambda (\mathbf{x}_2 - \mathbf{x}_1)] = \\ = \mathbf{x}_1^T D \mathbf{x}_1 + 2\lambda (\mathbf{x}_2 - \mathbf{x}_1)^T D \mathbf{x}_1 + \lambda^2 (\mathbf{x}_2 - \mathbf{x}_1)^T D (\mathbf{x}_2 - \mathbf{x}_1). \quad (3.55)$$

Так как предположение о положительной полуопределенности означает, что каждая часть равенства (3.55) неотрицательна и так как $0 \leq \lambda \leq 1$ (или $\lambda \geq \lambda^2$), то

$$\lambda (\mathbf{x}_2 - \mathbf{x}_1)^T D (\mathbf{x}_2 - \mathbf{x}_1) \geq \lambda^2 (\mathbf{x}_2 - \mathbf{x}_1)^T D (\mathbf{x}_2 - \mathbf{x}_1). \quad (3.56)$$

Используя (3.56) в (3.55), можно записать

$$[\mathbf{x}_1 + \lambda (\mathbf{x}_2 - \mathbf{x}_1)]^T D [\mathbf{x}_1 + \lambda (\mathbf{x}_2 - \mathbf{x}_1)] \leq \\ \leq \mathbf{x}_1^T D \mathbf{x}_1 + 2\lambda (\mathbf{x}_2 - \mathbf{x}_1)^T D \mathbf{x}_1 + \lambda (\mathbf{x}_2 - \mathbf{x}_1)^T D (\mathbf{x}_2 - \mathbf{x}_1) = \\ = \mathbf{x}_1^T D \mathbf{x}_1 + \lambda (\mathbf{x}_2 - \mathbf{x}_1)^T D \mathbf{x}_1 + \lambda (\mathbf{x}_2 - \mathbf{x}_1)^T D \mathbf{x}_2 = \\ = \lambda \mathbf{x}_2^T D \mathbf{x}_2 + (1 - \lambda) \mathbf{x}_1^T D \mathbf{x}_1. \quad (3.57)$$

Формула (3.57) идентична (3.54), что и требовалось доказать.

Как указывалось в 2.5, выпуклость гарантирует глобальный экстремум.

Теперь мы кратко рассмотрим вычислительные алгоритмы для решения задачи квадратичного программирования. Более подробное рассмотрение и описание других методов читатель найдет в ряде источников [7—11].

Метод Вульфа [12]

По аналогии с (2.23) запишем функцию Лагранжа $L(\mathbf{x}, \mathbf{u})$ задачи квадратичного программирования, сформулированной в (3.52):

$$L(\mathbf{x}, \mathbf{u}) = \mathbf{c}^T \mathbf{x} + \mathbf{x}^T D \mathbf{x} + \mathbf{u}^T \mathbf{g}(\mathbf{x}) \quad (3.58)$$

или, в скалярной форме согласно (2.20),

$$L(\mathbf{x}, \mathbf{u}) = \sum_{i=1}^n c_i x_i + \sum_{i=1}^n \sum_{j=1}^m x_i d_{ij} x_j + \sum_{j=1}^m u_j g_j(\mathbf{x}). \quad (3.59)$$

Так как все ограничения $g_j(\mathbf{x})$ являются равенствами, то к данной задаче минимизации применим метод, аналогичный методу множителей Лагранжа, роль которых здесь играют u_j . Действительно, в этом случае условия Куна — Таккера (2.32) принимают вид [9]

$$\left. \frac{\partial L}{\partial u_j} \right|_{\mathbf{x}^0, \mathbf{u}^0} = 0, \quad j = 1, \dots, m. \quad (3.60)$$

Понятно, что (2.31) остается справедливым, так как условие $x_i \geq 0$ не снимается.

Применяя первое из неравенств (2.31) к (3.58), с учетом (3.52) получаем

$$\frac{\partial L}{\partial \mathbf{x}} = \mathbf{c} + 2D\mathbf{x} + A^T \mathbf{u} \geq 0. \quad (3.61)$$

Применяя (3.60) к (3.58), вновь получаем исходные ограничивающие равенства в виде (3.50). Введем теперь новую дополнительную переменную

$$\mathbf{v} = \frac{\partial L}{\partial \mathbf{x}}, \quad (3.62)$$

или

$$v_i = \frac{\partial L}{\partial x_i}, \quad i = 1, \dots, n. \quad (3.63)$$

Используя (3.50), (3.61), (3.63) и (2.31), можно переписать задачу квадратичного программирования (3.52) в виде следующей эквивалентной задачи. Найти вектор \mathbf{x}^0 , удовлетворяющий следующим условиям:

$$A\mathbf{x} = \mathbf{b}, \quad (3.64)$$

$$2D\mathbf{x} - \mathbf{v} + A^T \mathbf{u} = -\mathbf{c}, \quad (3.65)$$

$$\mathbf{x}^T \mathbf{v} = 0, \quad (3.66)$$

$$x_i \geq 0, \quad v_i \geq 0, \quad i = 1, \dots, n.$$

Из теоремы Куна — Таккера и предыдущего рассмотрения следует, что вектор \mathbf{x}^0 , являющийся решением уравнений от (3.64) до (3.66), является также оптимальным решением задачи квадратичного программи-

рования (3.52). Следует заметить, что (3.64) и (3.65) линейны относительно переменных \mathbf{x} , \mathbf{v} и \mathbf{u} и их можно переписать в следующей матричной форме:

$$\begin{bmatrix} A & 0 & 0 \\ 2D & -I_n & A^T \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{v} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ -\mathbf{c} \end{bmatrix}. \quad (3.67)$$

Размерности подматриц в матрицах левой части равенства (3.67) следующие:

$$A: m \times n,$$

$$2D: n \times n,$$

$-I_n: n \times n$ (отрицательная единичная матрица). Размерность всей матрицы $(m+n) \times (m+2n)$. Таким образом, имеем $m+n$ уравнений с $m+2n$ неизвестными. Так как $x_i v_i = 0$ ($i=1, \dots, n$) и все $x_i \geq 0$ ($v_i \geq 0$), то только n из $2n$ переменных \mathbf{x} , \mathbf{v} могут отличаться от нуля. Это означает, что $n+m$ уравнений (3.67) могут иметь только $n+m$ ненулевых переменных. Следовательно, единственные решения уравнения (3.67), которые должны рассматриваться, являются базисными решениями. Чтобы идентифицировать базисные решения уравнения (3.67), можно непосредственно применить симплексный метод линейного программирования. Это составляет основную идею алгоритма Вулфа. Для детального ознакомления читатель может обратиться к источникам [7; 9, гл. 8; 12].

Метод Фрэнк и Вулфа

Алгоритм Фрэнк и Вулфа [13] исходит из тех же уравнений (3.64) и (3.65). Отличие этого алгоритма от алгоритма Вулфа заключается в том, что третье условие (3.66) вначале не учитывается. Делается попытка получить допустимое решение уравнения (3.67) без учета условия

$$x_i v_i = 0, \quad i=1, \dots, n, \quad (3.68)$$

которое является тем же условием (3.66). Если не будет получено ни одного допустимого решения, процесс останавливают. Если, однако, уравнение (3.67) имеет

допустимое решение, не обязательно удовлетворяющее уравнению (3.68), то исходная задача квадратичного программирования (3.52) обладает ограниченным оптимальным решением [7]. После того как получено допустимое базисное решение уравнения (3.67), алгоритм, используя симплексный метод, переходит от одного допустимого базисного решения к другому до тех пор, пока не будет выполнено условие (3.68), т. е. пока не будет достигнуто оптимальное решение.

3.3. Нелинейное программирование

Градиентный подход

Как указывалось в параграфе 2.4, задача математического программирования хотя бы с одним нелинейным ограничением является задачей нелинейного программирования. С другой стороны, даже когда все ограничения линейны, но нелинейна целевая функция, вновь имеем задачу нелинейного программирования. Естественно, что квадратичное программирование является частным случаем нелинейного программирования. Однако, как следует из параграфа 3.2, применение теоремы Куна — Таккера к задаче квадратичного программирования приводит ее к задаче типа линейного программирования, что позволяет применить симплексный метод. Вследствие этого задачи квадратичного программирования обычно выделяют в отдельный класс.

Численное решение задачи нелинейного программирования связано со значительно большими трудностями. В отличие от случаев линейного или квадратичного программирования (с выпуклой целевой функцией), не существует вычислительного алгоритма, который гарантировал бы решение любой задачи нелинейного программирования за конечное число шагов. Было разработано несколько различных алгоритмов. Некоторые из алгоритмов используют так называемый градиентный подход, который будет рассмотрен в этом параграфе [7, гл. 9; 9, гл. 11—14].

Вначале будет рассмотрен случай линейных ограничений и нелинейной целевой функции.

Рассматривается следующая задача:

$$\max \{f(\mathbf{x}) \mid A\mathbf{x} \leq \mathbf{b}; x_i \geq 0, i=1, \dots, n\}, \quad (3.69)$$

где \mathbf{x} — n -мерный переменный вектор, A — постоянная матрица размерности $m \times n$, \mathbf{b} — m -мерный постоянный вектор, $f(\mathbf{x})$ — нелинейная, вогнутая целевая функция переменных \mathbf{x} .

В качестве иллюстрации на рис. 3.1 представлена двумерная задача ($n=2$). Заштрихованная область представляет собой допустимую область, т. е. множество всех точек $\mathbf{x} = (x_1, x_2)$, которые удовлетворяют всем ограничениям. Каждая из кривых от F_0 до F_5 представляет собой геометрическое место точек постоянного

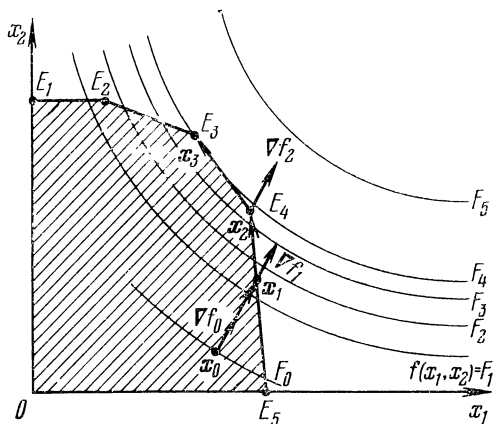


Рис. 3.1. Решение градиентным методом.

значения целевой функции $f(x_1, x_2)$. Номера кривых соответствуют возрастающим значениям целевой функции, т. е. $F_0 < F_1 < F_2 < F_3 < F_4 < F_5$. Точки: $0, E_1, \dots, E_5$ являются крайними точками допустимой области. Из рис. 3.1 видно, что крайняя точка E_3 представляет оптимальное решение, при котором целевая функция имеет максимальное значение

$$f_{\max}(\mathbf{x}) = F_4$$

и которое удовлетворяет всем ограничениям. Вопрос

заключается в том, как с помощью вычислительного алгоритма достичь оптимальной точки E_3 , учитывая, что $f(\mathbf{x})$ нелинейна?

Обычно начинают от точки \mathbf{x}_0 в допустимой области. Так как решается задача максимизации, то следует двигаться от точки \mathbf{x}_0 в направлении возрастания $f(\mathbf{x})$. Более того, желательно, чтобы $f(\mathbf{x})$ возрастала возможно быстрее. Градиент функции $f(\mathbf{x})$:

$$\nabla f(\mathbf{x}) = \left| \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right| \quad (3.70)$$

есть вектор, направление которого совпадает с направлением наискорейшего возрастания величины данной функции. Поэтому траектория должна выходить из начальной точки \mathbf{x}_0 в направлении градиента. Однако, так как $f(\mathbf{x})$ нелинейна, то направление градиента будет меняться вдоль траектории. Поэтому направление продвижения надо от времени до времени пересматривать. Одна из важнейших задач для градиентных методов заключается в определении оптимальной величины шага для каждой рассматриваемой точки.

Более общий класс методов по сравнению с чисто градиентными составляют такие алгоритмы, в которых направление движения линейно зависит от градиента, но не обязательно совпадает с ним. Такие методы мы будем также называть градиентными. Введем общие обозначения, которые мы будем применять при дальнейшем рассмотрении градиентного подхода:

\mathbf{x}_i — точка, лежащая в допустимой области, достигаемая после i -го шага;

∇f_i — градиент функции $f(\mathbf{x}_i) = \nabla f(\mathbf{x}_i) = \nabla f(\mathbf{x})_{\mathbf{x}=\mathbf{x}_i}$;

\mathbf{r}_i — единичный вектор, совпадающий по направлению с шагом, который делается из точки \mathbf{x}_i ; таким образом,

$$\mathbf{r}_i^T \mathbf{r}_i = 1; \quad (3.71)$$

d_i — скаляр, величина которого равна длине шага из точки \mathbf{x}_i в направлении \mathbf{r}_i .

Используя введенные обозначения, можно записать

$$\mathbf{x}_{i+1} = \mathbf{x}_i + d_i \mathbf{r}_i. \quad (3.72)$$

Новый вектор направления \mathbf{r}_i следующим образом зависит от градиента ∇f_i ;

$$\mathbf{r}_i = H_i \nabla f_i, \quad (3.73)$$

где H_i — матрица размера $n \times n$, называемая *направляющей матрицей* или *метрикой* [14].

Различные градиентные методы отличаются друг от друга метрикой H_i . Например, если

$$H_i = \frac{I_n}{|\nabla f_i|}, \quad (3.74)$$

где I_n — n -мерная единичная матрица, то направление следующего шага коллинеарно с градиентом ∇f_i в исходной точке \mathbf{x}_i . Размер шага d_i выбирается в этом случае так, что значение целевой функции в следующей точке $f(\mathbf{x}_{i+1})$ оказывается максимальным. Этот метод обычно называют методом *наибыстрейшего подъема* [14]. Используя формулы от (3.72) до (3.74), получаем

$$f(\mathbf{x}_{i+1}) = f(\mathbf{x}_i + d_i \nabla f_i). \quad (3.75)$$

Необходимое условие максимума функции (3.75)

$$\frac{df(\mathbf{x}_{i+1})}{d(d_i)} = \nabla^T f_i \nabla f_{i+1} = 0. \quad (3.76)$$

Однако надо быть уверенным в том, что следующая точка допустима. Из (3.76) следует, что при методе *наибыстрейшего подъема* градиенты в двух соседних точках *ортогональны*. Это означает, что к оптимальному решению ведет зигзагообразный путь, с поворотами под прямым углом в каждой точке и обычно с относительно малыми длинами шагов.

Метод *наибыстрейшего подъема* служит для решения задач на безусловный экстремум. Ниже мы рассмотрим его модификации, пригодные для решения задач с ограничениями.

Метод возможных направлений

Метод *возможных направлений* был разработан Зойтендейком [16].

Рассмотрим задачу, сформулированную в (3.69). Матрицу ограничений A размера $m \times n$ разобьем на

векторы-строки:

$$A = \begin{bmatrix} a_1^T \\ \vdots \\ a_m^T \end{bmatrix}. \quad (3.77)$$

Тогда ограничения можно представить в следующем виде:

$$\mathbf{a}_j^T \mathbf{x} \leq b_j, \quad j=1, \dots, m. \quad (3.78)$$

Рассмотрим задачу перехода из точки \mathbf{x}_i в следующую точку \mathbf{x}_{i+1} , где \mathbf{x}_i и \mathbf{x}_{i+1} связаны между собой следующим образом:

$$\mathbf{x}_{i+1} = \mathbf{x}_i + d_i \mathbf{r}_i, \quad (3.79)$$

а при достаточно малом d_i \mathbf{x}_{i+1} допустимо, т. е.

$$\mathbf{a}_j^T \mathbf{x}_{i+1} \leq b_j. \quad (3.80)$$

Из предположения о допустимости \mathbf{x}_i следует, что условие (3.80) эквивалентно неравенству

$$\mathbf{a}_j^T \mathbf{r}_i \leq 0, \quad j \in J, \quad (3.81)$$

где J — множество всех индексов j , для которых

$$\mathbf{a}_j^T \mathbf{x}_i = b_j. \quad (3.82)$$

Направление \mathbf{r}_i , удовлетворяющее условиям (3.81) и (3.82), называется *возможным направлением*. Переход из \mathbf{x}_i в \mathbf{x}_{i+1} должен быть сделан в направлении возрастания значения целевой функции $f(\mathbf{x})$. Другими словами, должно быть удовлетворено неравенство

$$\mathbf{r}_i^T \nabla f_i > 0. \quad (3.83)$$

Зойтендейк [16] предложил несколько методов отыскания возможных направлений итераций из точки \mathbf{x}_i . Все эти методы представляют собой последовательные решения вспомогательных задач математического программирования следующего типа.

Во всех методах отыскивается максимум одной и той же вспомогательной функции

$$\nabla f_i^T \mathbf{r}_i \quad (3.84)$$

при ограничениях, имеющих, согласно (3.81), вид

$$\mathbf{a}_j^T \mathbf{r}_i \leq 0, \quad j \in J. \quad (3.85)$$

Методы (будем обозначать их T_i) различаются дополнительными ограничениями:

$$T_1: \quad \mathbf{r}_i^T \mathbf{r}_i \leq 1, \quad (3.86)$$

$$T_2: \quad -1 \leq r_{ik} \leq 1, \quad k = 1, \dots, n, \quad (3.87)$$

где r_{ik} — k -я компонента вектора \mathbf{r}_i ,

$$T_3: \quad \left. \begin{array}{l} r_{ik} \leq 1 \quad \text{при } \nabla f_{ik} > 0, \\ r_{ik} \geq -1 \quad \text{при } \nabla f_{ik} < 0, \end{array} \right\} \quad (3.88)$$

где ∇f_{ik} — k -я компонента градиента ∇f_i ,

$$T_4: \quad \nabla f_i^T \mathbf{r}_i \leq 1, \quad (3.89)$$

$$T_5: \quad \mathbf{a}_j^T (\mathbf{r}_i + \mathbf{x}_i) \leq b_j \quad \text{при всех } j. \quad (3.90)$$

Условие (3.85) фактически включено в (3.90).

Так как градиент ∇f_i в точке \mathbf{x}_i задан, то целевая функция в (3.84) линейна. Как видно из ограничений, методы от T_2 до T_5 приводят к задачам линейного программирования. Только в случае T_1 неравенство (3.86) представляет собой нелинейное ограничение. Каждая линейная вспомогательная задача может быть решена с помощью симплекс-алгоритма. После того как найдено возможное направление \mathbf{r}_i в точке \mathbf{x}_i , надо определить длину соответствующего шага. Это можно осуществить с помощью следующих двух кандидатов для этой величины:

1. Значения $d_i^{(1)}$, при котором перемещение вдоль вектора $d_i \mathbf{r}_i$ выводит за пределы допустимой области, определяемой формулой (3.78).

2. Значения $d_i^{(2)}$, при котором перемещение вдоль вектора $d_i \mathbf{r}_i$ приводит к максимуму функции $f(\mathbf{x})$, т. е. значения $d_i^{(2)}$, при котором выполняется равенство

$$\mathbf{r}_i^T \nabla f(\mathbf{x}_i + d_i^{(2)} \mathbf{r}_i) = 0. \quad (3.91)$$

(Предлагаем читателю сравнить с (3.76).)

После того как вычислены оба значения $d_i^{(1)}$ и $d_i^{(2)}$ берется меньшее из них

$$d_i = \min(d_i^{(1)}, d_i^{(2)}). \quad (3.92)$$

Описанный итерационный процесс продолжается до тех пор, пока не будет достигнута точка \mathbf{x}_p , где

$$\max [\nabla f_p^T \mathbf{r}_p] = 0, \quad (3.93)$$

т. е. где нет нового направления \mathbf{r}_p , которое ведет к увеличению целевой функции $f(\mathbf{x})$.

Зойтендейк развил свой метод также применительно к задачам с нелинейными ограничениями, например вида

$$g_j(\mathbf{x}) \leq b_j, \quad j=1, \dots, m, \quad (3.94)$$

где функция $g_j(\mathbf{x})$ — нелинейные выпуклые функции. Допустим, что последняя итерация привела в допустимую точку \mathbf{x}_i . Обозначим через J_1 класс всех ограничений, для которых

$$g_j(\mathbf{x}_i) \leq b_j, \quad j \in J_1. \quad (3.95)$$

Гиперплоскость, касательная к каждой из поверхностей (3.95), описывается следующим уравнением:

$$\nabla^T g_j(\mathbf{x}_i) \mathbf{x} = \nabla^T g_j(\mathbf{x}_i) \mathbf{x}_i. \quad (3.96)$$

Этот тип гиперплоскости называется *опорной гиперплоскостью*.

Необходимо найти новую допустимую точку

$$\mathbf{x}_{i+1} = \mathbf{x}_i + d_i \mathbf{r}_i, \quad (3.97)$$

при переходе в которую целевая функция возрастает, т. е. выполняется неравенство

$$f(\mathbf{x}_{i+1}) > f(\mathbf{x}_i). \quad (3.98)$$

Если \mathbf{x}_{i+1} — допустимая точка, т. е. если удовлетворяется условие

$$g_j(\mathbf{x}_{i+1}) \leq b_j, \quad j \in J_1, \quad (3.99)$$

то необходимо, чтобы

$$\nabla g_j(\mathbf{x}_i) \mathbf{r}_i \leq 0. \quad (3.100)$$

Для определения направления \mathbf{r}_i Зойтендейк [16] предложил следующую процедуру. Вводится дополнительная переменная s и решается следующая задача

математического программирования. Найти максимум s при ограничениях

$$\nabla^T g_j(\mathbf{x}_i) \mathbf{r}_i + s \leq 0, \quad j \in J_1, \quad (3.101)$$

$$-\nabla^T f(\mathbf{x}_i) \mathbf{r}_i + s \leq 0, \quad (3.102)$$

$$\mathbf{r}_i^T \mathbf{r}_i = 1. \quad (3.103)$$

В результате решения этой задачи оказывается найденным новое направление \mathbf{r}_i , при котором траектория остается в допустимой области и выполняется условие $\nabla f^T \mathbf{r}_i > 0$, т. е. увеличивается значение целевой функции. После того как найдено \mathbf{r}_i , устанавливают максимальную длину шага d_m , равную наибольшему d_i , при котором

$$g_j(\mathbf{x}_i + d_i \mathbf{r}_i) \leq b_j \quad \text{для всех } j. \quad (3.104)$$

Чтобы определить фактическую длину шага $0 \leq d_i \leq d_m$, надо решить следующую задачу:

$$\max \{f(\mathbf{x}_i + d_i \mathbf{r}_i) \mid 0 \leq d_i \leq d_m\}. \quad (3.105)$$

Более подробные сведения о методе Зойтендейка можно найти в его книге [16].

Метод проекции градиента

Метод проекции градиента был разработан Розеном [17, 18]. Прежде чем перейти непосредственно к описанию алгоритма, необходимо ввести понятие проекции градиента. Рассмотрим систему линейных ограничений в виде (3.78). Если точка \mathbf{x} принадлежит границе допустимой области, то для нее некоторые из неравенств (3.78) обращаются в равенства:

$$\mathbf{a}_j^T \mathbf{x} = b_j, \quad j \in J, \quad (3.106)$$

где J — некоторое подмножество индексов из $\{1, \dots, m\}$. Для определенности будем считать, что $J = \{1, \dots, q\}$, т. е. что в равенства обращаются q первых неравенств. Множество точек, удовлетворяющих (3.106), есть пересечение q гиперплоскостей, т. е. $(n - q)$ -мерное линейное многообразие (предполагается, что $\mathbf{a}_1, \dots, \mathbf{a}_q$ линейно независимы). Подпространство той же размерности, полученное параллельным переносом этого многообразия

в начало координат, будем обозначать Q . Иначе говоря, Q — множество точек, удовлетворяющих условию

$$\mathbf{a}_j^T \mathbf{x} = 0, \quad j = 1, \dots, q. \quad (3.107)$$

Подпространство, базисом которого служат векторы $\mathbf{a}_1, \dots, \mathbf{a}_q$, обозначим \mathbf{Q} , т. е. всякое $\mathbf{x} \in \mathbf{Q}$ можно представить в виде

$$\mathbf{x} = \sum_{j=1}^q u_j \mathbf{a}_j. \quad (3.108)$$

Легко проверить, что Q и \mathbf{Q} ортогональны. Любой вектор \mathbf{x} может быть однозначно представлен в виде

$$\mathbf{x} = \mathbf{x}_Q + \mathbf{x}_Q, \quad (3.109)$$

где

$$\mathbf{x}_Q \in Q, \quad \mathbf{x}_Q \in \mathbf{Q}, \quad \mathbf{x}_Q^T \mathbf{x}_Q = 0. \quad (3.110)$$

Вектор \mathbf{x}_Q называется *проекцией* вектора \mathbf{x} на подпространство Q (соответственно \mathbf{x}_Q — проекцией на \mathbf{Q}). Проекция обладает следующими свойствами:

$$1. \quad \mathbf{x}_Q^T (\mathbf{x} - \mathbf{x}_Q) = 0, \quad (3.111)$$

$$2. \quad \|\mathbf{x}_Q - \mathbf{x}\| < \|\mathbf{y} - \mathbf{x}\| \text{ для всех } \mathbf{y} \in Q, \mathbf{y} \neq \mathbf{x}_Q. \quad (3.112)$$

Выпишем явную формулу для вычисления проекции. Запишем (3.107) в матричном виде

$$A_q \mathbf{x} = 0, \quad (3.113)$$

где A_q — матрица $q \times n$, строками которой служат векторы \mathbf{a}_j^T , $j = 1, \dots, q$. При этом (3.108) принимает вид $A_q^T \mathbf{u} = \mathbf{x}$. Докажем, что тогда

$$\mathbf{x}_Q = P_Q \mathbf{x}, \quad (3.114)$$

где \mathbf{x} — произвольный вектор, а матрица проектирования P_Q задается формулой

$$P_Q = I - A_q^T (A_q A_q^T)^{-1} A_q. \quad (3.115)$$

Здесь $(A_q A_q^T)^{-1}$ существует в силу предположения о линейной независимости $\mathbf{a}_1, \dots, \mathbf{a}_q$. Обозначим

$$\mathbf{x}_Q = P_Q \mathbf{x}, \quad (3.116)$$

$$\mathbf{x}_Q = \mathbf{x} - \mathbf{x}_Q \quad (3.117)$$

и покажем, что выполняется (3.110). Действительно,

$$A_q x_Q = A_q (I - A_q^T (A_q A_q^T)^{-1} A_q) x = (A_q - A_q) x = 0, \quad (3.118)$$

т. е. $x_Q \in Q$. Далее

$$x_Q = x - (I - A_q^T (A_q A_q^T)^{-1} A_q) x = A_q^T (A_q A_q^T)^{-1} A_q x = A_q^T u, \quad (3.119)$$

где

$$u = (A_q A_q^T)^{-1} A_q x. \quad (3.120)$$

Следовательно, $x_Q \in Q$. Наконец,

$$\begin{aligned} x_Q^T x_Q &= [P_Q x]^T (x - x_Q) = \\ &= [(I - A_q^T (A_q A_q^T)^{-1} A_q) x]^T [A_q^T (A_q A_q^T)^{-1} A_q x] = 0, \end{aligned} \quad (3.121)$$

т. е. действительно выполняется (3.110).

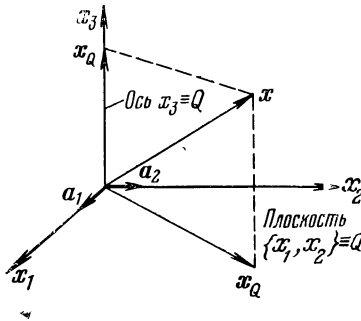


Рис. 3.2. Проекция в трехмерном пространстве.

Матрица проектирования, как нетрудно проверить, обладает свойствами:

$$1. \quad P_Q^T = P_Q, \quad (3.122)$$

$$2. \quad P_Q P_Q = P_Q. \quad (3.123)$$

Пример. Рассмотрим трехмерное пространство — рис. 3.2 ($n=3$).

Пусть ограничения, соответствующие (3.106), заданы в виде

$$x_1 = 0, \quad x_2 = 0, \quad (3.124)$$

или

$$A_2 x = 0, \quad x = [x_1, x_2, x_3]^T, \quad q=2,$$

где

$$A_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

Пересечением двух многообразий, заданных в (3.124), является ось x_3 . На векторы a_1 и a_2 , где

$$a_1^T = [1 \ 0 \ 0] \quad \text{и} \quad a_2^T = [0 \ 1 \ 0],$$

натянута все двумерное пространство Q , которым является плоскость $x_1 - x_2$. Матрица проектирования равна в этом случае

$$P_2 = A_2^T (A_2 A_2^T)^{-1} A_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$P_2 = I_3 - P_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Теперь можно вернуться к решению задачи нелинейного программирования. Допустим, что по окончании i -й итерации была достигнута точка x_i , в которой удовлетворяется следующее множество ограничений:

$$a_j x_i = b_j, \quad j \in J. \quad (3.125)$$

Другими словами, точка x_i принадлежит пересечению q гиперплоскостей, заданных уравнениями (3.125). Как и при решении любой задачи, должно быть определено направление r_i для следующего шага. Главная идея метода Розена заключается в том, что r_i берется в направлении проекции градиента ∇f_i на Q — подпространство, задаваемое уравнениями $a_j^T x = 0, j \in J$. Таким образом,

$$r_i = \alpha P_Q \nabla f_i, \quad (3.126)$$

где α — коэффициент пропорциональности и P_Q определяется уравнением (3.114). Выражение $P_Q \nabla f_i$ представляет собой так называемую *проекцию градиента*. Длина шага d_i выбирается обычным образом из условий, что $f(x)$ должно продолжать увеличиваться вдоль траектории и что траектория должна оставаться внутри допустимой области. Если проекция градиента $P_Q \nabla f_i$ равна нулю, то, как показывает (3.120), это означает, что градиент ∇f_i принадлежит многообразию Q , натянутому на векторы $a_j, j \in J$. Следовательно, ∇f_i может быть пред-

ставлен в виде линейной комбинации базисных векторов

$$\nabla f_i = \sum_{j \in J} u_j a_j, \quad (3.127)$$

где u_j — скалярные коэффициенты.

Напомним, что Q и Q ортогональны. В силу этого, если $u_j \geq 0$ для всех $j \in J$, то удовлетворяются условия Куна — Таккера (2.31) и (2.32) и, следовательно, x_i является оптимальной точкой. Если некоторые из u_j отрицательны, то из многообразия надо исключить связанные с ними ограничения, т. е. соответствующие a_j , и продолжать действовать по-прежнему.

В случае нелинейных ограничений берут проекцию градиента на многообразии, образованное касательными гиперплоскостями в точке x_i . Чтобы обеспечить допустимость, осуществляется процедура коррекции в направлении на допустимую область. Обычно она состоит из значительного числа итераций. С подробностями читатель может познакомиться в [18].

Методы последовательной минимизации без ограничений

Метод последовательной минимизации без ограничений (часто называемый методом штрафных функций) был предложен Кэрролом [19] и позднее был строго обоснован и развит Фиакко и Мак-Кормиком [20]. Одно из основных достоинств этого метода заключается в его эффективности при нелинейных целевых функциях и ограничениях как в форме равенств, так и неравенств. Выпуклость или вогнутость функций не является необходимыми условиями применения этого метода. Однако, когда эти условия не выполняются, алгоритм может сходиться к локальному, а не к глобальному экстремуму.

Рассматриваемая задача не отличается от общей задачи математического программирования, сформулированной в (2.6). Для удобства повторим ее здесь:

$$\begin{aligned} \min \{f(x) \mid g_i(x) \geq 0, i=1, \dots, q; \\ h_j(x) = 0, j=1, \dots, p\}. \end{aligned} \quad (3.128)$$

Основная идея метода последовательной минимизации без ограничений заключается в преобразовании за-

дачи математического программирования с ограничениями вида (3.128) в ряд задач минимизации без ограничений, путем учета ограничений с помощью некоторых штрафных функций. С этой целью вводится модернизированная целевая функция вида

$$P(\mathbf{x}, r) = f(\mathbf{x}) - r \sum_{i=1}^q \ln g_i(\mathbf{x}) + r^{-1} \sum_{j=1}^p h_j^2(\mathbf{x}), \quad (3.129)$$

где r — весовой коэффициент, а $r^{-1} = 1/r$. Алгоритм работает следующим образом:

1. Выбирается начальная точка \mathbf{x}_0 так, чтобы выполнялось условие

$$g_i(\mathbf{x}_0) > 0, \quad i=1, \dots, q. \quad (3.130)$$

2. Выбирается начальное значение r , например $r_0 = 1$.

Величина r_0 не играет существенной роли. Практически можно начинать процесс при любом r_0 , лишь бы последние два члена в (3.129) не были слишком малы.

3. Двигаясь из точки \mathbf{x}_0 в направлении уменьшения значения $P(\mathbf{x}, r_0)$, минимизируют функцию $P(\mathbf{x}, r_0)$. Допустим, что минимум $P(\mathbf{x}, r_0)$ достигается в точке \mathbf{x}_1 . Для минимизации без ограничений может быть использован ряд методов. Некоторые из этих методов будут рассмотрены в параграфе 3.4.

4. Следующим образом вычисляется новое значение r :

$$r_1 = \frac{r_0}{K}, \quad K > 1. \quad (3.131)$$

Обычно K выбирают порядка 4 или 5; выбор этот не критичен. Опыт показал, что слишком большие значения K , например $K > 10$, не дают преимуществ, так как при этом последние члены $P(\mathbf{x}, r)$ уменьшаются слишком рано. Подставляют в (3.129) новый коэффициент r_1 и, исходя из точки \mathbf{x}_1 , минимизируют функцию $P(\mathbf{x}, r_1)$.

5. Повторяют шаг 4 для все меньших и меньших значений r , определяемых по формуле

$$r_{k+1} = \frac{r_k}{K}. \quad (3.132)$$

Вычисление заканчивается, когда будет выполнен критерий окончательной сходимости, т. е. при

$$|P(\mathbf{x}, r_{h-1}) - P(\mathbf{x}, r_h)| < \varepsilon, \quad \varepsilon > 0, \quad (3.133)$$

где ε — заданное малое число, зависящее от требуемой точности.

Когда r становится очень малым, и ограничения задачи удовлетворяются, то, как видно из (3.129),

$$P_{\min}(\mathbf{x}, r) = f_{\min}(\mathbf{x}) + \delta, \quad (3.134)$$

где δ — число, которое можно сделать сколь угодно малым, задавая тем самым необходимый, достаточно жесткий, критерий сходимости. Справедливость равенства (3.134) можно пояснить следующим образом.

Как уже указывалось, алгоритм ведет вычисления по точкам \mathbf{x}_k , лежащим в области $g_i(\mathbf{x}) > 0, i=1, \dots, q$. Если хотя бы одно из ограничений $g_i(\mathbf{x})$ окажется слишком близким к границе допустимости, т. е. если $g_i(\mathbf{x}) \rightarrow 0$, то, как видно из (3.129), $P(\mathbf{x}, r) \rightarrow \infty$, что противоречит требованию минимизации. Поэтому, пока $P(\mathbf{x}, r)$ уменьшается, ограничения в форме неравенств удовлетворяются автоматически. Соответственно, когда r стремится к нулю, второй член правой части равенства (3.129) будет мал. С другой стороны, когда r уменьшается, r^{-1} становится очень большим. При минимизации $P(\mathbf{x}, r)$ положительные функции $h_j^2(\mathbf{x})$, естественно, стремятся к нулю, чтобы соответственно удовлетворить также ограничениями в форме равенств $h_j(\mathbf{x}) = 0, j=1, \dots, p$. Таким образом, видно, что последовательная минимизация функции $P(\mathbf{x}, r)$ при уменьшающихся значениях r , естественно, должна приводить к сходимости функции $f(\mathbf{x})$ к минимуму при всех заданных ограничениях.

Сходимость метода для случая, когда $f(\mathbf{x})$ и все $h_j^2(\mathbf{x})$ выпуклые, а все $g_i(\mathbf{x})$ вогнутые, была строго доказана Фиакко и Мак-Кормиком [20]. Ими же была разработана эффективная программа для ЦВМ.

3.4. Методы минимизации без ограничений

Введение

Некоторые из методов, описанных в 3.3, и, в частности, метод последовательной минимизации без ограничений требуют решения большого числа промежуточных задач отыскания экстремума без ограничений. Одна из возможностей для осуществления таких решений заклю-

чается в применении прямого поиска. Существует много эффективных методов поиска [21; 22 гл. 6]. Однако при применении этих методов к задачам с большим числом переменных возникают значительные трудности. Намного лучшие результаты дает применение методов, использующих градиентный подход, рассмотренный в 3.3. Поэтому в настоящем параграфе этим методам будет уделено основное внимание. Различные методы поиска не рассматриваются, так как существует ряд превосходных книг, посвященный этим вопросам [21, 22 гл. 6].

Методы переменной метрики

Методы переменной метрики, принадлежащие к типу градиентных методов, разрабатывались многими авторами. Исчерпывающий обзор этих методов был опубликован Пирсоном [14] и ряд вычислительных оценок получен другими авторами [14, 23—25, 32—36].

Нами здесь будут использованы обозначения, введенные в 3.3, в связи с изложением градиентного подхода. Дополнительно надо определить следующие величины:

s_i — вектор фактического шага из точки x_i :

$$s_i = x_{i+1} - x_i = d_i r_i = d_i H_i \nabla f_i, \quad (3.135)$$

y_i — изменение градиента при переходе из точки x_i в точку x_{i+1} :

$$y_i = \nabla f_{i+1} - \nabla f_i, \quad (3.136)$$

$S_i = [s_0, s_1, \dots, s_{i-1}]$ — матрица размера $n \times i$, столбцами которой являются векторы шагов s_0, s_1, \dots, s_{i-1} ,

$Y_i = [y_0, y_1, \dots, y_{i-1}]$ — матрица размера $n \times i$, столбцами которой являются векторы y_0, y_1, \dots, y_{i-1} .

Как указывалось в 3.3, матрица H_i , называемая *метрикой*, устанавливает зависимость вектора направления r_i от градиента ∇f_i (см. (3.73) и (3.135)). Например, в методе наискорейшего подъема $H_i = \frac{1}{|\nabla f_i|}$ при всех i .

В методах переменной метрики H_i изменяется на каждом шаге. Различные методы отличаются друг от друга алгоритмом, согласно которому изменяется H_i . После того как установлены новые значения H_i и соответствен-

но r_i , длина шага d_i определяется с помощью одномерной минимизации (или максимизации):

$$f(\mathbf{x}_{i+1}) = \min_d [f(\mathbf{x}_i + d\mathbf{r}_i)]. \quad (3.137)$$

Одномерная задача, сформулированная в (3.137), может быть решена одним из методов поиска [21, 22]. Ниже перечисляются рекуррентные формулы для H_i , используемые в различных методах.

1. Обобщенный метод Ньютона [23]:

$$H_i = [A(\mathbf{x}_i)]^{-1}, \quad (3.138)$$

где $A(\mathbf{x})$ — гессиан—матрица, равная

$$A(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \vdots & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}. \quad (3.139)$$

Этот метод является наиболее эффективным, обеспечивающим быструю сходимость в задачах широкого класса. Однако могут возникнуть трудности, когда в задаче минимизации матрица A неположительно определена или A^{-1} не существует; Фиакко и Мак-Кормик предложили модернизацию этого метода, которая позволяет обойти некоторые из упомянутых трудностей [20, стр. 166—167].

Метод Ньютона относится к классу так называемых градиентных методов второго порядка. Хотя при применении этого метода возникают свои трудности и требуется больший объем памяти ЦВМ, чем при методах первого порядка, где используются лишь производные первого порядка, тем не менее для определенного класса задач он оказывается наиболее эффективным [25].

2. Алгоритм проектирования [14, 16, 23].

$$\left. \begin{aligned} H_0 &= R, \\ H_{i+1} &= H_i - \frac{(H_i y_i)(H_i y_i)^T}{y_i^T H_i y_i}, \end{aligned} \right\} \quad (3.140)$$

где R — симметричная положительно определенная матрица (например, $R=I$). Через каждые n шагов для H_i восстанавливается значение R .

3. Метод Флетчера — Пауэлла — Давидона. Этот метод, впервые предложенный Давидоном [26] и модифицированный Флетчером и Пауэллом [27]; использует следующую рекуррентную формулу:

$$\left. \begin{aligned} H_0 &= R, \\ H_{i+1} &= H_i - \frac{(H_i y_i)(H_i y_i)^T}{y_i^T H_i y_i} + \frac{s_i s_i^T}{s_i^T y_i}, \end{aligned} \right\} \quad (3.141)$$

где \bar{R} — симметричная положительно определенная матрица. Мак-Кормик и Пирсон [23] ввели вариант, в котором матрица H_i через каждые $n+1$ шагов берется равной R . Решение задачи при этом ускоряется в 4—5 раз [23]. Интересно отметить, что при $i \rightarrow \infty$ $H_i \rightarrow [A(x_i)]^{-1}$, т. е. к гессиону, используемому в методе Ньютона. Поэтому метод можно назвать квазиньютоновским.

Как алгоритм проектирования, так и алгоритм Флетчера—Пауэлла—Давидона принадлежат общему классу методов сопряженных направлений [28]. Направления r_0, r_1, \dots, r_{n-1} называются сопряженными относительно положительно определенной симметричной матрицы A , если

$$r_i^T A r_j = 0, \quad 0 \leq i \neq j \leq n-1. \quad (3.142)$$

К тому же классу относится и так называемый метод сопряженных градиентов.

Алгоритмы и сообщения об экспериментальных вычислениях с применением методов сопряженных градиентов можно найти в ряде источников [23, 24, 29—31, 37]. Описание иных алгоритмов с переменной метрикой и некоторых экспериментальных результатов и применений можно найти в работах [14, 20, 23—25, 29, 32—36].

Исчерпывающее изложение различных методов минимизации без ограничений можно найти в книге Фиакко и Мак-Кормика [20, гл. 7 и 8].

Сравнительное рассмотрение [25]. Для иллюстрации относительной эффективности различных методов минимизации без ограничений ниже приводится численный

пример. Методы минимизации без ограничений применялись совместно с методом штрафных функций (см. 3.3).

Решалась задача нелинейного программирования, сформулированная для системы четвертого порядка, которая рассматривается в 6.4 — формулы от (6.41) до (6.44).

Т а б л и ц а 3.3

[14, 25]*)

Метод	Формула	Число итераций	Длительность вычислений (сек)
1. Ньютона	$H_i = A^{-1}(x_i)$	61	1,40
2. Проектирования	$H_{i+1} = H_i + \frac{(H_i y_i)(H_i y_i)^T}{y_i^T H_i y_i}$	156	2,93
3. Несимметричный переменный метрики 1	$H_{i+1} = H_i + \frac{(s_i - H_i y_i)(H_i^T y_i)^T}{y_i^T H_i y_i}$	344	6,13
4. Модифицированный Флетчера — Пауэлла	$H_{i+1} = H_i + \frac{(s_i - H_i y_i)(s_i - H_i y_i)^T}{y_i^T (s_i - H_i y_i)}$	354	6,63
5. Флетчера — Ривза	$r_{i+1} = -\nabla f_{i+1} + r_i \frac{\nabla f_{i+1}^T \nabla f_{i+1}}{\nabla f_i^T \nabla f_i}$	399	7,66
6. Несимметричный переменный метрики 2	$H_{i+1} = H_i + \frac{(s_i - H_i y_i) s_i^T}{s_i^T y_i}$	580	9,17
7. Флетчера — Пауэлла — Давидона	$H_{i+1} = H_i - \frac{(H_i y_i)(H_i y_i)^T}{y_i^T H_i y_i} + \frac{s_i y_i^T}{s_i^T y_i}$	2015	32,24
8. Наилейшего спуска	$H_i = \frac{I_n}{\ \nabla f\ }$	4141	57,58

*) В большинстве методов $H_0 = I_n$ и H_i берется равным H_0 через каждые $n-1$ или n , или $n+1$ шагов.

Приняты следующие обозначения переменных:

$$\begin{array}{llll} x & x_2, & \omega_{n1} & x_4, \\ q & x_2, & \delta_2 & x_5, \\ \delta_1 & x_3, & \omega_{n2} & x_6. \end{array}$$

При этом задача нелинейного программирования формулируется следующим образом: минимизировать

$$J = cx_1 + x_2 + x_5$$

при следующих ограничениях:

$$\begin{aligned} & (2x_3^2 - 1)x_4^2 + (2x_5^2 - 1)x_6^2 \geq 0, \\ x_4 [4x_4^2x_6^2(4x_3^2x_5^2 - 2x_3^2 - 2x_5^2 + 1) + x_4^2 + x_6^2] + \\ & \quad + 1 - 2x_2(x_3x_4 + x_5x_6) \geq 0, \\ 2x_1x_4^2x_6^2 [(2x_3^2 - 1)x_6^2 + (2x_5^2 - 1)x_4^2] + \\ & \quad + 2x_6x_4x_2(x_2x_6 + x_5x_4) - x_4^2 - x_6^2 - 4x_3x_5x_4x_6 \geq 0, \\ & x_{3\min} \leq x_3 \leq x_{3\max}, 0 \leq x_4 \leq x_{4\max}, \\ & x_{5\min} \leq x_5 \leq x_{5\max}, 0 \leq x_6 \leq x_{6\max}. \end{aligned}$$

Заметим, что целевая функция линейна, а некоторые ограничения резко нелинейны. Вычисления производились на вычислительной системе ИБМ 360/91. Результаты вычислений приведены в табл. 3.3.

Следует подчеркнуть, что эти результаты нельзя трактовать как имеющие общий характер. Они означают лишь, что для данного частного типа задачи метод Ньютона оказался наиболее эффективным. Для другой задачи результат может быть иным.

Литература

1. C. V. Dantzig, Linear Programming and Extensions. Princeton University Press, Princeton, N. J., 1963. [Русский перевод: Дж. Данциг, Линейное программирование, его применения и обобщения, «Прогресс», 1966.]
2. G. Hadley, Linear Programming, Addison-Wesley, Reading, Mass., 1962.
3. M. Simonnard, Linear Programming, Prentice-Hall, Englewood Cliffs, N. J., 1966.
4. S. I. Gass, Linear Programming, 3rd ed., McGraw-Hill, New York, 1969. [Русский перевод: С. Гасс, Линейное программирование, Физматгиз, 1961.]

5. W. Llewellyn, *Linear Programming*, Holt, Rinehart and Winston, New York, 1964.
6. F. A. Valentine. *Convex Sets*, McGraw-Hill, New York, 1964.
7. G. Hadley, *Nonlinear and Dynamic Programming*, Addison-Wesley, Reading, Mass, 1964, chap. 7. [Русский перевод: Дж. Хедли, *Нелинейное и динамическое программирование*, «Мир», 1967.]
8. J. Abadie, ed., *Nonlinear Programming*, Wiley, New York, 1967, chap. VII by E. M. L. Beale.
9. H. P. Kunzi, W. Krelle and W. Oettli. *Nonlinear Programming*, Blaisdell, Waltham, Mass., 1966, chaps. 4—10. [Русский перевод: Г. П. Кюнзи, В. Крелле. *Нелинейное программирование*, «Советское Радио», 1965.]
10. J. C. G. Boot, *Quadratic Programming*, North-Holland, Amsterdam, 1964.
11. H. P. Kunzi, H. G. Tzsachach and C. A. Zehnder, *Numerical Methods of Mathematical Optimization*, Academic Press, New York, 1968.
12. P. Wolfe, *The Simplex Method for Quadratic Programming*, *Econometrica* 27, pp. 382—398 (1959).
13. M. Frank and P. Wolfe, *An Algorithm for Quadratic Programming*, *Naval Res. Logistics Quart.* 3, pp. 95—110 (1956).
14. J. D. Pearson, *On Variable Metric Methods of Minimization*, *Computer J.* 11, pp. 171—178 (1969).
15. G. Zoutendijk, *Nonlinear Programming: A Numerical Survey*, *J. SIAM Control* 4, pp. 194—210 (1966).
16. G. Zoutendijk, *Methods of Feasible Directions*, American Elsevier, New York, 1960. [Русский перевод: Г. Зойтендейк, *Методы возможных направлений*, ИЛ, 1963.]
17. J. B. Rosen, *The Gradient Projection Method for Nonlinear Programming, Part I, Linear Constraints*, *J. SIAM* 8, pp. 181—217 (1960).
18. J. B. Rosen, *The Gradient Projection Method for Nonlinear Programming, Part II, Nonlinear Constraints*, *J. SIAM* 9, pp. 514—532 (1961).
19. C. W. Carroll, *The Created Response Surface Technique Optimizing Nonlinear Restrained Systems*, *Operations Res.* 9, pp. 169—184 (1964).
20. A. V. Fiacco and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. Wiley, New York, 1968. [Русский перевод: А. Фиакко и Г. Мак-Кормик, *Нелинейное программирование. Методы последовательной безусловной минимизации*, «Мир», 1972.]
21. D. J. Wilde, *Optimum Seeking Methods*, Prentice-Hall, Englewood Cliffs, N. J., 1964. [Русский перевод: Д. Дж. Уайлд, *Методы поиска экстремума*, «Наука», 1967.]
22. D. J. Wilde and C. S. Beightler, *Foundation of Optimization*, Prentice-Hall, Englewood Cliffs, N. J., 1967.
23. G. P. McCormick and J. D. Pearson, *Variable Metric Methods and Unconstrained Optimization*. Joint Conference on Optimization, University of Keele, Staffordshire, England, March 1968,

24. G. E. Meyers, Properties of the Conjugate Gradient and Davidson Methods, *J. Optimization Theory Applications* **2**, pp. 209—219 (1968).
25. D. Tabak, Comparative Study of Various Minimization Techniques Used in Mathematical Programming, *IEEE Trans. Automatic Control* **AC-14**, p. 572 (1969).
26. W. C. Davidson, Variable Metric Method of Minimization, Report ANL-5990 (rev), Argonne National Laboratory, Argonne, Ill., 1959.
27. R. Fletcher and M. J. D. Powell, A Rapidly Convergent Descent Method for Minimization, *Computer J.* **6**, pp. 163—168 (1963).
28. M. R. Hestenes, The Conjugate Gradient Method for Solving Linear Systems, *Proceedings of the Symposium on Applied Mathematics*, vol. VI, McGraw-Hill, New York, 1956, pp. 83—102.
29. H. J. Kelley and G. E. Meyers, Conjugate Direction Methods for Parameter Optimization, *Astronautica Acta* (в печати).
30. L. S. Lasdon, S. K. Mitter and A. D. Waren, The conjugate Gradient Method for Optimal Control Problems, *IEEE Trans. Automatic Control* **AC-12**, pp. 132—138 (April 1967).
31. S. S. Tripathi and K. S. Narendra, Conjugate Direction Methods for Nonlinear Optimization Problems, *National Electronics Conference*, Chicago, Dec. 1968.
- 32*. Ю. М. Данилин, Методы сопряженных направлений для решения задач минимизации, *Кибернетика*, № 5 (1971), 122—136.
- 33*. J. W. Daniel, *The approximate minimization of functionals*, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1971.
- 34*. M. J. D. Powell, A survey on numerical methods for unconstrained optimization, *SIAM Review* **2**, № 1 (1970).
- 35*. M. J. D. Powell, Recent advances in unconstrained optimization, *Mathem. Progr.*, **1**, № 1 (1971).
- 36*. H. P. Kunzi, W. Oettli. *Nichtlineare Optimierung: Neuere Verfahren Bibliographie*, Springer, Berlin, Heidelberg, N. Y., 1969.
- 37*. Б. Т. Поляк, Метод сопряженных градиентов в задачах на экстремум, *Журнал вычислительной математики и математической физики* **9**, № 4 (1969), 807—821.

**ОПТИМАЛЬНОЕ УПРАВЛЕНИЕ
И МАТЕМАТИЧЕСКОЕ ПРОГРАММИРОВАНИЕ****4.1. Введение**

В гл. 2 были рассмотрены основные понятия математического программирования, а в гл. 3 был приведен обзор алгоритмов для численного решения задач математического программирования.

При совместном рассмотрении методов математического программирования и задач оптимального управления возникает ряд вопросов.

1. Какова связь между математическим программированием и задачей оптимального управления?

2. Как можно использовать вычислительные алгоритмы математического программирования для эффективно численного решения задач оптимального управления?

Цель этой главы заключается в том, чтобы наметить первые шаги при попытке ответить на эти вопросы. Различным аспектам оптимального управления посвящено большое число работ. Лишь несколько источников было выбрано для перечня, помещенного в конце главы [1—3].

Чтобы при рассмотрении задач оптимального управления иметь возможность сопоставления, в 4.2 сформулирована одна из таких задач. Эта формулировка будет служить основой рассмотрения как в остальной части данной главы, так и в последующих главах.

В 4.3 делается попытка дать ответ на вопрос 1. Не входя в дальнейшие детали, можно рассматривать этот вопрос следующим образом.

Задачи математического программирования — как и оптимального управления — являются задачами оптимизации. Другими словами, они являются задачами отыскания экстремума функции многих переменных, подчиненных различным условиям или ограничениям. Естественно, следует ожидать, что между ними существует определенная связь. Эта связь показана в 4.3.

Попытка рассмотреть вопрос 2 или, вернее, только некоторые его аспекты делается в 4.4.

Здесь обсуждаются некоторые общие соображения, которые возникают в связи с применением математиче-

ского программирования к численному решению задач оптимального управления. Более детально это обсуждение будет продолжено в последующих главах, где рассмотрены также иллюстрирующие примеры вычислений и описаны различные технические применения.

4.2. Формулировка задачи

Вначале будет рассмотрена задача оптимального управления [1—3] для динамических систем с непрерывным временем. Различные частные случаи обсуждаются в последующих главах. Предполагается, что рассматриваемая система является неавтономной, нелинейной и имеет сосредоточенные параметры. Динамическое поведение этой системы может быть описано системой нелинейных обыкновенных дифференциальных уравнений первого порядка:

$$\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, \mathbf{u}, t), \quad (4.1)$$

где \mathbf{y} — n -мерный вектор состояния $[y_1, y_2, \dots, y_n]^T$, \mathbf{u} — m -мерный управляющий вектор $[u_1, u_2, \dots, u_m]^T$, t — время, \mathbf{f} — n -мерная нелинейная векторная функция, $\dot{\mathbf{y}} = \frac{d\mathbf{y}}{dt}$.

Предположим, что в некоторый начальный момент t_0 система находится в заданном состоянии

$$\mathbf{y}(t_0) = \mathbf{y}_0. \quad (4.2)$$

Такое предположение справедливо для большинства практических случаев. Однако возможны случаи, когда начальное состояние является неизвестной переменной. В этих случаях, как правило, бывает задано некоторое промежуточное состояние или конечное состояние; это не меняет в основном подход к задаче, поскольку обычно можно решить задачу, используя обратное время, изменяющееся от конечного момента к начальному. В некоторых случаях вектор состояния вообще не задается предварительно ни для одного момента времени.

В только что описанной системе цель управления заключается в том, чтобы из начального состояния привести систему к моменту $t = t_f$ в новое состояние, расположенное внутри области, определяемой условием

$$\mathbf{s}(\mathbf{y}_f) \geq 0, \quad (4.3)$$

где s есть k -мерная нелинейная векторная функция, которая определяет конечную область. Конечное состояние, в общем случае не фиксированное (свободное), обозначается следующим образом:

$$y_f = y(t_f). \quad (4.4)$$

В общем случае (4.3) определяет замкнутую область, ограниченную поверхностями

$$s_i(y_j) = 0. \quad (4.5)$$

Конечное состояние y_f может лежать как внутри области, определяемой неравенством (4.3), так и на ее границе, определяемой уравнениями (4.5). В частности, конечная область может состоять лишь из одной точки. Момент окончания процесса t_f может быть фиксированным или свободным.

Управление, которое переводит систему из состояния y_0 в состояние y_f , должно осуществляться таким образом, чтобы заданный показатель качества был экстремальным. Выражение для показателя качества в общем виде можно записать следующим образом:

$$J = \int_{t_0}^{t_f} F[y(t), u(t), t] dt + G(y_f), \quad (4.6)$$

где F и G — нелинейные функции переменных состояния и управления. Как видно из (4.6), показатель качества может зависеть от траектории и управления, или от конечного состояния, либо от того и другого. Особый интерес представляет задача управления, оптимального по быстродействию, когда $F \equiv 1$ и $G = 0$, т. е.

$$J = \int_{t_0}^{t_f} dt = (t_f - t_0), \quad (4.7)$$

где t_f — переменная.

В большинстве практических случаев на переменные состояния и управления могут быть наложены дополнительные ограничения. Обычно эти ограничения могут быть выражены системой неравенств или равенств следующего вида:

$$g(y, u) \geq 0, \quad (4.8)$$

$$h(y, u) = 0, \quad (4.9)$$

где \mathbf{g} и \mathbf{h} в общем случае являются нелинейными векторными функциями. Например, очень часто ограничены абсолютные значения переменных состояния и управления, т. е. наложены действующие в течение всего времени следующие условия в форме неравенств:

$$\left. \begin{aligned} u_{i \max} - |u_i| &\geq 0, & i = 1, \dots, m, \\ y_{i \max} - |y_i| &\geq 0, & i = 1, \dots, n, \end{aligned} \right\} \quad (4.10)$$

где $u_{i \max}$ и $y_{i \max}$ — заданные скалярные величины. Понятно, что неравенства (4.10) представляют собой одну из простейших форм ограничений, которые могут быть наложены на переменные состояния и управления.

Задача оптимального управления с непрерывным временем теперь может быть сформулирована следующим образом.

Найти вектор оптимального управления $\mathbf{u}(t)$, $t_0 \leq t \leq t_f$, который переводит систему из начального состояния \mathbf{y}_0 в конечную область \mathbf{s} так, что при этом обеспечивается экстремум показателя качества (4.6) и удовлетворяются динамические уравнения системы (4.1), условия для конечной области (4.3) и ограничения (4.8) и (4.9).

Иначе это можно записать в виде

$$\min (\text{или } \max) \left\{ \int_{t_0}^{t_f} F(\mathbf{y}, \mathbf{u}, t) dt + G(\mathbf{y}_f) \right\} \dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, \mathbf{u}, t);$$

$$\mathbf{y}(t_0) = \mathbf{y}_0; \mathbf{s}(\mathbf{y}_f) \geq 0; \mathbf{g}(\mathbf{y}, \mathbf{u}) \geq 0; \mathbf{h}(\mathbf{y}, \mathbf{u}) = 0 \}. \quad (4.11)$$

В дальнейшем рассматривается связь между задачей оптимального управления, определяемой формулой (4.11), и задачей математического программирования, изложенной в гл. 2.

4.3. Математическое программирование и оптимальное управление

Между задачей математического программирования, изложенной в гл. 2, и задачей оптимального управления, изложенной в 4.2, существует совершенно очевидная связь. Задача оптимального управления фактически яв-

ляется задачей математического программирования. Это отмечалось рядом авторов, например [4, стр. XV; 21—25]. Чтобы пояснить это, надо задачу оптимального управления, сформулированную в (4.11), представить несколько иначе.

Прежде всего, рассмотрим интеграл в выражении (4.6) для показателя качества.

Интеграл можно представить как предел суммы [5]. Чтобы осуществить это для интеграла в (4.6), необходимо разбить интервал времени от t_0 до t_f на N подынтервалов $(t_1 - t_0)$, $(t_2 - t_1)$, ..., $(t_N - t_{N-1})$, где $t_j = t_N$. В пределах каждого интервала $t_k - t_{k-1}$ ($k=1, \dots, N$) можно выделить произвольное t'_k , для которого $t_{k-1} \leq t'_k \leq t_k$. Используя такое разбиение, интеграл (4.6) можно записать в виде

$$\int_{t_0}^{t_f} F[\mathbf{y}(t), \mathbf{u}(t), t] dt = \lim_{N \rightarrow \infty} \sum_{k=1}^N F[\mathbf{y}(t'_k), \mathbf{u}(t'_k), t'_k] (t_k - t_{k-1}). \quad (4.12)$$

Момент окончания процесса t_f , понятно, конечен, но общее число N временных подынтервалов стремится к бесконечности.

Уравнения состояний (4.1) при таком разбиении времени можно записать в виде

$$\lim_{h_k \rightarrow 0} \frac{1}{h_k} [\mathbf{y}(t_k) - \mathbf{y}(t_{k-1})] = \mathbf{f}[\mathbf{y}(t_{k-1}), \mathbf{u}(t_{k-1}), t_{k-1}], \quad (4.13)$$

где

$$h_k = t_k - t_{k-1}, \quad k=1, \dots, N; \quad N \rightarrow \infty. \quad (4.14)$$

Аналогично, можно переписать дополнительные ограничения (4.8) и (4.9) в виде

$$\left. \begin{aligned} \mathbf{g}[\mathbf{y}(t_k), \mathbf{u}(t_k)] &\geq 0, \\ \mathbf{h}[\mathbf{y}(t_k), \mathbf{u}(t_k)] &= 0. \end{aligned} \right\} \quad (4.15)$$

Условия окончания (4.3) и второй член $G(\mathbf{y}_f)$ выражения (4.6) для показателя качества является функциями только одной точки в пространстве состояний и поэтому остаются неизменными. Теперь можно объединить

уравнения от (4.12) до (4.15) в формулировку задачи математического программирования, аналогичную данной в (4.11):

$$\begin{aligned} \min (\text{или max}) \left\{ \lim_{N \rightarrow \infty} \sum_{k=1}^N F [y (t'_k), u (t'_k), t'_k] (t_k - t_{k-1}) + \right. \\ \left. + G [y (t_f)] \lim_{h_k \rightarrow 0} \frac{1}{h_k} [y (t_k) - y (t_{k-1})] = \right. \\ \left. = f [y (t_{k-1}), u (t_{k-1}), t_{k-1}], \right. \\ \left. \begin{array}{l} h_k = t_k - t_{k-1}, k = 1, \dots, N; N \rightarrow \infty; s [y (t_N)] \geq 0; \\ g [y (t_k), u (t_k)] \geq 0, \quad h [y (t_k), u (t_k)] = 0, \\ k = 0, \dots, N; N \rightarrow \infty. \end{array} \right\} \end{aligned} \quad (4.16)$$

Легко видеть, что (4.16) действительно является формулировкой задачи математического программирования, неизвестными переменными которой являются

$$\begin{aligned} y (t_k), y (t'_k), \quad k = 1, \dots, N; \quad u (t_k), u (t'_k), \\ k = 0, 1, \dots, N - 1; \quad N \rightarrow \infty. \end{aligned}$$

Эта задача математического программирования отличается от рассмотренной в гл. 2 тем, что в задаче (4.16) число переменных бесконечно, в то время как обычная задача математического программирования имеет конечное число переменных. Другими словами, *задача оптимального управления* для непрерывной системы образует бесконечномерную задачу математического программирования в бесконечномерном пространстве.

Гурвиц [6] распространил теорему Куна — Таккера на задачи в бесконечномерных пространствах. Позднее эта идея была распространена на теорию оптимального управления другими авторами [7—11, 26—29]. В указанных работах была установлена связь необходимых и достаточных условий для задач оптимального управления с условиями Куна — Таккера — Гурвица.

Поскольку задача оптимального управления действительно является задачей математического программирования, ясно, что должна существовать непосредственная связь между обычно применяемой трактовкой задач

оптимального управления и методами, применяемыми в математическом программировании. Одним из наиболее популярных методов оптимального управления является принцип максимума Понтрягина [12].

Халкин и Нейштадт [13, 14] сформулировали обобщенный принцип максимума, который дает необходимые условия экстремума для общих задач оптимизации. Они показали, что условия Куна — Таккера как для конечномерного, так и бесконечномерного пространств могут быть получены из их условий. Аналогичные работы были опубликованы другими авторами. Например, Хензон [15] показал, что некоторые формы задач математического программирования имеют аналоги в оптимальном управлении. Хо и Брентани [16] исследовали задачи оптимального управления с ограничениями в форме неравенств. Они доказали достаточность своих условий, применив условия Куна — Таккера для седловой точки и тем самым показав связь между задачей оптимального управления и методами математического программирования.

Изящный путь единообразного представления различных условий экстремума недавно предложили Уайлд и Бейтлер [17]. Эти авторы осуществили унификацию, введя некоторые специальные понятия, которые можно описать следующим образом. Рассматривается задача оптимизации с n переменными и m ($m < n$) ограничениями в форме равенств. Из n переменных выбирается m переменных, которые рассматриваются как «переменные состояния». Переменные состояния выражаются в виде функций остальных $n - m$ переменных, которые являются «решающими переменными». Введено важное понятие связанной производной параметра качества по решающим переменным в допустимой области. Аналогично, были введены коэффициенты чувствительности, которые являются частными производными оптимального параметра качества по отношению к возмущениям в ограничениях в форме равенств. Далее, было введено понятие свободных производных. Свободные производные в принципе представляют собой то же, что и коэффициенты чувствительности. Различие между ними в том, что свободные производные определены в каждой точке допустимой области, в то время как коэффициенты чувствитель-

ности определены только в окрестности экстремума. Используя эти понятия, авторы получили как условия Куна — Таккера, так и принцип максимума, показав тем самым их тесную связь между собой.

4.4. Соображения о вычислениях

Как известно, при численном решении непрерывных дифференциальных уравнений оперируют с дискретными значениями независимых переменных. Так же поступают с задачами, относящимися к непрерывным системам оптимального управления. Независимо от применяемого метода или способа, чтобы получить возможность решения на ЦВМ дифференциальных уравнений задачи и интеграла, определяющего показатель качества, используется дискретное время. Другими словами, дифференциальные уравнения заменяют разностными уравнениями, а интегрирование — конечным суммированием. Когда в задаче математического программирования, порожденной задачей оптимального управления, дело доходит до практических вычислений, может оказаться, что число переменных хотя и конечно, но чрезмерно велико. Во многих случаях это число может быть уменьшено до практически приемлемой величины. Такие случаи и будут рассмотрены в следующих главах.

Задача оптимального управления, сформулированная в 4.2, теперь будет переформулирована в дискретной форме [18]. Весь временной интервал от начального момента t_0 до конечного t_f делится на N , в общем случае неравных, временных интервалов. Рассматриваемые дискретные моменты времени обозначены следующим образом: $t_0, t_1, t_2, \dots, t_{N-1}, t_N = t_f$, а временные интервалы между ними T_1, T_2, \dots, T_N . Таким образом,

$$T_i = t_i - t_{i-1} \quad (4.17)$$

и

$$\sum_{i=1}^N T_i = T = t_f - t_0. \quad (4.18)$$

Обычно T , как и все T_i , априорно неизвестны и их определение должно являться частью решения задачи оптимизации. Такое представление временных интерва-

лов в виде отдельных переменных позволяет наложить на них определенные ограничения.

Интегральный член выражения (4.6) для показателя качества переписывается в виде суммы

$$J = \sum_{i=1}^N F [y(i), u(i-1), t_i] T_i + G [y(N)], \quad (4.19)$$

где $y(i) = y(t_i)$, $u(i) = u(t_i)$. Это можно рассматривать как простейшую форму аппроксимации выражения (4.6). Само собой разумеется, что можно применить и более совершенные формы численного интегрирования [19]. Во многих случаях показатель качества с самого начала может быть определен как сумма

$$J = \sum_{i=1}^N F [y(i), u(i-1), t_i] + G [y(N)]. \quad (4.20)$$

При формулировании задачи оптимального управления в виде задачи математического программирования каждая скалярная компонента вектора состояния или управления рассматривается для каждого дискретного момента времени $t_i (i=1, \dots, N)$ как особая переменная. Как указывалось выше, временные интервалы $T_i (i=1, \dots, N)$ также рассматриваются как переменные задачи. Все переменные задачи математического программирования рассматриваются как компоненты вектора переменных. Каждая переменная обозначается через $x(i) (i=1, \dots, N_v)$, где N_v — общее число переменных, рассматриваемых в задаче математического программирования. Скалярные компоненты задачи управления могут рассматриваться как переменные $x(i)$ следующим образом. Понятно, не существенно, в каком порядке будут обозначены эти переменные.

Всего количество переменных $N_v = N(n+m+1)$.

Принятые обозначения для компонент управления и состояния строятся следующим образом: например, $u_j(i)$ — это j -я компонента вектора $u(t)$ при $t=t_i$.

Для переменных задачи математического программирования $x(i)$ есть i -я переменная задачи, или i -я компонента вектора x . Теперь переменные $x(i) (i=1, \dots, N_v)$ подставляются в выражение (4.19) или (4.20) для показателя качества, в выражение (4.3), определяющее ограничения

Переменные задачи управления	Переменные задачи математического программирования	Переменные задачи управления	Переменные задачи математического программирования
T_1	$x(1)$	$u_1(N-1)$	$x[N+(N-1)m+1]$
T_2	$x(2)$.	.
.	.	.	.
.	.	.	.
T_N	$x(N)$	$u_m(N-1)$	$x(N+Nm)$
$u_1(0)$	$x(N+1)$	$y_1(1)$	$x(N+Nm+1)$
$u_2(0)$	$x(N+2)$	$y_2(1)$	$x(N+Nm+2)$
.	.	.	.
.	.	.	.
$u_m(0)$	$x(N+m)$	$y_n(1)$	$x(N+Nm+n)$
$u_1(1)$	$x(N+m+1)$.	.
$u_2(1)$	$x(N+m+2)$.	.
.	.	.	.
.	.	$y_1(N)$	$x[N+Nm+$
.	.	$y_2(N)$	$+(N-1)n+1]$
.	.	.	.
$u_m(1)$	$x(N+2m)$.	.
.	.	.	.
.	.	$y_n(N)$	$x(N+Nm+Nm)$
.	.	.	.

в конце процесса, а также во все другие ограничения, как например (4.8) и (4.9) для переменных управления, состояния и времени.

Естественно, что уравнения состояния системы должны удовлетворяться в любой момент времени. При применении математического программирования здесь могут возникнуть особые трудности.

По существу, должны быть рассмотрены два случая:

1. Решение уравнений состояния (4.1) в замкнутой форме известно.

2. Решение уравнений (4.1) в замкнутой форме неизвестно.

Эти два случая рассматриваются ниже.

1. Решение в замкнутой форме уравнения состояния известно. Решение уравнения (4.1) для произвольного времени t можно записать в виде

$$y(t) = \Phi[t, t_0, u, y(t_0)], \quad (4.21)$$

где Φ — n -мерная векторная функция. Для интервала между t_{i-1} и t_i решение (4.1) можно записать аналогично:

$$y(i) - \Phi_i[T_i, u(i-1), y(i-1)] = 0. \quad (4.22)$$

Записав (4.22) для компонент, получим для задачи математического программирования nN ограничений в виде равенств в форме (4.9).

Уравнение (4.22) можно рассматривать так же, как разностные уравнения состояния дискретной системы.

Управляющий вектор $u(i-1)$ считается постоянным на интервале $t_{i-1} \leq t \leq t_i$. Для непрерывной системы, конечно, это является приближением. В то же время, оно точно описывает импульсную систему.

При применении математического программирования переменные $x(i)$ надо подставить в уравнение (4.22) вместо переменных y , u и T_i , так же как и во все другие ограничения.

2. Решение в замкнутой форме неизвестно. В этом случае мы должны обратиться к численному интегрированию, как это приходится делать всегда, когда неизвестно решение в замкнутой форме для дифференциального уравнения. Другими словами, уравнения состояния (4.1) дискретизируются и записываются в виде разностных уравнений:

$$y(i+1) - y(i) = T_{i+1} f[y(i), u(i), t_i], \\ i=0, 1, \dots, N-1. \quad (4.23)$$

Уравнение (4.23), так же как и (4.22), может служить nN ограничениями в форме равенств. Следует подчеркнуть, что в то время как уравнение (4.22) является точным, уравнение (4.23) является аппроксимацией и полученные при его применении результаты должны соответствующим образом трактоваться.

Теоретическое обоснование некоторых методов дискретизации задач оптимального управления непрерывных систем недавно опубликовала Каллум [20] (см. также [30*—36*]).

Если мы имеем систему с очень быстрыми изменениями переменных состояния, например, высокочастотную колебательную систему, то рассмотренная аппроксимация может оказаться непригодной. С другой стороны,

когда переменные состояния изменяются со временем относительно медленно, можно получить требуемую степень точности, путем ограничения интервала между дискретными моментами времени

$$T_i \leq T_{\max}, \quad i=1, \dots, N. \quad (4.24)$$

В некоторых дискретных системах, кроме того, из практических соображений, может быть установлен ограничивающий минимальный интервал, т. е.

$$T_i \geq T_{\min}, \quad i=1, \dots, N. \quad (4.25)$$

Заметим также, что (4.23) получено с помощью очень простого подхода. Применяя более совершенные разностные схемы [19], можно улучшить точность решений.

Подводя итоги, вновь представим задачу оптимального управления в виде компактной формулировки задачи математического программирования. При этом будем предполагать, что решение в замкнутой форме для уравнений состояния неизвестно, поскольку так обстоит дело в большинстве задач:

$$\begin{aligned} \min (\text{или } \max) \left\{ \sum_{i=1}^N F[\mathbf{y}(i), \mathbf{u}(i-1), t_i] + G[\mathbf{y}(N)] \right. \\ \left. \mathbf{s}[\mathbf{y}(N)] \geq 0; \right. \\ \mathbf{g}(\mathbf{y}(i), \mathbf{u}(i), t_i) \geq 0; \mathbf{h}(\mathbf{y}(i), \mathbf{u}(i), t_i) = 0, \\ \left. i=1, \dots, N; \mathbf{y}(i+1) - \mathbf{y}(i) = T_{i+1} \mathbf{f}[\mathbf{y}(i), \mathbf{u}(i), t_i], \right. \\ \left. i=0, 1, \dots, N-1 \right\}. \quad (4.26) \end{aligned}$$

Последующие главы посвящены рассмотрению различных частных случаев задачи, сформулированной в (4.26).

Литература

1. A. P. Sage, Optimum Systems Control, Prentice-Hall, Englewood Cliffs, N. J., 1968.
2. M. Athans and P. L. Falb, Optimum Control, McGraw-Hill, New York, 1966. [Русский перевод: М. А т а н с, П. Ф а л б, Оптимальное управление, «Машиностроение», 1968.]
3. E. B. Lee and L. Markus, Foundations of Optimal Control Theory, Wiley, New York, 1967. [Русский перевод: Э. Б. Ли, Л. Маркус, Основы теории оптимального управления, «Наука», 1972.]

4. J. Abadie, ed., *Nonlinear Programming*, Wiley, New York, 1967.
5. R. Courant, *Differential and Integral Calculus*. vol. 1, 2nd. ed., Wiley-Interscience, New York, 1957. [Русский перевод первого издания: Курант Р., Курс дифференциального и интегрального исчисления, т. I, ГТТИ, 1931.]
6. L. Hurwicz, *Programming in Linear Spaces*, in *Studies in Linear and Nonlinear Programming* by K. J. Arrow, L. Hurwicz, and H. Uzava, Stanford University Press, Stanford, Cal., 1958, pp. 38—102. [Русский перевод: Гурвиц Л., Программирование в линейных пространствах, в книге К. Дж. Эрроу, Л. Гурвиц и Х. Удзава «Исследования по линейному и нелинейному программированию», ИЛ, 1962, стр. 65—155.]
7. D. L. Russell, *The Kuhn-Tucker Conditions in Banach Space with an Application to Control Theory*, *J. Math. Anal. Appl.*, 15, pp. 200—212 (1966).
8. B. N. Pshenichniy, *Linear Optimal Control Problems*, *J. SIAM Control* 4, pp. 577—593 (1966).
9. R. M. van Slyke, *A Duality Theory for Abstract Mathematical Programs with Applications to Optimal Control Theory*, Report DI-82-0671, Boeing Scientific Research Laboratories, Seattle, Wash., Oct. 1967.
10. J. Ponstein, *Multiplier Functions in Optimal Control*, *J. SIAM Control* 6, pp. 648—658 (1968).
11. D. O. Norris, *Lagrangian Saddle Points and Optimal Control*, *J. SIAM Control* 5, pp. 594—599 (1967).
12. Л. С. Понтрягин, В. Г. Болтянский, Р. В. Гамкрелидзе и Е. Ф. Мищенко, *Математическая теория оптимальных процессов*, Физматгиз, 1961.
13. H. Halkin and L. W. Neustadt, *General Necessary Conditions for Optimization Problems*, *Proc. Natl. Acad. Sci. U. S.* 56, pp. 1056—1071 (1966).
14. L. W. Neustadt, *An Abstract Variational Theory with Applications to a Broad Class of Optimization Problems*, *J. SIAM Control*, part I: *General Theory* 4, pp. 505—527 (1966); part II: *Applications* 5, pp. 90—137 (1967).
15. M. A. Hanson, *Bounds for Functionally Convex Optimal Control Problems*, *J. Math. Anal. Appl.* 8, pp. 84—89 (1964).
16. Y. C. Ho and P. В. Brentani, *On Computing Optimal Control with Inequality Constraints*, *J. SIAM Control* 1, pp. 319—348 (1963).
17. D. J. Wilde and C. S. Beightler, *Foundations of Optimization*, Prentice-Hall, Englewood Cliffs, N. J., 1967.
18. D. Табак and В. С. Куо, *Application of Mathematical Programming in the Design of Optimal Control Systems*, *Intern. J. Control* 10, pp. 545—552 (1969).
19. J. В. Scarborough, *Numerical Mathematical Analysis*, 5th ed., The Johns Hopkins Press, Baltimore, 1963. [Русский перевод: Дж. Скарборо, Численные методы математического анализа, ГТТИ, 1934.]
20. J. Cullum, *Discrete Approximations to Continuous Optimal Control Problems*, *SIAM J. Control* 7, pp. 32—49 (1969).
- 21*. А. Брайсон, Ю-ши Хо, *Прикладная теория оптимального управления*, «Мир», 1972.

- 22*. А. А. Первозванский, О связи основных теорем математического программирования и принципа максимума, Изв. АН СССР, Техническая кибернетика, № 1, 1967.
- 23*. Л. С. Гноенский, С. М. Мовшович, О применении методов математического программирования к задаче оптимального регулирования, Изв. АН СССР, Техническая кибернетика, № 5, 1964.
- 24*. А. И. Пропой, Элементы теории оптимальных дискретных процессов, «Наука», 1973.
- 25*. M. S a p n o p, C. C u l l u m, E. P o l a k. Theory of optimal control and mathematical programming, McGraw-Hill, 1970.
- 26*. А. Я. Дубовицкий, А. А. Милютин, Задачи на экстремум при наличии ограничений, Журнал вычислительной математики и математической физики 5, № 3 (1965).
- 27*. И. В. Гирсанов, Лекции по математической теории экстремальных задач, МГУ, 1970.
- 28*. Б. Н. Пшеничный, Необходимые условия экстремума, «Наука», 1969.
- 29*. Е. Г. Гольштейн, Теория двойственности в математическом программировании и ее приложения, «Наука», 1971.
- 30*. Ю. М. Ермольев, В. П. Гуленко, Конечноразностный метод в задачах оптимального управления, Кибернетика, № 3, 1967.
- 31*. Б. М. Будақ, Е. М. Беркович, Е. Н. Соловьева, О разностных аппроксимациях в задачах оптимального управления, Вестник МГУ, серия математики, механики, № 2, 1968.
- 32*. Б. М. Будақ, Е. М. Беркович, Е. Н. Соловьева, О сходимости разностных аппроксимаций для задач оптимального управления, Журнал вычислительной математики и математической физики, т. 9, № 3, 1969.
- 33*. Б. М. Будақ, Е. М. Беркович, Разностные аппроксимации для задач оптимального управления с подвижными концами при наличии фазовых ограничений I, II, III, Вестник МГУ, серия математики, механики, № 6. 1969; № 1, 1970; № 3, 1970.
- 34*. J. W. D a n i e l, On the convergence of a numerical method for optimal control problems, Journ. Opt. Th. Appl. 4, pp. 330—342 (1969).
- 35*. J. W. D a n i e l, The Ritz-Galerkin method for abstract optimal control problems, SIAM J. Control 11, № 1 (1973).
- 36*. W. E. B o s a r g e, O. G. J o h n s o n, Direct method approximation to the state regulator problem using a Ritz-Treftz suboptimal control, IEEE Trans. Autom. Contr. AC-15 (1970).

5.1. Введение

Задачи оптимального управления, как и многих других отраслей техники, в большей мере разработаны для линейных систем, чем для систем нелинейных. Это является следствием двух очевидных основных причин. Первая из них заключается в том, что для системы линейных уравнений состояния известно аналитическое решение в замкнутой форме. Естественно, что это облегчает аналитическое рассмотрение этого класса задач оптимального управления. Отчасти по этой же причине значительная часть работ, посвященных решению задач оптимального управления методами математического программирования, относится к линейным системам. Другая важная причина предпочтения, отдаваемого рассмотрению линейных систем, определяется тем, что во многих случаях задачу оптимального управления для линейной системы можно свести к задаче линейного программирования, которая численными методами решается значительно легче. Как указывалось в гл. 3, фактически это единственный класс задач математического программирования, в которых всегда можно за конечное число итераций достигнуть глобального экстремума. Даже если экстремум бесконечен или решение отсутствует, существующие алгоритмы позволяют легко получить информацию об этом. Разумеется, это не относится к задачам нелинейного программирования, за исключением класса задач квадратного программирования, рассмотренного в гл. 3. Поэтому всегда целесообразно попытаться сформулировать, если это возможно, задачу оптимального управления в виде задачи линейной или квадратичного программирования.

В данной главе будут рассмотрены несколько методов формулировки задачи оптимального управления для линейных непрерывных систем в виде задачи математического программирования. Различные методы иллюстрируются примерами практического характера.

5.2. Формулировка задачи

Рассмотренная в гл. 4 задача оптимального управления будет теперь сформулирована для линейной системы управления, т. е. для динамической системы, описываемой линейными уравнениями состояния. Используя понятия и обозначения, введенные в гл. 4, можно для линейной системы n -го порядка записать следующую систему уравнений состояния:

$$\dot{\mathbf{y}} = A\mathbf{y} + B\mathbf{u}, \quad (5.1)$$

где \mathbf{y} — n -мерный вектор состояния, $\dot{\mathbf{u}}$ — m -мерный вектор управления, A — $(n \times n)$ -мерная постоянная матрица, B — $(n \times m)$ -мерная постоянная матрица.

В общем случае матрицы A и B зависят от времени. Однако в настоящей главе предполагается, что они постоянны. Когда зависимость A и B от времени должна быть учтена, имеем следующее решение уравнений состояния (5.1) [9]:

$$\mathbf{y}(t) = \Phi(t, t_0)\mathbf{y}(t_0) + \int_{t_0}^t \Phi(t, \tau)B(\tau)\mathbf{u}(\tau) d\tau,$$

где

$$\frac{\partial \Phi(t, \tau)}{\partial t} = A(t)\Phi(t, \tau), \quad \Phi(\tau, \tau) = I.$$

Отсюда путем численного интегрирования можно получить решение уравнений состояния с переменными коэффициентами. Однако аналитическое решение для них, в отличие от систем с постоянными параметрами [9], неизвестно. Этот тип задач будет рассмотрен в гл. 6 вместе с нелинейными системами.

Показатель качества линейной системы, так же как и раньше, можно выразить в виде (4.6), (4.19) или (4.20). Однако по причинам, указанным в 5.1, особый интерес представляет показатель качества в виде линейной или квадратичной формы.

Некоторые из наиболее распространенных форм показателя качества перечисляются ниже [1—8]:

1) задача минимальной энергии

$$J = \int_{t_0}^{t_f} u^2(t) dt, \quad (5.2)$$

2) задача минимального расхода топлива

$$J = \int_{t_0}^{t_f} |u(t)| dt, \quad (5.3)$$

3) задача на быстрдействие

$$J = \int_{t_0}^{t_f} dt = (t_f - t_0). \quad (5.4)$$

Эти задачи будут подробно рассмотрены в следующих параграфах данной главы. Следует подчеркнуть, что приведенный перечень возможных показателей качества нельзя считать исчерпывающим. В зависимости от особенностей задачи могут применяться другие формы показателя качества.

Кроме показателя качества в задачу могут входить ограничения, заданные в форме (4.8) и (4.9). Если какая-либо из функций \mathbf{g} или \mathbf{h} нелинейна, то даже когда сама система управления линейна, мы вынуждены будем решать нелинейную задачу.

При рассмотрении многих практических задач достаточно наложить линейные ограничения типа (4.10) и (4.11).

Решение уравнений состояния (5.1) для любого момента времени можно представить в замкнутой форме

$$\mathbf{y}(t) = \Phi(t) \left[\mathbf{y}_0 + \int_0^t \Phi^{-1}(\tau) B \mathbf{u}(\tau) d\tau \right], \quad (5.5)$$

где $\Phi(t) = \Phi(t, 0)$ — $(n \times n)$ -мерная переходная матрица состояния, а \mathbf{y}_0 — начальное состояние в момент 0.

Переходя, как это было описано в 4.4, к дискретному времени, можно переписать (5.5) для интервала между двумя дискретными моментами t_{j-1} и t_j в виде

$$\mathbf{y}(j) = \Phi(t_j, t_{j-1}) \left[\mathbf{y}(j-1) + \int_{t_{j-1}}^{t_j} \Phi(t_{j-1}, \tau) B d\tau \mathbf{u}(j-1) \right], \\ j = 1, \dots, N. \quad (5.6)$$

Для линейных систем с постоянными параметрами переходная матрица $\Phi(t)$ имеет следующий вид [9]:

$$\Phi(t) = e^{At} \quad (5.7)$$

и

$$\Phi(t_j, t_{j-1}) = \Phi(t_j - t_{j-1}) = e^{A(t_j - t_{j-1})} = e^{AT_j}. \quad (5.8)$$

Уравнения переходных состояний с дискретным временем (5.6) можно теперь переписать в виде системы ограничивающих равенств следующим образом:

$$\mathbf{y}(j) - e^{AT_j} \left[\mathbf{y}(j-1) + \int_{t_{j-1}}^{t_j} e^{A(t_j - 1 - \tau)} B d\tau \mathbf{u}(j-1) \right] = \mathbf{0},$$

$$j = 1, \dots, N. \quad (5.9)$$

Уравнение (5.6) или (5.9) образуют систему из nN уравнений, являющихся ограничивающими равенствами. (Здесь n — размерность вектора \mathbf{y}).

Следует подчеркнуть, что в (5.6) и (5.9) управляющий вектор $\mathbf{u}(j-1)$ предполагается *постоянным* в интервале $t_{j-1} \leq t \leq t_j$. Это предположение необходимо для того, чтобы получить возможность численного решения линейных систем более общего вида. Как будет пояснено далее, можно для определенного класса систем избежать этого ограничения, применив так называемое обобщенное программирование [7].

Итак, типичная формулировка математического программирования задачи оптимального управления для линейной системы имеет следующий вид:

$$\left. \begin{aligned} \min \left\{ \sum_{i=1}^N F[\mathbf{y}(i), \mathbf{u}(i-1), i], \right. \\ \left. \begin{aligned} &u_{i_{\max}} - |u_i| \geq 0, i = 1, \dots, m; \\ &y_{i_{\max}} - |y_i| \geq 0, i = 1, \dots, n; \quad \mathbf{s}[\mathbf{y}(N)] \geq \mathbf{0}; \\ &\mathbf{y}(j) - e^{AT_j} \left[\mathbf{y}(j-1) + \right. \\ &\left. + \int_{t_{j-1}}^{t_j} e^{A(t_j - 1 - \tau)} B d\tau + \mathbf{u}(j-1) \right] = \mathbf{0}, \quad j = 1, \dots, N. \end{aligned} \right\} \end{aligned} \quad (5.10)$$

Здесь исходная, интегральная форма показателя качества представлена в дискретном виде. Разумеется, что

представленная формулировка не является наиболее общей и исчерпывающей. Однако она охватывает широкий класс практических случаев. Показатель качества может быть представлен в различной форме, например, в одной из форм, заданных уравнениями от (5.2) до (5.5) или в любом другом виде.

Конечная область, определяемая системой ограничений в виде неравенств $s \geq 0$ в отдельных случаях может состоять лишь из одной точки. Обычно начальное состояние $y(0)$ задается, но иногда оно может быть определено в виде множества $y(0) \in Y_0$. Функция s , определяющая конечную область, в общем случае может быть нелинейной.

В наиболее общем случае, когда интервалы T_j не заданы и представляют собой часть неизвестных переменных задачи, нелинейными являются также уравнения состояния, определяющие ограничения, записанные в форме равенств.

Подводя итоги, можно сказать, что в формулировке задачи математического программирования для линейной системы не обязательно ограничения являются строго линейными. В более общих случаях некоторые из ограничений могут быть нелинейными. С другой стороны, чтобы получить задачу математического программирования со строго линейными ограничениями, система должна быть линейной или линеаризованной, т. е. уравнения, описывающие состояния системы, должны быть линейными.

Можно добавить, что для обеспечения линейности всех ограничений в задаче, рассматриваемой в (5.10), надо выполнить следующие условия:

1. Интервалы $T_j (j=1, \dots, N)$ должны быть заданы априорно, однако в общем случае они могут быть неравными.

2. Функции $s[y(N)]$ должны быть линейными.

Чтобы получить задачу линейного программирования, функция F в (5.10) должна быть линейной по отношению ко всем ее переменным, а для получения задачи квадратичного программирования F должна быть соответственно квадратичной.

В следующих параграфах приведены примеры, иллюстрирующие различные формулировки задачи.

5.3. Линейная система с заданными временными интервалами

В этом параграфе будет рассмотрен пример, относящийся к более общему случаю формулировки задачи оптимального управления, как задачи математического программирования [5, 10]. Общность этого случая определяется тем, что дискретные временные интервалы в формулировке задачи математического программирования априорно неизвестны. Временные интервалы составляют часть переменных задачи оптимального управления и математического программирования. Их значения должны быть найдены в результате решения задачи. Из уравнения (5.10) видно, что разностные уравнения состояния являются нелинейными вследствие присутствия членов e^{AT_j} и $e^{A(t_j-\tau)}$, так как T_j и t_j априорно неизвестны. Эти уравнения образуют систему nN нелинейных ограничивающих равенств задачи математического программирования.

Для иллюстрации рассмотренного выше метода будет приведен пример линейной системы. Следующие уравнения состояния описывают рассматриваемую систему:

$$\left. \begin{aligned} \dot{y}_1 &= y_2, \\ \dot{y}_2 &= -0,16y_1 - 0,032y_2 + 0,16u. \end{aligned} \right\} \quad (5.11)$$

Цель управления заключается в переводе системы из начального состояния

$$y(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

в конечное состояние

$$y_f = \begin{bmatrix} y_{1f} \\ y_{2f} \end{bmatrix}$$

при минимуме определенного квадратичного показателя качества и отсутствии перелетов y_1 перед достижением целевой точки. Значения управляющей входной переменной в этой задаче предполагаются известными. Неизвестными переменными являются только интервалы времени,

по истечении которых значение управления может изменяться.

Допустим, что требуемое конечное состояние есть

$$y_f = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

и пусть управляющая переменная u соответствует рис. 5.1. Конечный момент обозначен через $t_f = t_4$, а общее число интервалов выбрано равным 4.

Моменты t_1, \dots, t_4 или интервалы T_1, \dots, T_4 должны быть найдены в результате вычислений, выполняемых при решении задачи оптимизации.

Уменьшение величины u в момент t_2 производится,

чтобы предотвратить перелет для y_1 . В конечный момент t_4 y_1 достигает требуемого значения y_{1f} или заданной доли этой величины.

Запишем уравнения переходных состояний (5.11) для случая постоянного u в виде

$$y(t) = \Phi(t, t_0)y(t_0) + \mathbf{b}(t, t_0)u, \tag{5.12}$$

где

$$\Phi(t, t_0) = e^{-0.016(t-t_0)} \times \begin{bmatrix} \sin[0,4(t-t_0) + 87,7^\circ] & 2,5 \sin 0,4(t-t_0) \\ 0,4 \sin 0,4(t-t_0) & -\sin[0,4(t-t_0) - 87,7^\circ] \end{bmatrix}$$

и

$$\mathbf{b} = \begin{bmatrix} 1 - e^{-0.016(t-t_0)} \sin[0,4(t-t_0) + 87,7^\circ] \\ 0,4 e^{-0.016(t-t_0)} \sin 0,4(t-t_0) \end{bmatrix}.$$

Для интервала T_j уравнение (5.12) запишется в следующем виде:

$$y_1(j) = e^{-0.016T_j} \left\{ \begin{aligned} & [\sin(0,4T_j + 1,531) y_1(j-1) + \\ & + 2,5 \sin(0,4T_j) y_2(j-1)] + \\ & + [1 - e^{-0.016T_j} \sin(0,4T_j + 1,531)] u_{j-1}, \end{aligned} \right\} \tag{5.13}$$

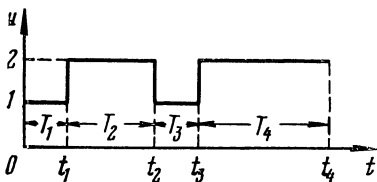


Рис. 5.1. Управление.

$$y_2(j) = -e^{-0,016T_j} [0,4 \sin(0,4T_j) y_1(j-1) + \sin(0,4T_j - 1,531) y_2(j-1)] + 0,4 e^{-0,016T_j} \sin(0,4T_j) u_{j-1}, \quad j=1, \dots, 4.$$

Примем для данного случая показатель качества:

$$\text{минимизировать } J = \sum_{j=1}^{N-1} [y_1^2(j) + y_2^2(j)]. \quad (5.14)$$

На временные интервалы дополнительно накладываются следующие ограничения:

$$\sum_{i=1}^N T_i \leq t_{\max} = 0,0305 \text{ сек} \quad (5.15)$$

и

$$T_i \geq T_{\min} = 10^{-3} \text{ сек}, \quad i=1, \dots, N. \quad (5.16)$$

Следует отметить, что ограничения (5.15) и (5.16) относятся исключительно ко временным интервалам. Аналогичную задачу рассмотрел Фолц [11]. В этой статье были получены необходимые условия оптимальности.

Введение ограничений на интервалы времени при применении нелинейного программирования не вызывает вычислительных трудностей. Эти ограничения учитываются так же, как и любые другие, и не вносят в задачу особых осложнений.

Итак, должна быть численно решена следующая задача. Найти моменты переключения t_1 , t_2 , t_3 и t_4 , когда управление должно последовательно принимать значения 1, 2, 1, 2, с тем чтобы система перешла без перелетов от значения $y_1(t_0) = 0$ к $y_1(t_4)$, отличающемуся от 2 не более чем на 1%, т. е. равному $y_1(t_4) = 1,98$, при минимуме выражения (5.14), с соблюдением ограничений (5.13), (5.15) и (5.16). Таким образом,

$$y_1(j) \leq 2, \quad j=1, 2, 3. \quad (5.17)$$

Значение $y_2(t_f)$ взято равным 0,001. Следуя изложенному в гл. 4, принимаем переменные задачи управления в качестве переменных задачи математического программирования.

Переменные математического программирования подставляем в выражение (5.14) для показателя качества и в ограничения (5.13) и (5.15), (5.16) и (5.17). В ре-

Переменные задачи управления	Переменные задачи математического программирования	Переменные задачи управления	Переменные задачи математического программирования
T_1	$x(1)$	$y_2(1)$	$x(6)$
T_2	$x(2)$	$y_1(2)$	$x(7)$
T_3	$x(3)$	$y_2(2)$	$x(8)$
T_4	$x(4)$	$y_1(3)$	$x(9)$
$y_1(1)$	$x(5)$	$y_2(3)$	$x(10)$

зультате приходим к следующей задаче математического программирования:

$$\min \left\{ \sum_{j=5}^{10} x^2(j) \mid x(i) \geq 10^{-3}, i = 1, 2, 3, 4; \sum_{i=1}^4 x(i) \leq 0,0305; \right.$$

$$\left. x(5) \leq 2; x(7) \leq 2; x(9) \leq 2; \right.$$

$$x(5) = 1 - \exp[-0,016x(1)] \sin[0,4x(1) + 1,531];$$

$$x(6) = 0,4 \exp[-0,016x(1)] \sin[0,4x(1)];$$

$$x(7) = \exp[-0,016x(2)] \{ \sin[0,4x(2) + 1,531]x(5) + \\ + 2,5 \sin[0,4x(2)]x(6) \} + 2 \{ 1 - \exp[-0,016x(2)] \times \\ \times \sin[0,4x(2) + 1,531] \};$$

$$x(8) = -\exp[-0,016x(2)] \{ 0,4 \sin[0,4x(2)]x(5) + \\ + \sin[0,4x(2) - 1,531]x(6) \} + 2 \{ 1 - \exp[-0,016x(2)] \times \\ \times \sin[0,4x(2) + 1,531] \};$$

$$x(9) = \exp[-0,016x(3)] \{ \sin[0,4x(3) + 1,531]x(7) + \\ + 2,5 \sin[0,4x(3)]x(8) \} + 0,4 \exp[-0,016x(3)] \times \\ \times \sin[0,4x(3)];$$

$$1,98 = \exp[-0,016x(4)] \{ \sin[0,4x(4) + 1,521]x(9) + \\ + 2,5 \sin[0,4x(4)]x(10) \} + 2 \{ 1 - \exp[-0,016x(4)] \times \\ \times \sin[0,4x(4) + 1,531] \};$$

$$0,001 = \exp[-0,016x(4)] \{ 0,4 \sin[0,4x(4)]x(9) + \\ + \sin[0,4x(4) - 1,531]x(10) \} + 0,8 \exp[-0,016x(4)] \times \\ \times \sin[0,4x(4)] \}.$$

Теперь мы имеем типичную задачу математического программирования с десятью переменными, восемью

ограничениями в виде неравенств и восемью ограничивающими равенствами. Задача представлена в форме, пригодной для осуществления процедуры математического программирования. Показатель качества (целевая функция) — квадратичный и ограничения в форме неравенств — линейны, однако ограничения в форме равенств нелинейны. Таким образом, мы имеем задачу нелинейного программирования. Дальнейшая обработка исходных данных для вычислений зависит от используемой процедуры математического программирования. В данном случае была применена процедура метода последовательной минимизации без ограничений, рассмотренная в гл. 3. Более подробно разработанный пример использования метода последовательной минимизации без ограничений приведен в 6.4.

Результаты решения нашей задачи приводятся ниже:

i	T_i сек	t_i сек	y_1	y_2	u
1	11,99	11,99	0,561	0,550	1,0
2	2,42	14,41	0,736	0,550	2,0
3	1,00	15,41	0,471	0,678	1,0
4	5,95	21,36	1,980	0,001	2,0

Оптимальный показатель качества оказался равным

$$J_{\text{opt}} = 2,15.$$

Задача была записана на языке Фортран IV и решалась на ЦВМ ИБМ 7094. Для решения задачи потребовалось приблизительно 1,5 мин. Это является обычной продолжительностью решения методом последовательной минимизации без ограничений задач такого объема и такой сложности. Следует заметить, что применение принципа максимума привело бы к крайне трудной двухточечной краевой задаче. Хотя в настоящее время есть несколько программ, пригодных для практического использования, тем не менее учет ограничений, наложенных на переменные состояния в методах, основанных на принципе максимума, представляет весьма внушитель-

ную задачу. В то же время, при применении математического программирования учет ограничений, подобных рассмотренным, вовсе не представляет собой проблемы.

5.4. Задача минимального расхода топлива

В настоящем параграфе рассматривается задача оптимального управления подачей топлива [1, 2, 4]. В частности, будет рассмотрен класс задач, которые могут быть сформулированы в виде задач линейного программирования. Учитывая (15.3) и (15.10), можно сформулировать следующую задачу:

$$\min \left\{ \sum_{i=0}^{N-1} \sum_{j=0}^M |u_j(i)| \left| A_u \mathbf{u}(i) \leq \mathbf{b}_u; i = 0, 1, \dots, N-1; \right. \right. \\ \left. \left. A_y \mathbf{y}(i) \leq \mathbf{b}_y; i = 1, \dots, N-1; S_y(N) \leq \mathbf{b}_s \right\}, \quad (5.18)$$

где A_u , A_y и S — постоянные матрицы с размерами соответственно $m \times m$, $n \times n$ и $n \times n$, а \mathbf{b}_u , \mathbf{b}_y и \mathbf{b}_s — постоянные векторы с размерами соответственно m , n и n . Интервал равен

$$T = t_i - t_{i-1} \text{ для всех } i = 1, \dots, N.$$

Выражение для $\mathbf{y}(N)$ в (5.18) можно записать следующим образом. Используя (5.9) и полагая постоянным $T_i = T$ для всех i , можно записать

$$\mathbf{y}(i+1) = e^{AT} \mathbf{y}(i) + \int_{t_i}^{t_{i+1}} e^{A(t_{i+1}-\tau)} B d\tau \mathbf{u}(i) = \\ = E \mathbf{y}(i) + F \mathbf{u}(i),$$

где $E = e^{AT}$ — $(n \times n)$ -мерная постоянная матрица и $F = \int_{t_i}^{t_{i+1}} e^{A(t_{i+1}-\tau)} B d\tau$ есть постоянная матрица с размерами $n \times m$.

Конечное состояние [4] равно

$$\mathbf{y}(N) = E^N \mathbf{y}(0) + \sum_{i=1}^N E^{i-1} F \mathbf{u}(N-i). \quad (5.19)$$

Далее, это выражение подставляем в формулировку задачи (5.18). Ограничения в (5.18) для состояния и управления являются линейными и имеют более общий характер, чем ограничения в (5.10).

Задача, сформулированная в (5.18), имеет следующие неизвестные переменные: $u_j(i)$, $j=1, \dots, m$; $i=0, 1, \dots, N-1$. Общее число переменных равно mN . Число линейных ограничений равно соответственно:

ограничения на управления в форме неравенств mN ,
ограничения на состояния в форме неравенств $n(N-1)$,
ограничения на конечное состояние в форме неравенств n . Общее число ограничений равно $N(n+m)$.

Во многих практических случаях расход топлива прямо пропорционален абсолютной величине управляющей переменной, что и определяет формулировку показателя качества в (5.18). Очевидно, что этот показатель качества нелинеен относительно $u_j(i)$ и поэтому нельзя непосредственно применить линейное программирование для численного решения этого класса задач. Существует несколько методов представления показателя качества задачи минимального расхода топлива в линейной форме. Два таких метода будут здесь приведены.

Метод ограничивающих переменных. Этот метод предложили Фас и Хиггинс [2]. Новая совокупность вспомогательных переменных $a(i)$ ($i=0, \dots, N-1$) вводится так, что удовлетворяются следующие ограничивающие неравенства:

$$|u_j(i)| \leq a_j(i), \quad j=1, \dots, m; \quad i=0, 1, \dots, N-1, \quad (5.20)$$

или

$$\left. \begin{aligned} u_j(i) &\leq a_j(i), \\ -u_j(i) &\leq a_j(i), \end{aligned} \right\} j=1, \dots, m; \quad i=0, 1, \dots, N-1. \quad (5.21)$$

При этом показатель качества равен

$$\min J = \sum_{i=0}^{N-1} \sum_{j=1}^m a_j(i), \quad (5.22)$$

т. е. представляет собой линейную функцию переменных $a_j(i)$. Применение этого метода означает введение в задачу дополнительных mN переменных и $2mN$ ограничений.

Фас [4] разработал другой, эффективный вариант данного метода численного решения этого класса задач, который намного уменьшает требуемый объем вычислений. При этой формулировке задачи общее число ограничений равно $2mN$, половина из которых типа ограничения

$$a_j(i) \leq u_{\max}.$$

Метод разности переменных. Этот метод был применен Уэспи [1]. Управляющие переменные $u_j(i)$ ($j=1, \dots, m$; $i=0, 1, \dots, N-1$) следующим образом разбиваются на пары неотрицательных переменных:

$$\left. \begin{aligned} u_j(i) &= u'_j(i) - u''_j(i), \\ u'_j(i) &\geq 0, \quad j = 1, \dots, m, \\ u''_j(i) &\geq 0, \quad i = 0, 1, \dots, N-1. \end{aligned} \right\} \quad (5.23)$$

Показатель качества равен теперь

$$\min J = \sum_{i=0}^{N-1} \sum_{j=1}^m [u'_j(i) + u''_j(i)]. \quad (5.24)$$

Уравнение (5.23) подставляют во все ограничения (5.18). При этом общее число ограничений остается прежним. Однако, как и в предыдущем методе, общее число переменных удваивается. В итоге общая матрица ограничений будет иметь прежнее число столбцов, а число ее строк уменьшится на $2mN$.

5.5. Задача встречи при минимальном расходе топлива

В качестве иллюстративного примера применения математического программирования к задаче минимального расхода топлива, в настоящем параграфе рассматривается задача оптимальной встречи в космосе. Излагаемые в этом параграфе результаты были получены Уэспи [1].

Полет космического корабля можно разбить на несколько фаз: запуск, вывод на орбиту, переход на другую орбиту, промежуточная фаза и окончательная коррекция.

Необходимо рассмотреть фазу перехода с одной земной орбиты на другую. При этом орбиты могут существенно различаться по энергии. Разница высот может составлять, например, сотню морских миль. После выполнения такого перехода скорость космического корабля и его положение в заданный момент времени может все же отличаться от требуемых положения и скорости. Предположим, что допустимое отклонение, зависящее от очень многих факторов, связанных с запуском, составляет величины порядка от 1 до 10 морских миль и 100—300 футов в секунду.

Промежуточная коррекция траектории и маневр на конечном участке проводятся со следующими целями:

1. Обеспечить в заранее заданный момент заданные положение и скорость космического корабля, с тем чтобы последующая траектория свободного полета соответствовала требуемым эфемеридам.

2. Обеспечить согласование с положением и скоростью другого орбитального космического корабля для выполнения условий встречи.

Маневр на конечном участке траектории необходим также для поддержания требуемых эфемерид при наличии консервативных и неконсервативных возмущающих сил.

Принимаются следующие предположения:

1. Предполагается, что двигатели взаимно перпендикулярны и неподвижны относительно корабля.

2. Используются линеаризованные уравнения движения с дискретным временем (типа Уилтшир — Клохеси [12]).

3. Принята координатная система с началом координат, совпадающим с целью. В этой системе координат линеаризованные уравнения дают достаточную точность при интересующих нас относительных расстояниях.

4. Минимизируется расход топлива.

5. Начальная и целевая орбиты могут быть круговыми и эллиптическими, и могут происходить изменения плоскости орбиты в пределах возможных ускорений, времени маневрирования и областей линеаризации.

6. Навигационная система (определение положения объектов) предполагается достаточно точной для решения задачи минимизации расхода топлива.

Предполагая, что орбита круговая, имеем следующую систему уравнений:

$$\left. \begin{aligned} \ddot{x} - 2\omega\dot{y} &= F_x, \\ \ddot{y} + 2\omega\dot{x} - 3\omega^2 y &= F_y, \\ \ddot{z} + \omega^2 z &= F_z, \end{aligned} \right\} \quad (5.25)$$

где x, y, z — декартовы координаты управляемого спутника в координатной системе с началом, совпадающим с целью, $\omega^2 = \mu/R_T^3$, μ — гравитационная постоянная Земли, R_T — радиус круговой орбиты спутника-цели в земной системе координат, F_x, F_y, F_z — составляющие в прямоугольной системе координат относительного ускорения управляемого спутника относительно целевого спутника.

Эти уравнения встречи приведены Клохеси и Уилтширом [12].

Рассмотрим первые два уравнения системы (5.25). Эти уравнения могут быть использованы для описания маневров в плоскости орбиты независимо от третьего уравнения, описывающего движение вне плоскости по оси z . Пусть $x=y_1, \dot{y}_1=y_2, y=y_3, \dot{y}_3=y_4, F_x=u_1$ и $F_y=u_2$. Тогда имеем

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \dot{y}_3 \\ \dot{y}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2\omega \\ 0 & 0 & 0 & 1 \\ 0 & -2\omega & +3\omega^2 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \quad (5.26)$$

Аналогично, для уравнения относительно z , полагая $z=y_5, \dot{y}_5=y_6$ и $u_3=F_z$, получим

$$\begin{bmatrix} \dot{y}_5 \\ \dot{y}_6 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \omega^2 & 0 \end{bmatrix} \begin{bmatrix} y_5 \\ y_6 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_3. \quad (5.27)$$

Уравнения состояния для плоскости имеют вид

$$\dot{\mathbf{y}} = \mathbf{F}\mathbf{y} + \mathbf{B}\mathbf{u},$$

где F и B — постоянные матрицы. Переходная матрица

состояния получается следующая:

$$\Phi(t) = \mathcal{L}^{-1}[Is - F]^{-1},$$

где \mathcal{L}^{-1} обозначает операцию обратного преобразования Лапласа и I — единичную матрицу [9].

Следовательно,

$$\Phi(t) = \begin{bmatrix} 1 & \left| \begin{array}{l} \frac{4}{\omega} \cos \omega t - (3t) \\ 4 \cos \omega t - 3 \\ \frac{2}{\omega} \cos \omega t - \left(\frac{2}{\omega}\right) \\ -2 \sin \omega t \end{array} \right. & \left| \begin{array}{l} 6\omega - 6 \sin \omega t \\ 6\omega - 6\omega \cos \omega t \\ 4 - 3 \cos \omega t \\ 3\omega \sin \omega t \end{array} \right. & \left| \begin{array}{l} \frac{2}{\omega} - \frac{2}{\omega} \cos \omega t \\ 2 \sin \omega t \\ \frac{1}{\omega} \sin \omega t \\ \cos \omega t \end{array} \right. \end{bmatrix}. \quad (5.28)$$

Из (5.26) и (5.28) можно получить разностные уравнения. Для нашего случая имеем

$$y(k+1) = Ay(k) + Bu(k), \quad k=0, 1, \dots, N-1, \quad (5.29)$$

где

$$A = \Phi(T) = \begin{bmatrix} 1 & \left| \begin{array}{l} \frac{4}{\omega} \sin \omega T - 3T \\ 4 \cos \omega T - 3 \\ \frac{2}{\omega} \cos \omega T - \frac{2}{\omega} \\ -2 \sin \omega T \end{array} \right. & \left| \begin{array}{l} 6\omega T - 6 \sin \omega T \\ 6\omega - 6\omega \cos \omega T \\ 4 - 3\omega T \\ 3\omega \sin \omega T \end{array} \right. & \left| \begin{array}{l} \frac{2}{\omega} - \frac{2}{\omega} \cos \omega T \\ -\sin \omega T \\ \frac{1}{\omega} \sin \omega T \\ \cos \omega T \end{array} \right. \end{bmatrix}, \quad (5.30)$$

$$B = \begin{bmatrix} \left| \begin{array}{l} -\frac{4}{\omega^2} \cos \omega T - \left(\frac{3T^2}{2}\right) - \frac{4}{\omega^2} \\ \frac{4}{\omega} \sin \omega T - (3T) \\ \frac{2}{\omega^2} \sin \omega T - \left(\frac{2T}{\omega}\right) \\ \frac{2}{\omega} \cos \omega T - \left(\frac{2}{\omega}\right) \end{array} \right. & \left| \begin{array}{l} \frac{2T}{\omega} - \frac{2}{\omega^2} \sin \omega T \\ -\frac{2}{\omega} \cos \omega T - \left(\frac{2}{\omega}\right) \\ -\frac{1}{\omega^2} \cos \omega T - \left(\frac{1}{\omega^2}\right) \\ \frac{1}{\omega} \sin \omega T \end{array} \right. \end{bmatrix} \quad (5.31)$$

$$\mathbf{u}(k) = [u_1(k), u_2(k)].$$

Приведенные ниже результаты, полученные на ЦВМ, относятся к цели, находящейся на круговой орбите радиусом 300 морских миль. Интервал между дискретными

моментами времени был выбран равным $T=20$ сек. При этих параметрах $\omega T=0,022$ рад и

$$A = \begin{bmatrix} 1,000 & 19,994 & 0,000 & 0,438 \\ 0 & 0,999 & 0,000 & 0,044 \\ 0 & -0,438 & 1,000 & 19,998 \\ 0 & -0,044 & 0,000 & 1,000 \end{bmatrix}, \quad (5.32)$$

$$B = \begin{bmatrix} 199,969 & 2,923 \\ 19,994 & 0,438 \\ -2,923 & 199,992 \\ -0,438 & 19,998 \end{bmatrix}. \quad (5.33)$$

Начальные и конечные условия задачи:

$$\mathbf{y}(0) = [y_1(0), y_2(0), y_3(0), y_4(0)]^T \quad \text{и} \quad \mathbf{y}(N) = [0, 0, 0, 0]^T.$$

Из (5.19), произведя замену $A \equiv E$ и $B \equiv F$, получим

$$-\mathbf{y}(0) = \sum_{j=0}^{N-1} A^{-1-j} B \mathbf{u}(j). \quad (5.34)$$

Обозначим

$$R(j) = A^{-1-j} B, \quad (5.35)$$

так что

$$R(j+1) = A^{-1} R(j). \quad (5.36)$$

Тогда задача линейного программирования приобретает следующий вид:

1) найти $\mathbf{u}(k) = [u_1(k), u_2(k)]^T$, $k=0, 1, \dots, N-1$,

2) при условии

$$-\mathbf{y}(0) = \sum_{j=0}^{N-1} R(j) \mathbf{u}(j), \quad |u_i(j)| \leq M, \quad i = 1, 2, \quad (5.37)$$

3) чтобы минимизировать расход топлива

$$\sum_{k=0}^{N-1} \{|u_1(k)| + |u_2(k)|\}. \quad (5.38)$$

Хотя ограничения (5.37) и линейны, тем не менее применить метод линейного программирования все же нельзя, так как целевая функция (5.38) содержит абсолютное значение. Чтобы преодолеть эту трудность, применим метод разности переменных согласно (5.23) и

(5.24). Введем следующие новые переменные:

$$\left. \begin{array}{l} u'_i(k) \geq 0, \\ u''_i(k) \geq 0, \end{array} \right\} k = 0, 1, \dots, N-1; i = 1, 2,$$

такие, что

$$\left. \begin{array}{l} u_1(k) = u'_1(k) - u''_1(k), \\ u_2(k) = u'_2(k) - u''_2(k). \end{array} \right\} \quad (5.39)$$

Тогда показатель качества (5.38) может быть записан так:

$$\min \sum_{k=0}^{N-1} [u'_1(k) + u''_1(k) + u'_2(k) + u''_2(k)]. \quad (5.40)$$

Теперь ограничения имеют следующий вид:

$$-y(0) = \sum_{j=0}^{N-1} R(j)[u'(j) - u''(j)], \quad (5.41)$$

$$u'_i(k) + u''_i(k) \leq M, \quad i = 1, 2, \quad (5.42)$$

где $k=0, 1, \dots, N-1$.

Введем теперь переменные задачи линейного программирования.

Управляющие переменные	Переменные линейного программирования	Управляющие переменные	Переменные линейного программирования
$u'_1(0)$	$x(1)$	$u''_1(0)$	$x(2N+1)$
$u'_2(0)$	$x(2)$	$u''_2(0)$	$x(2N+2)$
$u'_1(1)$	$x(3)$	$u''_1(1)$	$x(2N+3)$
$u'(1)$	$x(4)$	$u''_2(1)$	$x(2N+4)$
\vdots	\vdots	\vdots	\vdots
\vdots	\vdots	\vdots	\vdots
$u'_1(N-1)$	$x(2N-1)$	$u''_1(N-1)$	$x(4N-1)$
$u'_2(N-1)$	$x(2N)$	$u''_2(N-1)$	$x(4N)$

Общее число переменных равно $4N$. Теперь можно в окончательном виде сформулировать задачу линейного программирования:

$$\min \left\{ \sum_{i=1}^{2N} [x(i) + x(2N + i)] \mid x(i) \geq 0, i = 1, \dots, 4N; \right. \\ \left. x(i) + x(2N + i) \leq M, i = 1, \dots, 2N; \right. \\ \left. -y(0) = \sum_{j=1}^N R(j-1) \begin{bmatrix} x(2j-1) & -x(2N+2j-1) \\ x(2j) & -x(2N+2j) \end{bmatrix} \right\}. \quad (5.43)$$

Задача, сформулированная в виде (5.43), может быть решена с помощью любой машинной программы для

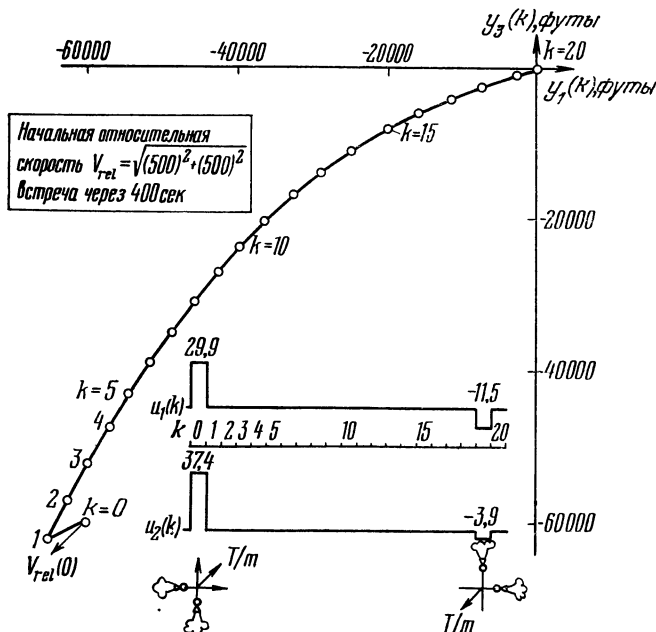


Рис. 5.2. Типичная траектория при большой относительной скорости [1].

линейного программирования. В нашем случае была использована система LP/90 [13]. Результаты вычислений приводятся на рис. 5.2—5.4.

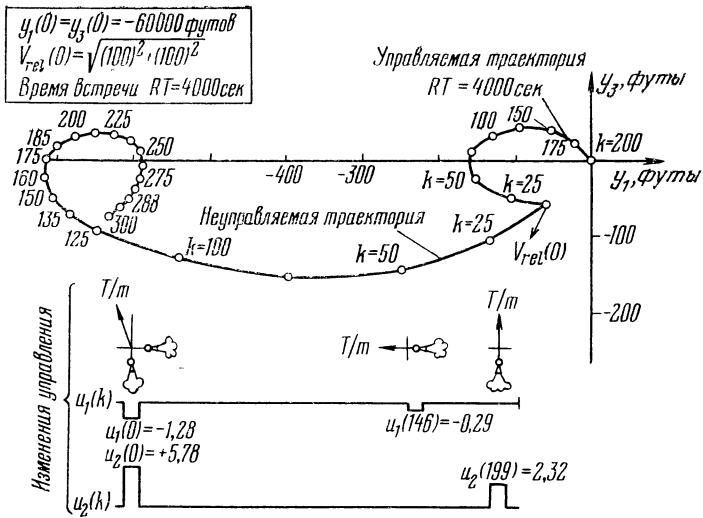


Рис. 5.3. Управляемая и неуправляемая траектория I [1].

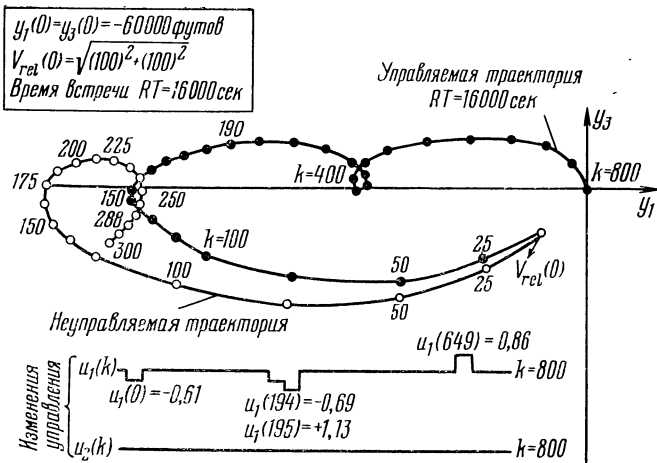


Рис. 5.4. Управляемая и неуправляемая траектория II [1].

На рис. 5.2 показана типичная траектория для сравнительно большой начальной относительной скорости. Оптимальная зависимость ускорения от времени была вычислена для времени встречи равной 200 сек и при интервалах дискретного времени, взятых равными 20 сек. Если уменьшить длительность интервалов, то показанные на рисунке импульсы станут более узкими, но будут иметь большую амплитуду. В пределе получится теоретическая зависимость от оптимального импульсного управления.

Случаи более реалистических начальных относительных скоростей показаны на рис. 5.3 и 5.4.

5.6. Задача управления, оптимального по быстродействию

Рассматриваемая система описывается следующей системой уравнений состояния [3]:

$$\dot{y} = Ay + Bu, \quad (5.44)$$

где y — n -мерный вектор состояний, u — r -мерный вектор управления, A — $(n \times n)$ -мерная постоянная матрица, B — $(n \times r)$ -мерная постоянная матрица.

На систему при всех t наложены следующие ограничения:

$$C_u u \leq v_u(t), \quad (5.45)$$

$$C_v y \leq v_v(t), \quad (5.46)$$

где v_u — q -мерный вектор, v_v — p -мерный вектор, C_u — $(q \times r)$ -мерная постоянная матрица, C_v — $(p \times n)$ -мерная постоянная матрица.

Целевое множество при $t = t_f$ определяется следующими неравенствами:

$$Gy(t_f) \leq v_g, \quad (5.47)$$

$$C_v y(t_f) \leq v_v, \quad (5.48)$$

где v_g — g -мерный вектор, G — $(g \times n)$ -мерная постоянная матрица.

Задача оптимального по быстродействию управления для рассматриваемой системы формулируется следующим образом.

Дано начальное состояние $y(0) = y_0$. Найти такую управляющую векторную функцию $u(t)$, что

- 1) $u(t)$ удовлетворяет ограничениям (5.45);
- 2) система, определяемая (5.44), переходит из $y(0) = y_0$ в некоторую точку целевого множества, заданного в (5.47) и (5.48);
- 3) полученная траектория должна удовлетворять ограничению (5.46);
- 4) время конца перехода t_f — минимально.

Алгоритм для приближенного решения этой задачи, использующий линейное программирование, предложил Фас [3]. Этот алгоритм основан на понятии *множества достижимости*, которое будет сейчас определено.

Определение 5.1. *Множество достижимости A_f* есть множество всех начальных состояний, которые могут быть достигнуты при возвращении по допустимым траекториям на интервале $0 \leq t \leq t_f$ из заданного целевого множества.

Другими словами, A_f есть множество всех начальных состояний, для которых существуют допустимые управления $u(t)$ на интервале $0 \leq t \leq t_f$, порождающие допустимые траектории, конец которых находится в заданном целевом множестве. Управления и траектории являются *допустимыми*, если они удовлетворяют соответствующим им ограничениям.

Минимальное время t_f в задаче управления, оптимального по быстродействию, соответствует времени, при котором y_0 оказывается впервые в множестве достижимости A_f . При исследовании некоторых свойств множества A_f для рассматриваемого в данном параграфе класса задач можно выявить сущность задачи управления, оптимального по быстродействию.

Когда система линейна и множество допустимых управляющих функций ограничено (в результате чего ограничены производные переменных состояния), множество достижимости непрерывно по t_f . Таким образом, минимальное время соответствует моменту, когда y_0 впервые появляется на границе A_f . Это также означает, что если при $t_f = t_{f1}$ y_0 не принадлежит множеству A_f и если при некотором другом времени $t_{f2} > t_{f1}$ y_0 есть внутренняя точка A_f , то при некотором промежуточном значении времени t_f y_0 должно лежать на границе A_f .

Второе важное свойство множества достижимости заключается в том, что оно во всех случаях является выпуклым. Это следует из выпуклости ограничений и линейности системы. Полное доказательство дано в [3].

Третье свойство, которым при определенных условиях обладает множество достижимости,— это описываемое ниже свойство *включения*.

Определение 5.2. Множество достижимости удовлетворяет отношению *включения* (и тем самым обладает свойством включения) на интервале $0 \leq t \leq t_{f2}$, если всякий раз, когда какое-либо начальное состояние y_0 является элементом множества достижимости при t_{f1} , оно является элементом множества достижимости при всех таких t_f , что $t_{f1} \leq t_f \leq t_{f2}$.

Это свойство является признаком того, что множество достижимости является неубывающим. Хотя этим свойством обладают множества достижимости многих систем, тем не менее оно не является общим свойством всех множеств достижимости. Этим свойством обладают все системы, у которых целевым множеством является начало координат. Однако в общем случае целевого множества, чтобы выполнялось отношение включения, должны быть выполнены дополнительные условия.

Достаточные условия, гарантирующие выполнение отношения включения данным множеством достижимости, дает следующая теорема, доказательство которой приведено в [3].

Теорема 5.1. Если на интервале $0 \leq t \leq t_f$ для любых двух значений времени t_{f0} и t_{f1} таких, что

$$0 \leq t_{f0} \leq t_{f1} \leq t_{f2}, \quad (5.49)$$

и любого состояния $y(t_{f0})$ в целевом множестве существует допустимая управляющая функция на интервале $t_{f0} \leq t \leq t_{f1}$ такая, что $y(t)$, определяемая выражением

$$y(t) = e^{A(t-t_{f0})}y(t_{f0}) + \int_{t_{f0}}^t e^{A(t-\sigma)}Bu(\sigma) d\sigma, \quad (5.50)$$

1) является допустимой траекторией на $t_{f0} \leq t \leq t_{f1}$,

2) $y(t_{f1})$ является элементом целевого множества, то множество достижимости удовлетворяет отношению включения на интервале $0 \leq t \leq t_{f2}$.

Следует отметить, что эти условия являются достаточными, но не необходимыми. Многие множества достижимости обладают свойством включения, даже когда приведенные в теореме 5.1 ограничения на целевое множество не выполняются. В тех случаях, когда условия теоремы выполняются при нескольких управлениях, говорят, что система удовлетворяет отношению включения второго рода.

Пусть множество достижимости удовлетворяет отношению включения второго рода. Тогда, если начальное состояние на границе множества достижимости достигнуто единственной траекторией, то эта траектория является оптимальной по быстродействию. Действительно, если бы это состояние принадлежало области достижимости в какой-то более ранний момент, то, тогда, согласно предположению, траектория не являлась бы единственной.

В действительности такая ситуация обычно и бывает. Оптимальная по быстродействию траектория обычно является единственной (однако это, конечно, не необходимо), и все точки на границе множества A_f , которые не принадлежат траекториям, оптимальным по быстродействию, достигаются многими траекториями. Этот факт играет важную роль в описываемом приближенном способе.

Главная идея этой процедуры приближения заключается в таком изменении формулировки задачи, согласно которой на фиксированном интервале времени может быть получена траектория, оканчивающаяся на целевом множестве и исходящая из пересечения заданной прямой и границы множества достижимости. потребовав, чтобы задаваемая прямая проходила через начало координат и требуемое начальное состояние u_0 , и установив продолжительность временного интервала, можно получить траекторию (если она существует), имеющую начало в $u(0)$, лежащем на границе множества достижимости.

Далее должны быть проверены условия, определяющие, действительно ли полученная траектория является оптимальной по быстродействию.

Первый шаг процедуры приближения заключается в делении временного интервала $[0, t_f]$ на N подынтер-

валов длиной T , где

$$T = \frac{t_f}{N}. \quad (5.51)$$

Далее, предполагаем, что компоненты управляющего вектора $\mathbf{u}(t)$ на каждом подынтервале постоянны. Это предположение, по существу, означает, что мы ищем кусочно-постоянную аппроксимацию для оптимального управления. При этом предположении определим последовательность постоянных векторов управления $\mathbf{u}(k)$ следующим образом:

$$\mathbf{u}(k) = \mathbf{u}(t), \quad (k-1)T \leq t < kT, \quad k=1, 2, \dots, N. \quad (5.52)$$

Еще одно предположение, которое делается в этой процедуре приближения, заключается в том, что ограничения на состояние и на управление накладываются только в дискретные моменты времени — один раз на каждом подынтервале. Так, ограничения на состояние будут иметь вид

$$C_y \mathbf{u}(kT) \leq \min \mathbf{v}_y(t) \equiv \mathbf{v}_y(k), \quad kT \leq t \leq (k+1)T, \\ k=0, \dots, N-1. \quad (5.53)$$

При этой формулировке ограничения на состояние вводятся для критической точки подынтервала.

Формулировка ограничений на управление делится на две части. Первая часть включает ограничения в виде простых границ вектора управления на подынтервале и задается следующим образом:

$$\mathbf{u}(k) \leq \mathbf{V}_u(k), \quad k=1, 2, \dots, N, \quad (5.54)$$

$$\mathbf{u}(k) \geq \mathbf{V}_L(k), \quad k=1, 2, \dots, N, \quad (5.55)$$

где $\mathbf{V}_u(k)$ и $\mathbf{V}_L(k)$ — соответственно верхний и нижний пределы $\mathbf{u}(k)$ на k -м подынтервале. Если допустимая область для управлений предполагается замкнутой и ограниченной, то в любой задаче ограничения в форме (5.54) и (5.55) либо существуют, либо могут быть наложены. В описываемом вычислительном алгоритме полезно иметь ограничения в этой форме. Вторая часть формулировки ограничений управления включает все ограничения, охватывающие линейные комбинации ком-

понент вектора управления и может быть представлена в виде

$$C_u u(k) \leq v_u(k), \quad k=1, 2, \dots, N, \quad (5.56)$$

где C_u — $(q \times r)$ -мерная матрица ограничений и $v_u(k)$ — q -мерный вектор ограничений для k -го подынтервала (см. (5.45)).

Описание целевого множества в (5.47) и (5.48) также разделено на две части. Первая состоит из простых границ

$$y(t_f) \leq X_u, \quad (5.57)$$

$$y(t_f) \geq X_l, \quad (5.58)$$

где X_u и X_l — n -мерные векторы, представляющие соответственно верхний и нижний пределы вектора $y(t_f)$. Вторая часть ограничений целевого множества состоит из ограничений, наложенных на линейные комбинации компонент вектора $y(t_f)$ и может быть записана в виде (см. также (5.47))

$$Gy(t_f) = v_g, \quad (5.59)$$

где G есть $(g \times n)$ -мерная матрица ограничений и v_g — g -мерный вектор ограничений.

Рассматриваемая постановка задачи требует, чтобы начальное состояние $y(0)$ принадлежало траектории, заканчивающейся на целевом множестве, и лежало на прямой, заданной в пространстве состояний. Пусть P есть n -мерный единичный вектор, направленный в y_0 . Прямая, проходящая через начало в направлении вектора P , может быть задана в виде пересечения $n-1$ гиперплоскостей. Таким образом, это ограничение можно представить в виде

$$Ey(0) = 0, \quad (5.60)$$

где E — $[(n-1) \times n]$ -мерная матрица ограничений. Элементы матрицы E не определены однозначно, так как единственное условие, которое должно быть соблюдено, заключается в том, что вектор P должен лежать в каждой гиперплоскости, определяемой строкой матрицы E , т. е.

$$EP = 0. \quad (5.61)$$

Таким образом, матрица E может быть составлена из любой совокупности линейно независимых строк, лишь бы было справедливо (5.61).

Теперь надо ввести все ограничения, наложенные на последовательность векторов управления $\mathbf{u}(k)$ и конечное состояние $\mathbf{y}(t_f)$. Это может быть сделано, так как состояние системы в любой момент jT можно выразить через последовательность векторов управления и конечное состояние в виде

$$\mathbf{y}(jT) = e^{-A(N-j)T} \mathbf{y}(t_f) + \sum_{k=j+1}^N \left[-e^{-A(k-j)T} \int_{-T}^0 e^{-As} B ds \right] \mathbf{u}(k). \quad (5.62)$$

Обозначая множитель при $\mathbf{y}(t_f)$ через $E_A(N-j)$ и множитель при $\mathbf{u}(k)$ через $D(k-j)$, можно записать

$$\mathbf{y}(jT) = E_A(N-j) \mathbf{y}(t_f) + \sum_{k=j+1}^N D(k-j) \mathbf{u}(k). \quad (5.63)$$

Используя (5.63), можно записать ограничения на состояния (5.53) в виде

$$C_y E_A(N-j) \mathbf{y}(t_f) + \sum_{k=j+1}^N C_y D(k-j) \mathbf{u}(k) \leq \mathbf{v}_y(j), \quad j=0, 1, 2, \dots, N-1, \quad (5.64)$$

и ограничение (5.60) в виде

$$E E_A(N) \mathbf{y}(t_f) + \sum_{k=1}^N E D(k) \mathbf{u}(k) \leq 0, \quad (5.65)$$

$$-E E_A(N) \mathbf{y}(t_f) - \sum_{k=1}^N E D(k) \mathbf{u}(k) \leq 0, \quad (5.66)$$

где два условия использованы для преобразования условий в форме равенств в условия в форме неравенств.

Полную совокупность ограничений от (5.54) до (5.59) и (5.64) до (5.66) теперь можно объединить в одно матричное неравенство, содержащее $(Nr+n)$ -мерный

перменный вектор \mathbf{u}_c , где

$$\mathbf{u}_c = \begin{pmatrix} \mathbf{u}(N) \\ \mathbf{u}(N-1) \\ \vdots \\ \mathbf{u}(1) \\ \mathbf{x}(t_f) \end{pmatrix}. \quad (5.67)$$

Запишем это ограничивающее неравенство в виде

$$A_i \mathbf{u}_c \leq \mathbf{u}_A. \quad (5.68)$$

Любые значения вектора \mathbf{u}_c , удовлетворяющие ограничениям (5.68), приводит к допустимым управлениям и требуемым конечным состояниям, что соответствует допустимым траекториям, исходящим из начальных состояний, расположенных внутри множества достижимости A_f на прямой, заданной уравнением (5.60).

Чтобы получить множество значений вектора \mathbf{u}_c , соответствующего начальному состоянию, расположенному на границе множества достижимости, необходимо и достаточно получить максимум функционала z , где

$$z = \mathbf{P}^T \mathbf{y}(0) = \mathbf{P}^T E_A(N) \mathbf{y}(t_f) + \sum_{k=1}^N \mathbf{P}^T D(k) \mathbf{u}(k). \quad (5.69)$$

Это выражение можно представить в виде

$$z = \mathbf{C}^T \mathbf{u}_c. \quad (5.70)$$

При данном значении z уравнение (5.69) представляет в пространстве состояний гиперплоскость, ортогональную \mathbf{P} . Максимизируя z при условии (5.68), находим последовательность управлений и конечное состояние, соответствующее начальному состоянию $\mathbf{y}(0)$, находящемуся на максимальном расстоянии от начала, но все же на множестве достижимости. Так как множество достижимости — выпуклое, то это состояние вполне определено.

Таким образом, задача определения \mathbf{u}_c , которое приводит к траектории, ведущей к пересечению прямой и границы множества достижимости, является задачей линейного программирования, максимизирующей (5.70) при условии (5.68).

Учитывая ограничения в форме неравенств и отсутствие у компонент вектора ограничений на знак, естественно рассматривать вышеприведенную задачу линейного программирования как двойственную задачу линейного программирования (см. гл. 2) и получить ее решение путем решения соответствующей задачи линейного программирования, заданной следующим образом: минимизировать $y = \mathbf{u}_f^T \mathbf{w}$ при условии

$$A_f^T \mathbf{w} = \mathbf{c}, \quad \mathbf{w} \geq 0, \quad (5.71)$$

где \mathbf{w} — двойственная векторная переменная.

Теория двойственности в линейном программировании позволяет получить решение для \mathbf{u}_f , когда найден оптимум \mathbf{w} . Задача (5.71) может быть решена с помощью симплексного метода (см. гл. 3).

Используя описанную процедуру и фиксированный временной интервал $[0, t_f]$, можно найти пересечение прямой и множества достижимости. Соотношение между этой точкой и требуемым начальным состоянием \mathbf{y}_0 можно определить путем сравнения скалярной величины целевой функции z и требуемого значения z^* , где

$$z^* = \mathbf{P}^T \mathbf{y}_0. \quad (5.72)$$

Когда z больше z^* , требуемое начальное состояние лежит внутри множества достижимости и временной интервал надо уменьшить. При z меньшем, чем z^* , временной интервал должен быть увеличен.

Для установления нового временного интервала удобно применить линейную интерполяцию. Пусть t_{f1} обозначает конец временного интервала, когда $z_1 < z^*$, и пусть t_{f2} — конец другого временного интервала, когда $z_2 > z^*$. Тогда новое время окончания процесса можно выбрать равным

$$t_f = t_{f1} + \frac{z^* - z_1}{z_2 - z_1} (t_{f2} - t_{f1}). \quad (5.73)$$

Используя подобную итерацию, мы либо найдем время окончания процесса при соответствующем значении z , лежащем в пределах допустимой ошибки для z^* , либо достигнем точки, после которой дальнейшее увеличение t_f не вызывает увеличения z . Последний случай (обычно указывающий на неединственность решения) имеет

место, когда требуемое начальное состояние $y(0)$ лежит вне области существования решений, т. е. вне множества достижимости для бесконечного интервала времени.

Таким образом, могут быть определены, если они существуют, временной интервал, конечное состояние, управление и траектория при $y(0)$, лежащем на границе множества достижимости. Остается доказать, что эта аппроксимация оптимальна по времени. Это можно сделать экспериментально, проверяя интервалы времени более короткие, чем оптимальный временной интервал, и убеждаясь в том, что значения z меньше z^* . В тех случаях, когда можно доказать, что множество достижимости удовлетворяет отношению включения второго рода, единственность решения (выявленная при решении задачи линейного программирования) гарантирует оптимальность по быстродействию.

В качестве примера рассмотрим систему четвертого порядка, описываемую следующим образом:

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \dot{y}_3 \\ \dot{y}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -2 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \quad (5.74)$$

Пусть вектор управления подчинен следующим ограничениям:

$$\left. \begin{aligned} 0,2e^{-t} - 1,3 &\leq u_1(t) \leq 1,0 - 0,2e^{-t}, \\ 0,2e^{-t} - 1,2 &\leq u_2(t) \leq 1,1 - 0,2e^{-t}, \\ |u_1(t)| + |u_2(t)| &\leq 1,0 - e^{-t}. \end{aligned} \right\} \quad (5.75)$$

Приведенные ограничения выбраны так, чтобы они имели общий характер. Заметим, что для компонент вектора управления установлены зависящие от времени несимметричные границы, так же как и зависящая от времени граница для суммы их абсолютных величин.

Пусть даны следующие ограничения на состояние:

$$\left. \begin{aligned} -0,6 &\leq y_2 \leq 0,7, \\ -0,6 &\leq y_4 \leq 0,7, \\ -0,5 &\leq y_2 \leq 0,5, \\ -0,5 &\leq y_4 \leq 0,5, \end{aligned} \right\} \begin{array}{l} 0 \leq t \leq 1, \\ t > 1. \end{array} \quad (5.76)$$

Таким образом, имеем систему зависящих от времени разрывных ограничений.

Пусть целевое множество задано следующим образом:

$$\left. \begin{aligned} -0,02 \leq y_1(t_f) \leq 0,02, \\ y_2(t_f) = 0, \\ -0,05 \leq y_3(t_f) \leq 0,05, \\ y_4(t_f) = 0, \\ |2y_1(t_f)| + |y_3(t_f)| \leq 0,06. \end{aligned} \right\} \quad (5.77)$$

Таким образом, некоторые компоненты заданы точно, а другие компоненты подчинены простым и комбинированным ограничениям.

Более детальное рассмотрение преобразований, необходимых для приведения задачи к виду, пригодному для непосредственного применения симплекс-алгоритма, предоставляется читателю в качестве упражнения.

Интересующая нас задача управления, оптимального по быстродействию, заключается в отыскании допустимого управления, которое в минимальное время переводит систему из

$$y(0) = \begin{bmatrix} 0,50 \\ 0 \\ 0,75 \\ 0 \end{bmatrix} \quad (5.78)$$

в целевое множество при ограничениях, наложенных как на управление, так и на состояние.

Для осуществления предложенной процедуры была применена ЦВМ и использован симплекс-алгоритм (гл. 3). Результаты, приведенные на рис. 5.5 и 5.6, были получены при 20-шаговой аппроксимации.

Ступенчатая аппроксимация u_1 и u_2 показана на рис. 5.5 и 5.6. Взято было конечное состояние

$$y(t_f) = \begin{bmatrix} 0,005 \\ 0 \\ 0,05 \\ 0 \end{bmatrix}, \quad (5.79)$$

которое находится на границе целевого множества. При допустимой величине ошибки для значения целевой функции в 0,002 найденное соответствующее начальное значение было равно

$$y(0) = \begin{bmatrix} 0,50128 \\ 0 \\ 0,75191 \\ 0 \end{bmatrix}, \quad (5.80)$$

т. е. близко к требуемому. Выбранный интервал времени был $[0, 1,768]$. Он является минимальным, если учесть, что решение единственное и множество достижимости

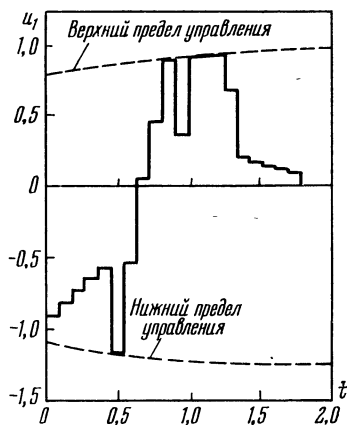


Рис. 5.5. Приближение для первой компоненты вектора управления, оптимального по быстродействию [3].

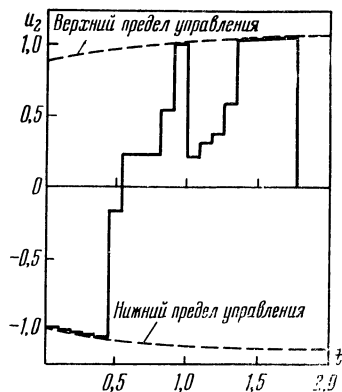


Рис. 5.6. Приближение для второй компоненты вектора управления, оптимального по быстродействию [3].

удовлетворяет отношению включения второго рода. (Можно показать, что для этого достаточно проверки условий теоремы (5.1) в крайних точках выпуклого целевого множества.)

5.7. Использование обобщенного программирования

Задача линейного программирования была сформулирована в гл. 2 и 3. Существует вариант более общей постановки задачи линейного программирования, называемый задачей обобщенного программирования. Он был предложен Данцигом и Вулфом [14, гл 22]. В задаче обобщенного программирования некоторые из столбцов матрицы ограничений могут быть переменными. В этом случае они должны быть определены в процессе решения задачи. Задача обобщенного программирования применима, в частности, для формулировки в виде задачи математического программирования линейной задачи оптимального управления. Это детально будет рассмотрено в данном параграфе. Идея применения обобщенного программирования к линейным задачам оптимального управления была предложена Данцигом [15] и рассматривалась в диссертациях Уэспи [1] и Ван-Слайка [16]. Однако никем не было предложено прямое численное решение задачи оптимального управления, сформулированной в виде задачи обобщенного программирования. Этот метод решения недавно получил Джизмаджан [7, 19]. Основные идеи этого алгоритма будут описаны в данном параграфе.

Вначале приведем постановку задачи обобщенного программирования, а далее рассмотрим его применение для решения задачи оптимального управления.

Задачу обобщенного программирования можно сформулировать следующим образом [7, 14, 19]: выбрать в выпуклом множестве $C \subseteq E^n$ вектор \mathbf{P} так, чтобы максимизировать z при условии

$$U_0 z + \mathbf{P}\mu = \mathbf{S}, \quad \mu \geq 0, \quad (5.81)$$

где U_0 и \mathbf{S} — заданные n -мерные векторы, а μ — скаляр. Итак, надо выбрать такой вектор \mathbf{P}^* или такую выпуклую комбинацию векторов \mathbf{P}^{i*} из множества C , чтобы линейные уравнения были удовлетворены, т. е. чтобы

$$U_0 z + \mathbf{P}^* = \mathbf{S} \quad (5.82)$$

или

$$U_0 z + \sum_i \mathbf{P}^{i*} \mu_i = \mathbf{S}, \quad \sum \mu_i = 1, \quad \mu_i \geq 0, \quad (5.83)$$

и чтобы полученное значение z было максимальным среди всех значений, даваемых любыми элементами множества C , удовлетворяющими этим линейным уравнениям. Заметим, что если некоторое множество векторов P^i находится в C , то и любая выпуклая комбинация элементов этого множества также находится в C . Следовательно, уравнения (5.82) и (5.83) эквивалентны, когда

$$P^* = \sum_i P^i \mu_i,$$

$$\sum \mu_i = 1, \quad \mu_i \geq 0.$$

Процедура решения основана на предположении о том, что вначале в нашем распоряжении имелось n выборок $P^i \in C$ таких, что следующая линейная программа, называемая *главная ограниченная программа*:

$$\max_{\mu} z \quad U_0 z + P^1 \mu_1 + \dots + P^n \mu_n = S,$$

$$\mu_1 + \dots + \mu_n = 1, \quad \mu_i \geq 0, \quad (5.84)$$

имела единственное допустимое невырожденное решение при базисе, определенном следующим образом:

$$B^0 = \begin{bmatrix} U_0 & P^1 & \dots & P^n \\ 0 & 1 & \dots & 1 \end{bmatrix}$$

и являющимся неособенным (по определению). Для каждого $P^i \in C$

$$P^0 = \sum_i P^i \mu_i^0$$

(где μ_i^0 есть решение задачи (5.84)) принадлежит C и является допустимым решением задачи (5.81), но не обязательно оптимальным решением.

Чтобы проверить P^0 (и тем самым любое решение задачи (5.81), порожденное базисом) на оптимальность, надо найти вектор-строку $\pi = \pi^0$, удовлетворяющий уравнению

$$\pi^0 B^0 = (1, 0, \dots, 0). \quad (5.85)$$

Зная π , находим такой вектор P^{n+1} , который не обязательно является единственным, и такую величину δ , что

$$\delta = \pi^0 P^{n+1} = \min_{P \in C} \pi^0 P, \quad (5.86)$$

где

$$\bar{\mathbf{P}} = \begin{bmatrix} \mathbf{P} \\ 1 \end{bmatrix}.$$

Если $\delta=0$, то решение является оптимальным.

Если $\delta < 0$, то уравнение (5.84) дополняется членом с \mathbf{P}^{n+1} и решается новая линейная программа.

Общая k -я итерация начинается с решения главной ограниченной программы:

максимизировать z при условии

$$\mathbf{U}_0 z + \sum_{i=1}^{n+k} \mathbf{P}^i \mu_i = \mathbf{S}, \quad \sum_i \mu_i = 1, \quad \mu_i \geq 0. \quad (5.87)$$

Пусть B^k есть оптимальный базис линейной программы (5.87) и π^k — двойственная (оптимальная) переменная к задаче (5.87), которая определена аналогично (5.85). Тогда δ^k и \mathbf{P}^{n+k+1} определяются из следующей подзадачи:

найти

$$\delta^{k+1} = \pi^k \mathbf{P}^{n+k+1} = \min_{\mathbf{P} \in C} \pi^k \bar{\mathbf{P}}. \quad (5.88)$$

Если $\delta^{n+1}=0$, то решение, полученное после k -й итерации главной задачи, является оптимальным. Если $\delta^{k+1} < 0$, то \mathbf{P}^{n+k+1} можно присоединить к (5.87) и решение задачи (5.81) будет улучшено.

Величина $-\delta^{k+1}$ есть максимальная величина, на которую можно улучшить z^k . Таким образом, $z^k - \delta^{k+1}$ образует верхнюю границу оптимального решения задачи (5.81). Известно, что эти оценки верхних границ могут существенно меняться от одной итерации к следующей. Поэтому выбирается наименьшая из таких оценок для всех итераций, включая очередную.

Как будет показано далее, применение обобщенного программирования для решения задачи оптимального управления порождает подзадачу, называемую задачей параметрического программирования [7].

Задача параметрического линейного программирования определяется следующим образом (рассматривается

только частный случай, однако он достаточен для данного рассмотрения):

$$\left. \begin{array}{l} \text{найти вектор } \mathbf{x}^*(t), \\ \text{минимизирующий } \boldsymbol{\alpha}^T(t)\mathbf{x} \\ \text{при условии } A\mathbf{x} = \mathbf{b}, x_i \geq 0; i = 1, \dots, n; t \in [T_1, T_2]. \end{array} \right\} \quad (5.89)$$

Аналогично, задача *параметрического квадратичного программирования* определяется следующим образом (вновь частный, но достаточный для нас случай):

$$\left. \begin{array}{l} \text{найти вектор } \mathbf{x}^*(t), \\ \text{минимизирующий } \boldsymbol{\alpha}^T(t)\mathbf{x} + \mathbf{x}^T D \mathbf{x} \\ \text{при условии } A\mathbf{x} = \mathbf{b}, x_i \geq 0; i = 1, \dots, n; t \in [T_1, T_2]. \end{array} \right\} \quad (5.90)$$

В обоих случаях A — это $(m \times n)$ -мерная матрица, \mathbf{b} — заданный m -мерный вектор, D — $(n \times n)$ -мерная положительно полуопределенная матрица (см. гл. 3), \mathbf{x} — n -мерный вектор переменных и

$$\boldsymbol{\alpha}^T(t) = [\alpha_1(t), \dots, \alpha_k(t), \dots, \alpha_n(t)] \quad (5.91)$$

— заданная векторная функция, каждая компонента которой является решением некоторого однородного линейного дифференциального уравнения с постоянными вещественными коэффициентами, которые могут зависеть от k . Такие функции $\alpha_k(t)$ имеют вид

$$\alpha_k(t) = \sum_{i=1}^n P_{ki}(t) e^{s_{ki}t}, \quad (5.92)$$

где $P_{ki}(t)$ — полином с вещественными коэффициентами степени m_{ki} , причем

$$\sum_{i=1}^n m_{ki} = n, \quad (5.93)$$

а s_{ki} — постоянные. Если s_{ki} — комплексная при i нечетном, s_{ki+1} будет с ней сопряженная и $P_{ki}(t) = P_{ki+1}(t)$. Отсюда следует, что $\alpha_k(t)$ — вещественные функции t .

Алгоритм и основные теоремы, связанные с решением задач этого класса, были разработаны Джизмаджаном [7, 20].

Сформулируем теперь рассмотренные задачи управления в форме обобщенных программ, а затем покажем, что подзадачи являются задачами параметрического программирования. Задача управления может быть сформулирована следующим образом:

$$\text{минимизировать } J = \int_0^T \dot{y}_0(t) dt = y_0(T), \quad y_0(0) = 0 \quad (5.94)$$

$$\left. \begin{array}{l} \text{при условии } \dot{\mathbf{y}}(t) = F\mathbf{y}(t) + G\mathbf{u}(t), \\ \mathbf{y}(0) \in S_0 \text{ (начальное состояние),} \\ \mathbf{y}(T) \in S_T \text{ (целевое состояние).} \end{array} \right\} \quad (5.95)$$

На вектор управления наложено следующее дополнительное ограничение:

$$\mathbf{u}(t) \in U = \{\mathbf{u} \mid A\mathbf{u} \geq \mathbf{b}\}, \quad (5.96)$$

где \mathbf{y} — n -мерный вектор, \mathbf{u} — m -мерный вектор F — $(n \times n)$ -мерная вещественная матрица, G — $(n \times m)$ -мерная вещественная матрица.

Далее,

$$\dot{y}_0(t) = \mathbf{f}_0^T \mathbf{y}(t) + \mathbf{g}_0^T \mathbf{u}(t) + h \sum_{i=0}^m |u_i(t)| + \mathbf{u}^T(t) Q \mathbf{u}(t), \quad (5.97)$$

где \mathbf{f}_0 — фиксированный вещественный n -мерный вектор, \mathbf{g}_0 — фиксированный вещественный m -мерный вектор, h — вещественная постоянная и Q — вещественная $(n \times n)$ -мерная матрица.

Прежде всего рассмотрим строго линейные функционалы стоимости, т. е. примем $h=0$ и $Q=0$.

Положим

$$\left. \begin{array}{l} \bar{F} = \left[\begin{array}{c|c} 0 & \mathbf{f}_0^T \\ \vdots & \vdots \\ 0 & F \end{array} \right] - [(n+1) \times (n+1)]\text{-матрица,} \\ \bar{G} = \left[\begin{array}{c} \mathbf{g}_0^T \\ G \end{array} \right] - [(n+1) \times m]\text{-матрица.} \end{array} \right\} \quad (5.98)$$

Тогда полностью линейную систему можно представить следующим образом:

$$\dot{\bar{\mathbf{y}}}(t) = \bar{F}\bar{\mathbf{y}}(t) + \bar{G}\mathbf{u}(t), \quad (5.99)$$

где

$$\bar{\mathbf{y}} = \begin{bmatrix} y_0 \\ \mathbf{y} \end{bmatrix}.$$

Если даны векторная функция $\mathbf{u}^i(t)$ и начальное состояние $\bar{\mathbf{y}}^i(0)$, то решение уравнения (5.99) для $t=T$ будет равно

$$\bar{\mathbf{y}}(T) = e^{T\bar{F}}\bar{\mathbf{y}}^i(0) + \int_0^T e^{(T-t)\bar{F}}\bar{G}\mathbf{u}^i(t) dt \quad (5.100)$$

и решение уравнения (5.95) будет равно

$$\mathbf{y}(T) = e^{TF}\mathbf{y}^i(0) + \int_0^T e^{(T-t)F}G\mathbf{u}^i(t) dt. \quad (5.101)$$

Введем множества

$$\begin{aligned} \bar{S}_T &= S_T - S_0^F, \\ S_0^F &= \{\mathbf{v} | \mathbf{y} \in S_0, \mathbf{v} = e^{TF}\mathbf{y}\}. \end{aligned}$$

Можно показать [7], что при выпуклых S_0 и S_T выпуклыми будут и S_0^F и \bar{S}_T . Без ограничения общности, можно положить

$$\mathbf{y}(0) = 0 \quad (5.102)$$

и

$$\mathbf{y}_0(0) = 0.$$

Определив векторные операторы от управления $\mathbf{P} = \mathbf{P}[u(t)]$ и $\bar{\mathbf{P}} = \bar{\mathbf{P}}[u(t)]$ следующим образом:

$$\mathbf{P} = \int_0^T e^{(T-t)F}G\mathbf{u}(t) dt \quad (5.103)$$

и

$$\bar{\mathbf{P}} = \int_0^T e^{(T-t)\bar{F}}\bar{G}\mathbf{u}(t) dt, \quad (5.104)$$

введем множества C и \bar{C} :

$$C \equiv \left\{ \mathbf{P} \mid \mathbf{u}(t) \in U, \mathbf{P} = \int_0^T e^{(T-t)F} \mathbf{G} \mathbf{u}(t) dt \right\},$$

$$\bar{C} \equiv \left\{ \bar{\mathbf{P}} \mid \mathbf{u}(t) \in U, \bar{\mathbf{P}} = \int_0^T e^{(T-t)\bar{F}} \bar{\mathbf{G}} \mathbf{u}(t) dt \right\}.$$

Обозначим $U_0^T = (1, 0, \dots, 0)$ и заметим, что первая компонента векторов \mathbf{S} множества \bar{S}_T по определению равна нулю. Заметим также, что первая компонента вектора $\bar{\mathbf{P}}$ представляет цену применения управления (и соответствующей траектории), порождающего $\bar{\mathbf{P}}$. Таким образом, мы ищем вектор $\bar{\mathbf{P}} \in \bar{C}$, векторную функцию $\mathbf{u}(t)$, порождающую $\bar{\mathbf{P}}$, и вектор $\bar{\mathbf{S}} \in \bar{S}_T$, обеспечивающие

$$\max_{\mathbf{P} \in C} \lambda, \quad \mu, \nu \geq 0 \quad (5.105)$$

при условии $U_0 \lambda + \bar{\mathbf{P}} \mu = \bar{\mathbf{S}} \nu, \mu = 1, \nu = 1,$

где μ и ν — скаляры. Максимизация λ эквивалентна минимизации $J[\mathbf{u}(t)]$ — первой компоненты вектора $\bar{\mathbf{P}}$, порождаемого векторной функцией $\mathbf{u}(t)$. Так как $\bar{\mathbf{P}}$ должен быть взят из выпуклого множества \bar{C} и $\bar{\mathbf{S}}$ должен быть взят из выпуклого множества \bar{S}_T , то приведенная выше формулировка представляет собой обобщенную программу типа Данцига — Вулфа [14].

Решение задачи обобщенного программирования содержит вектор \mathbf{P} в достижимом множестве R_T , управляющую функцию $\mathbf{u}(t)$ в достижимой области управления U , порождающую \mathbf{P} , и вектор \mathbf{S} в множестве ограниченных конечных состояний S_T , так что

$$\mathbf{P} \equiv \mathbf{S}.$$

Это равенство обеспечивает перевод системы из любой начальной точки $\mathbf{y}(0) \in S_0$ в конечную точку $\mathbf{y}(T) \in S_T$ с помощью векторной функции $\mathbf{u}(t)$, выбранной из U . Таким образом, имеем допустимое управление. Минимизируя J по всем допустимым множествам \bar{P} и \bar{S} , можно найти допустимое решение при наименьшей цене. Это и есть оптимальное решение задачи непрерывного управления.

Чтобы завершить формулировку задачи обобщенного программирования, надо описать ее подзадачу [14, гл. 22]. Предположим, что для образования допустимого решения задачи (5.105) имеется не менее $n+2$ векторов \mathbf{P}^i и/или \mathbf{S}^i , так что задача

$$\left. \begin{aligned} \max_{\lambda, \mu, \nu} \lambda, \quad \mu, \nu \geq 0, \\ \mathbf{U}_0 \lambda + \mathbf{P}^1 \mu_1 + \mathbf{P}^2 \mu_2 + \dots + \mathbf{P}^j \mu_j = \mathbf{S}^{j+1} \nu_1 + \dots + \mathbf{S}^{j+p} \nu_p, \\ \mu_1 + \mu_2 + \dots + \mu_j = 1, \\ \nu_1 + \nu_2 + \dots + \nu_p = 1 \end{aligned} \right\} \quad (5.106)$$

разрешима и имеет вектор двойственного решения

$$\bar{\boldsymbol{\pi}}^T = (\pi_0, \boldsymbol{\pi}^T, \pi_{n+1}, \pi_{n+2}),$$

где $\boldsymbol{\pi}^T = (\pi_1, \dots, \pi_n)$. Тогда подзадача разбивается на две:

$$\min_{\bar{\mathbf{s}} \in \bar{\mathbf{S}}_T} \bar{\boldsymbol{\pi}}^T \begin{bmatrix} \bar{\mathbf{S}} \\ 0 \\ 1 \end{bmatrix}, \quad (5.107)$$

$$\min_{\bar{\mathbf{p}} \in \bar{\mathbf{C}}} \bar{\boldsymbol{\pi}}^T \begin{bmatrix} \bar{\mathbf{P}} \\ 1 \\ 0 \end{bmatrix}. \quad (5.108)$$

Решение задачи (5.107) зависит от вида множества $\bar{\mathbf{S}}_T$, простейшим случаем является задача с фиксированным концом, который состоит из единственного элемента. В этом случае подзадача (5.107) тривиальна и не нуждается в рассмотрении. Если $\bar{\mathbf{S}}_T$ — выпуклое многогранное множество, то (5.107) представляет собой линейную программу, которая должна решаться для каждой итерации основной задачи.

Подзадача в (5.108) может быть записана следующим образом:

$$\min_{\bar{\mathbf{p}} \in \bar{\mathbf{C}}} \bar{\boldsymbol{\pi}}^T \begin{bmatrix} \mathbf{P} \\ 1 \\ 0 \end{bmatrix} = \min_{\bar{\mathbf{p}} \in \bar{\mathbf{C}}} \bar{\boldsymbol{\pi}}^T \begin{bmatrix} \int_0^T e^{(T-t)\bar{\mathbf{F}}} \bar{\mathbf{C}} \mathbf{u}(t) dt \\ 1 \\ 0 \end{bmatrix}. \quad (5.109)$$

Так как требование $\bar{\mathbf{P}} \in \bar{\mathcal{C}}$ эквивалентно требованию $\mathbf{u}(t) \in U$ при всех значениях t , то (5.109) можно записать в виде

$$\min_{\mathbf{u}(t) \in U} \bar{\boldsymbol{\pi}}^T \begin{bmatrix} \int_0^T e^{(T-t)\bar{F}} \bar{G} \mathbf{u}(t) dt \\ 1 \\ 0 \end{bmatrix}$$

или, поскольку $\bar{\boldsymbol{\pi}}$ не зависит от t ,

$$\min_{\mathbf{u}(t) \in U} \left[\int_0^T (\boldsymbol{\pi}_0, \boldsymbol{\pi}^T) e^{(T-t)F} G \mathbf{u}(t) dt \right] + \pi_{n+1}. \quad (5.110)$$

Минимум интеграла достигается, когда подынтегральная функция будет минимальна в каждой точке. Пусть

$$\boldsymbol{\gamma}(t) = (\boldsymbol{\pi}_0, \boldsymbol{\pi}^T) e^{(T-t)\bar{F}} \bar{G} \quad (5.111)$$

представляет собой m -мерную векторную функцию. Тогда подзадача запишется в следующем виде:

$$\left. \begin{array}{l} \text{найти такое } \mathbf{u}(t) \in U, t \in [0, T], \\ \text{чтобы } \boldsymbol{\gamma}^T(t) \mathbf{u}(t) \text{ было минимально.} \end{array} \right\} \quad (5.112)$$

Как видно из (5.111), $\boldsymbol{\gamma}(t)$ является решением однородного линейного дифференциального уравнения $(n+1)$ -го порядка с постоянными коэффициентами. Так как нас интересуют только такие U , которые являются многогранными множествами, то (5.112) можно записать в виде

$$\left. \begin{array}{l} \min \boldsymbol{\gamma}^T(t) \mathbf{u}(t), \\ A \mathbf{u}(t) \geq b, \quad t \in [0, T] \end{array} \right\} \quad (5.113)$$

(отметим, что неравенство может быть заменено на обратное или на равенство без ограничения общности). Таким образом, решение $\mathbf{u}(t)$ подзадачи может быть получено с помощью метода параметрического линейного программирования [7, 20].

Пример. Рассматривается задача типа минимального расхода топлива. Показатель качества:

$$\min_{u(t)} \int_0^3 |u(t)| dt.$$

Уравнение состояния:

$$\begin{aligned} \dot{y}_1 &= y_2, \\ \dot{y}_2 &= u. \end{aligned}$$

Ограничение на управление:

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

Начальное и конечное состояния:

$$y(0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{и} \quad y(3) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Таким образом,

$$F = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad e^{Ft} = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix}$$

и

$$e^{F(T-t)} G = \begin{bmatrix} 1 & T-t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} T-t \\ 1 \end{bmatrix}.$$

Поэтому

$$P = \int_0^3 \begin{bmatrix} T-t \\ 1 \end{bmatrix} u(t) dt$$

и

$$U = \{u \mid |u| \leq 1\}.$$

Согласно определениям, приведенным в начале параграфа,

$$S_0^F = \left\{ e^{FT} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\} = \left\{ \begin{bmatrix} 1 & 3 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right\} = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

$$S_T = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\},$$

$$\bar{S}_T = \{S_T - S_0^F\} = \left\{ \begin{bmatrix} -1 \\ 0 \end{bmatrix} \right\}.$$

Для решения этой задачи был применен алгоритм вычислений, подробно описанный в [7]. Задача решалась на вычислительной машине ИБМ 360/67. Сходимость была достигнута после 40 итераций при использовании 16 разрядов. На рис. 5.7, а, б показана управляющая функция при числе итераций $k=2, 3, 4, 5, 10, 15, 30$ и 40. На рис. 5.8 показана цена $J(\bar{u}^k)$ для каждого числа итераций.

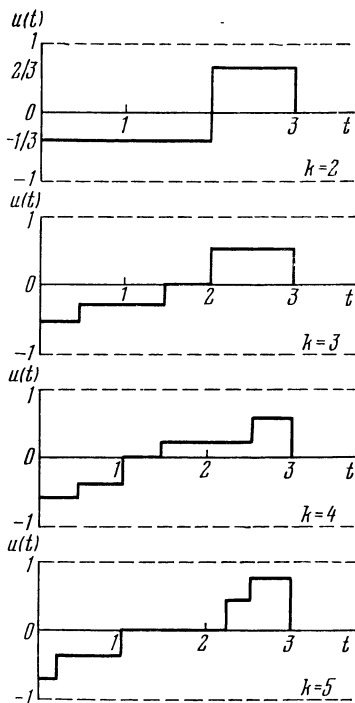


Рис. 5.7, а. Зависимость u от t при итерациях, соответствующих значениям $k=2, 3, 4, 5$ [7].

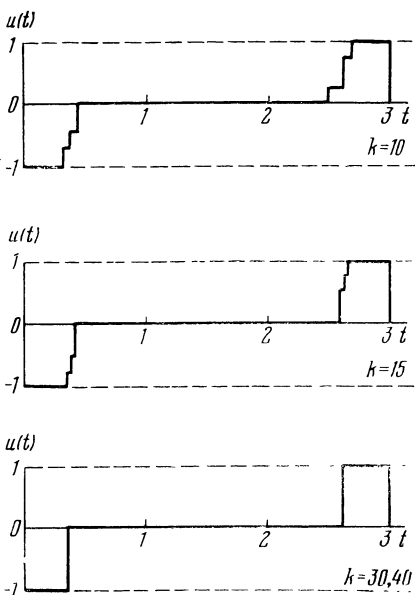


Рис. 5.7, б. Зависимость u от t при итерациях, соответствующих значениям $k=10, 15, 30, 40$ [7].

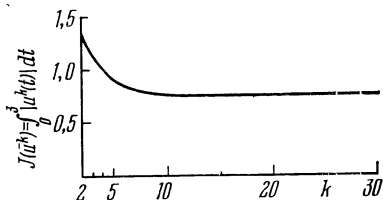


Рис. 5.8. Зависимость показателя качества от значения k [7].

5.8. Применение для вычисления оптимального управления ракетным ядерным реактором

Рассматриваемый ядерный ракетный реактор использует в качестве топлива водород. Рис. 5.9 схематически иллюстрирует ядерную ракетную движущую систему. Водород прокачивается через активную зону реактора, где тепло передается теплоносителю. Горячий водородный газ вырывается че-

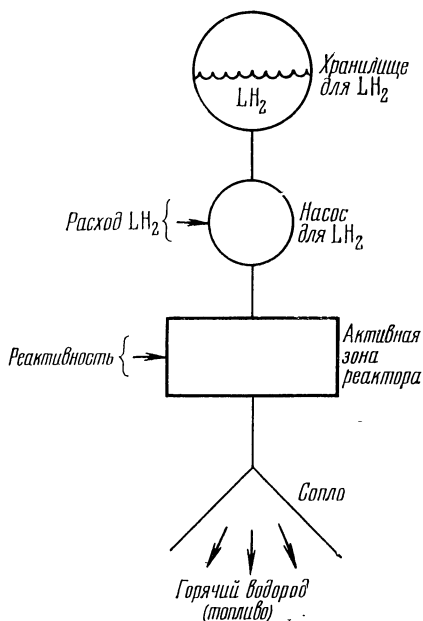


Рис. 5.9. Иллюстрирующая схема ракетного ядерного реактора [6].

рез сопло и создает реактивную движущую силу. Предполагается, что единственными переменными, которые могут управлять ядерным реактором, являются расход водорода u_2 и реактивность u_1 .

Поведение ядерного реактора можно приближенно представить следующими дифференциальными уравнениями:

$$\dot{Q} = \left(\frac{u_1 - \beta}{l} \right) Q + \lambda C, \quad (5.114)$$

$$\dot{C} = \frac{\beta}{l} Q - \lambda C, \quad (5.115)$$

где Q — уровень мощности реактора, C — концентрация эмиттеров запаздывающих нейтронов, u_1 — реактивность, u_2 — расход водорода, β — доля запаздывающих нейтронов, l — среднее время жизни нейтронов и λ — средняя постоянная распада эмиттеров запаздывающих нейтронов. Уравнения (5.114) и (5.115) являются классическими уравнениями кинетики образования нейт-

ронов, рассматриваемой как процесс с сосредоточенными параметрами [17].

Теплообмен между активной зоной реактора и водородом приближенно описывается уравнением

$$\dot{T} = \frac{Q}{M_c} - au_2 T, \quad (5.116)$$

где T — средняя температура активной зоны, M_c — эффективный запас реактивности в активной зоне и a — коэффициент теплопередачи. Уравнение (5.116) представляет процесс теплопередачи с сосредоточенными параметрами.

Рассматривается задача пуска реактора. Вначале реактор работает при относительно низком уровне мощности. Надо перевести реактор из этого режима низкого уровня мощности в режим уровня мощности и температуры, соответствующих максимуму тяги.

В течение этой пусковой фазы переменные состояния и управления подчинены различным ограничениям, которые определяются механическими причинами и соображениями надежности. Рассматриваются следующие ограничения:

$$\left. \begin{aligned} 0 &\leq Q \leq Q_{\max}, \\ 0 &\leq C, \\ 0 &\leq T \leq T_{\max}, \\ \dot{T} &\leq \dot{T}_{\max}, \\ -0,9\beta &\leq u_1 \leq 0,9\beta, \\ \dot{u}_1 &\leq \dot{u}_{1\max}, \\ u_{2\min} &\leq u_2 \leq u_{2\max}. \end{aligned} \right\} \quad (5.117)$$

Начальные граничные условия $[Q(0), C(0), T(0)]$ определяются стационарным низким уровнем мощности. Конечные значения мощности и температуры $[Q(t_1), T(t_1)]$ фиксированы критерием максимальной тяги. Конечное значение концентрации эмиттеров запаздывающих нейтронов $[C(t_1)]$ и самого времени окончания процесса (t_1) могут быть не фиксированы, но, конечно, подчинены условиям

$$0 \leq t_1, \quad 0 \leq C(t_1).$$

Наиболее важным критерием работы ракетного ядерного реактора является количество водорода, расходуемого в процессе перехода из одного режима в другой. Небольшой процент экономии водорода может дать большой процент возрастания полезной нагрузки. В частности, это справедливо потому, что экономится не только водородное топливо, но и сокращаются размеры бортовых водородных баков. По этой причине критерием оптимума является минимум величины J , определяемой следующим образом:

$$J = \int_0^{t_1} u_2(t) dt. \quad (5.118)$$

Процедура вычислений, выполняемых для получения решений задачи оптимизации ракетного ядерного реактора, основана на итерационном решении последовательности задач линейного программирования.

Рис. 5.10 схематически иллюстрирует этот алгоритм. Доказательство сходимости для ограниченного класса задач приводится в [18]. Первый шаг заключается в аппроксимации непрерывных дифференциальных уравнений и интегрального показателя качества системой дискретных алгебраических уравнений. Само собой разумеется, есть много методов получения этой аппроксимации, большинство из которых аналогично процедурам численного интегрирования. Здесь для дифференциальных уравнений процесса используется следующая аппроксимация второго порядка:

$$Q_{k+1} - Q_k = \frac{\Delta t}{2} \left\{ \left(\frac{u_{1k} - \beta}{l} \right) (Q_{k+1} + Q_k) + \lambda (C_{k+1} + C_k) \right\}, \quad (5.119)$$

$$C_{k+1} - C_k = \frac{\Delta t}{2} \left\{ \frac{\beta}{l} (Q_{k+1} + Q_k) - \lambda (C_{k+1} + C_k) \right\}, \quad (5.120)$$

$$T_{k+1} - T_k = \frac{\Delta t}{2} \left\{ \frac{1}{M_c} (Q_{k+1} + Q_k) - a u_{2k} (T_{k+1} + T_k) \right\}. \quad (5.121)$$

При получении уравнений (5.118), (5.119) и (5.120) принята ступенчатая аппроксимация для управлений, т. е. предполагается, что управления u_1 и u_2 являются

постоянными на интервалах времени Δt . Так как применена аппроксимация второго порядка, то ошибка от усечения рядов Тейлора на любом интервале от t до $t+\Delta t$

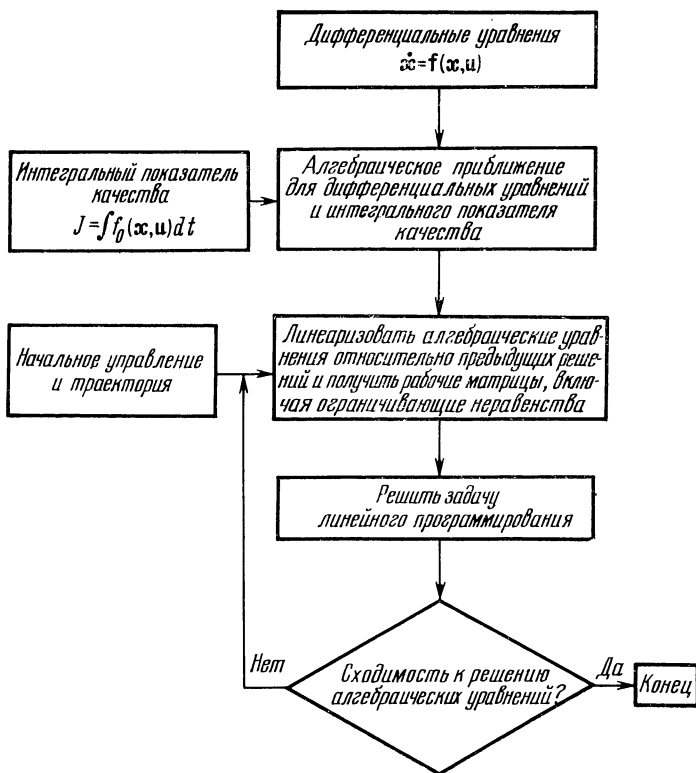


Рис. 5.10. Иллюстрирующая схема алгоритма линейного программирования для задач оптимального управления [6].

(или от k до $k+1$) пропорциональна $(\Delta t)^2$. Интервал Δt определяется следующим образом:

$$\Delta t = \frac{t_1 - t_0}{N}, \quad (5.122)$$

где N — число шагов по времени на интервале $t_1 - t_0$.

В результате того, что управления представлены ступенчатыми и постоянны на каждом интервале Δt , интег-

ральный показатель качества может быть записан в виде

$$J = \Delta t \sum_{k=1}^N u_{2k}. \quad (5.123)$$

Уравнения (5.119), (5.120), (5.121) и (5.123) — нелинейные и непосредственно непригодны для линейного программирования (см. гл. 2).

Чтобы применить процедуры линейного программирования, эти нелинейные уравнения надо линеаризировать. Линеаризацию можно получить, сохранив в нелинейных уравнениях первые два члена (постоянный и линейный) разложения в ряд Тейлора. Так как время t_1 не фиксировано, то Δt рассматривается как переменная.

Не считая ограничений скорости, линеаризация завершает формулировку задачи линейного программирования. С практической точки зрения ограничения скорости можно легко сформулировать в виде ограничивающих неравенств. Следующие неравенства предотвратят слишком быстрое нарастание температуры и реактивности:

$$u_{1_{k+1}} - u_{1_k} \leq \Delta t \dot{u}_{1_{\max}}, \quad (5.124)$$

$$T_{k+1} - T_k \leq \Delta t \dot{T}_{\max}. \quad (5.125)$$

Как видно из рис. 5.10, решение нелинейной задачи получается путем последовательного решения задач линейного программирования, причем каждая новая задача получается в результате линеаризации относительно предыдущего решения. Большинство мощных ЦВМ снабжены системами линейного программирования, что значительно упрощает программирование задач этого класса. Вычисления для данного случая были выполнены на ЦВМ ИБМ 360/40. Машины серии ИБМ/360 обеспечиваются системой математического программирования MPS/360, пригодной для решения задач линейного программирования. На языке этой системы MPS/360 потребовалось всего 16 операторов для решения задач линейного программирования, относящихся к ракетному ядерному реактору. Эта программа использует предыдущий оптимальный базис в качестве исходного базиса для каждой новой итерации.

Матрица коэффициентов и другие данные, относящиеся к линеаризованным алгебраическим уравнениям, показаны на рис. 5.11. Рекурсивный характер алгебраических уравнений и ограничений скорости в виде неравенств обуславливает образование блочной матрицы. Ненулевые коэффициенты этой матрицы вместе с коэффициентами столбца правой части и значениями границ для каждой переменной образуют входные данные для программы. При каждой итерации эти коэффициенты должны

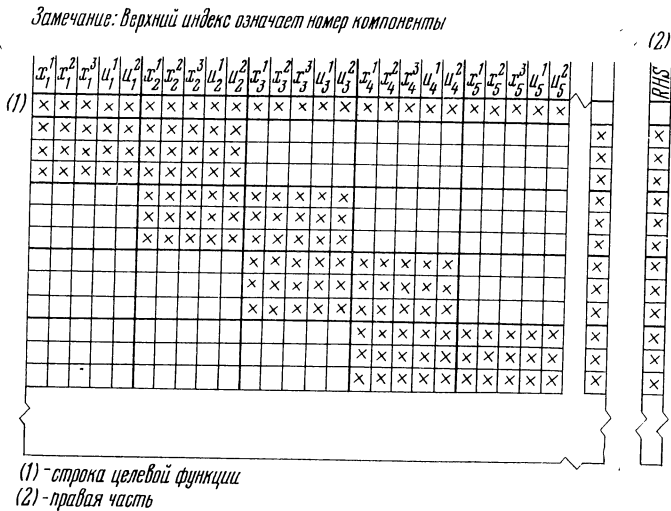


Рис. 5.11. Иллюстрация блочного вида рабочей матрицы при аппроксимации второго порядка [6].

строиться в должном формате для MPS/360. Так как матрица блочная, то данные могут образовываться сериями циклов DO в программе на Фортране. Алгоритм, показанный на рис. 5.10, был образован попеременной работой программ на языках Фортран и MPS/360. Показателем сходимости явилось то, что входные данные для программы MPS/360 оказывались оптимальными в соответствии со стандартными критериями точности, установленными в программе MPS/360.

Результаты вычислений для ракетного ядерного реактора были получены с помощью ранее описанного алгоритма линейного программирования.

На рис. 5.12 показаны траектории, полученные для случая отсутствия ограничений реактивности и скорости

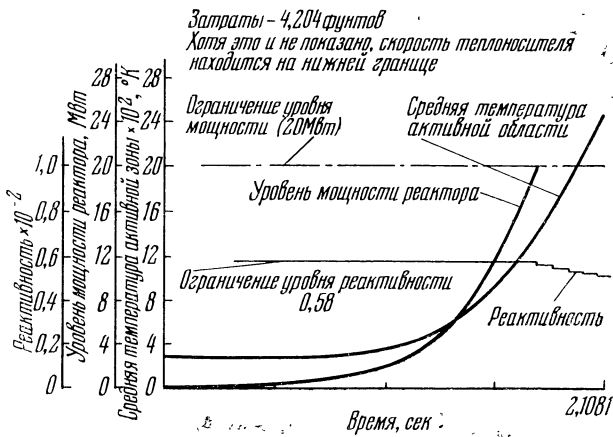


Рис. 5.12. Траектории ракетного ядерного реактора при ограничении уровня мощности (20 МВт) [6].

изменения температуры. Оптимальная стратегия заключается в том, что дается максимальная реактивность до тех пор, пока не будет достигнуто ограничение по уровню мощности. После этого для поддержания уровня мощности на границе ограничения реактивность должна быть уменьшена. Режим постоянного уровня мощности сохраняется до тех пор, пока выполняется граничное условие для конечной температуры. В течение всего процесса пуска скорость потока носителя тепла остается на уровне своей нижней границы. Этот процесс, конечно, соответствует абсолютному минимуму, поскольку интегральный показатель качества при этих условиях является равным своему абсолютному минимальному значению.

Данные, показанные на рис. 5.13, аналогичны приведенным на рис. 5.12. Однако, эффективный запас реактивности в активной зоне был искусственно уменьшен в 10 раз, так что температурное ограничение было дости-

гнуто до ограничения мощности. В этом случае оптимальный процесс является также процессом, оптимальным по быстродействию, но расход носителя тепла намного больше, чем в случае рис. 5.12. Более высокий

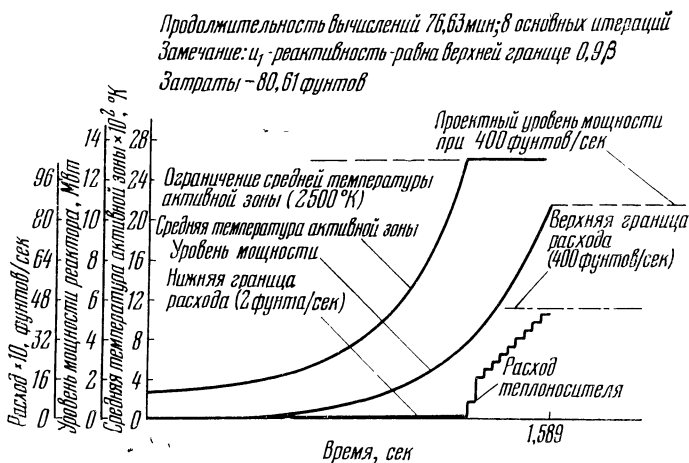


Рис. 5.13. Траектории ракетного ядерного реактора при быстрой температурной реакции [6].

расход носителя тепла является результатом того, что скорость потока водорода должна возрасти, чтобы на последней части траектории поддерживать температуру на уровне верхней границы. Оптимальность траектории, показанной на рис. 5.13, можно оспаривать, так как существует другая стратегия, при которой поддерживается температура на ее граничном значении. Можно было допустить сохранение скорости потока теплоносителя на нижней границе и использовать изменения реактивности для изменения уровня мощности Q так, чтобы удовлетворялось дифференциальное уравнение для температуры. Однако скорость изменения температуры на температурной границе равна нулю, так что уровень мощности должен был бы оставаться постоянным, т. е.

$$\dot{T} = 0 = \frac{Q}{M_c} - a u_2 \in_{in} T_{\max}.$$

Если уровень мощности остается постоянным, то граничное условие, определяющее конечный уровень мощности, не может быть выполнено.

Литература

1. C. M. W a e s p y, An Application of Linear Programming to Minimum Fuel Optimal Control, Ph. D. Thesis, UCLA, Los Angeles, 1967.
2. A. F. F a t h and T. J. H i g g i n s, Fixed-Time Fuel-Optimal Control of Linear State-Constrained Systems by Use of Linear Programming Techniques, 1968, JACC, pp. 462—467, Ann. Arbor, Mich.
3. A. F. F a t h, Approximation to the Time-Optimal Control of Linear State-Constrained Systems, 1968, JACC, pp. 962—969, Ann. Arbor, Mich.
4. A. F. F a t h, A computational Procedure for Fixed-Time Fuel-Optimal Control of Linear State-Costrained Systems, 1968, WESCON, Section 14, Los Angeles, Cal.
5. D. T a b a k, Application of Mathematical Programming in the Design of Optimal Control Systems, Ph. D. Thesis, University of Illinois, Urbana, 1967.
6. H. J. P r i c e and R. R. M o h l e r, Computation of Optimal Controls for a Nuclear Rocket Reactor. IEEE Trans. Nucl. Sci. NS-15, pp. 65—73 (1968).
7. G. S. J i z m a g i a n, Generalized Programming Solution of Continuous-Time Linear-System Optimal Control Problems, Ph. D. Thesis, Stanford University, Stanford, Cal., 1968.
8. R. O. B a r r, Jr., Computation of Optimal Controls by Quadratic Programming on Convex Reachable Sets, Ph. D. Thesis, University of Michigan, Ann. Arbor, Mich., 1968.
9. L. A. Z a d e h and C. A. D e s o e r, Linear System Theory, McGraw-Hill, New York, 1963. [Русский перевод: Л. Заде и Ч. Дезоер, Теория линейных систем, «Наука», 1970.]
10. B. C. K u o and D. T a b a k, A Nonlinear Programming Approach to Optimal Control of Discrete Systems with Constrained Sampling Times, 1968, JACC, pp. 918—922, Ann. Arbor, Mich.
11. R. A. V o l z, Optimal Control of Discrete Systems with Constrained Sample Times, 1967, JACC, pp. 16—22, Philadelphia, Pa.
12. W. H. C l o h e s s y and R. S. W i l t s h i r e, Terminal System for Satellite Rendezvous, J. Aerospace Sci. 27, pp. 653—658 (Sept. 1960).
13. LP/90 Usage Manual, 2nd ed., CEIR, Inc., Washington, D. C., July 31, 1963.
14. G. B. D a n t z i g, Linear Programming and Extensions, Princeton University Press, Princeton, N. J., 1963. [Русский перевод: Дж. Данциг, Линейное программирование, его применения и обобщения, «Прогресс», 1966.]
15. G. B. D a n t z i g, Linear Control Processes and Mathematical Programming, J. SIAM Control 4, pp. 56—60 (1966).
16. M. V a n S l y k e, Mathematical Programming and Optimal Control, Ph. D. Thesis, University of California, Berkeley, 1965.

17. M. A s h, Nuclear Reactor Kinetics, McGraw-Hill, New York, 1965.
18. J. B. R o s e n, Iterative Solution of Nonlinear Control Problems, J. SIAM Control 4, pp. 223—224 (1966).
19. G. S. J i z m a g i a n, Generalized Programming Solution of Optimal Control Problems, in Computing Methods in Optimization Problems, vol. 2 (L. A. Zadeh, L. W. Neustadt and A. V. Balakrishnan, eds.), Academic Press, New York, 1969, pp. 165—176.
20. G. S. J i z m a g i a n, An Algorithm for Parametric Linear Programming with Nonlinear Functions of the Parameter, 10th Meeting of the Institute of Management Sci., Atlanta, Georgia, Oct. 1969.
- 21*. И в а н и л о в Ю. П., П р о п о й А. И., О задачах динамического линейного программирования, Докл. АН СССР 198, № 8 (1971).

6.1. Введение

Общая формулировка задачи оптимального управления как задачи математического программирования для нелинейной системы с непрерывным временем вкратце была изложена в гл. 4 (см. (4.26)). В данной главе будут приведены несколько примеров вычислений, иллюстрирующих эту формулировку. Дополнительно будут приведены другие формулировки, необходимые для применения математического программирования к численному решению задач оптимального управления для нелинейных систем с непрерывным временем.

6.2. Управление процессом отравления ксеноном в ядерных реакторах

В качестве примера, иллюстрирующего применение математического программирования к решению задачи оптимального управления, сформулированной в (4.26), рассматривается задача оптимального управления включением ядерных реакторов при отравлении ксеноном [1—6]. Задача заключается в следующем.

Во время работы ядерного реактора в качестве осколков деления образуются некоторые виды ядер с очень большими вероятностями нейтронного поглощения. Наиболее сильными поглотителями нейтронов являются ксенон (Xe) и иод (I). Наиболее сильным является ксенон, который получается в результате радиоактивного распада иода. Когда реактор выключается и поток нейтронов практически уменьшается до нуля, ксенона вырабатывается больше, затем вследствие поглощения он исчезает. Таким образом, образуется пик количества ксенона. Так как период полураспада ксенона равен 9,2 часа, требуется очень много времени, чтобы количество ксенона уменьшилось настолько, чтобы был возможен нормальный режим работы реактора. Математически опи-

сать этот процесс можно следующим образом. Если рассматривать концентрацию ксенона и иода как переменные состояния, а поток нейтронов как переменную управления, то основные уравнения состояния, описывающие процесс, запишутся следующим образом [3]:

$$\left. \begin{aligned} \dot{x} &= -(\omega + r_0 u)x + g_1 y + g_2 u, \\ \dot{y} &= -y + u, \end{aligned} \right\} \quad (6.1)$$

где $g_1 = \gamma_1(\omega + r_0)$, $g_2 = \gamma_2(\omega + r_0)$, x — масштабированная, безразмерная концентрация ксенона, y — масштабированная, безразмерная концентрация иода, u — масштабированный, безразмерный поток ϕ/ϕ_0 , $\gamma_1 = \gamma_I/(\gamma_x + \gamma_I)$, $\gamma_2 = \gamma_x/(\gamma_x + \gamma_I)$, $r_0 = \frac{\sigma\phi_0}{\lambda_1}$, $\omega = \frac{\lambda_2}{\lambda_1}$, γ_x — относительный выход ксенона, образованного распадом, γ_I — относительный выход иода, образованного распадом, ϕ — поток тепловых нейтронов, ϕ_0 — единица измерения потока, λ_1 — постоянная распада иода, λ_2 — постоянная распада ксенона, σ — эффективное сечение поглощения ксенона для тепловых нейтронов.

Как видим, уравнения состояния (6.1) из-за члена ux в первом уравнении оказываются нелинейными. Будем выражать время в относительных единицах, взяв в качестве единицы, в данном случае, величину $\lambda_1^{-1} = 9,58$ часа.

Масштабирование всех переменных осуществляется так, чтобы начальными условиями были

$$x(0) = y(0) = 1$$

и

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

В момент завершения выключения, т. е. когда поток u уменьшен до нуля, уравнения (6.1) становятся линейными:

$$\left. \begin{aligned} \dot{x} &= -\omega x + g_1 y, \\ \dot{y} &= -y. \end{aligned} \right\} \quad (6.2)$$

Решение уравнений (6.2) может быть легко получено.

Если остановка происходит в момент $t = T$, когда x и y имеют соответственно значения $x(T)$ и $y(T)$, то

решение уравнений (6.2) для $t > T$ будет равно [3]

$$x(t) = x(T) \exp[-\omega(t-T)] + y(T) \left\{ \frac{g_1}{1-\omega} \exp[-\omega(t-T)] - \exp[-(t-T)] \right\}. \quad (6.3)$$

С помощью дифференцирования можно получить величину ксенонового пика, т. е. максимум $x(t)$:

$$x_p = Gy(T) \left[1 + F \frac{x(T)}{y(T)} \right]^{g_w}, \quad (6.4)$$

где $G = g_1 \omega^{w/(1-w)}$, $F = \frac{1-\omega}{g_1}$, $g_w = \frac{1}{1-\omega}$. Чем меньше величина x_p , тем раньше можно вновь осуществить пуск реактора. Этот фактор особенно важен в ядерных реакторах различных движущихся средств, где своевременный новый пуск может во многих случаях оказаться решающим.

Есть несколько путей постановки задачи управления. Ниже мы опишем два подхода.

1. Минимаксная задача [3]. Каким должен быть закон управления $u(t)$ в течение периода включения, $0 < t \leq T$, при заданной длительности T процесса выключения, чтобы был минимальным ксеноновый пик, зависящий от конечного состояния (см. (6.4)). Эта задача управления конечным состоянием была рассмотрена Эшем (Ash) [3], который применил для ее решения динамическое программирование [7]. На состояние и управление наложены следующие ограничения:

$$x \leq x_{\max}, \quad (6.5)$$

$$u \leq u_{\max}. \quad (6.6)$$

Из физических соображений ясно, что все рассматриваемые здесь переменные состояния и управления неотрицательны.

2. Задачи минимального времени [4, 5]. В этом случае конечное состояние при $t = T$ таково, что траектория x после этой точки достигает пика, величина которого задана $x_p = x_{\max}$. Множество возможных конечных точек образует в плоскости y, x целевую кривую, описываемую

следующим уравнением [4]:

$$x = g(y) = \left[\frac{x_{\max}}{(x_{\max}^\omega)^\omega} - \frac{(x_{\max}^\omega)^{1-\omega}}{1-\omega} \right] y^\omega - \frac{1}{1-\omega}. \quad (6.7)$$

Следовательно, задача заключается в том, чтобы достигнуть целевой кривой $g(y)$ за минимальное время. Робертс и Смит [5] показали, что задачи 1 и 2 — эквивалентны.

В этой работе, прежде чем применить современный метод оптимизации, задача была исследована с помощью классического подхода. Чтобы минимизировать ксеноновый пик x_p , надо найти переменные, от которых он зависит. Как видно из (6.4), x_p зависит только от конечного состояния $x(T)$ и $y(T)$. Теоретически существует бесконечное число процессов включения и законов управления, соответствующих данному конечному состоянию. Однако x_p во всех случаях будет одним и тем же. Поэтому, прежде чем искать оптимальный закон, который должен действовать в период выключения, надо задаться вопросом о том, каким должно быть конечное состояние $x(T)$, $y(T)$, чтобы x_p было минимально. Определяя частные производные выражения (6.4) для произвольного значения $x(T)$, найдем, что x_p будет минимально при условии

$$\frac{x(T)}{y(T)} = \frac{1}{F(g_\omega - 1)}. \quad (6.8)$$

Рассматривая (6.8) как уравнение целевой кривой в плоскости yx , можно обеспечить минимум ксенонового пика x_p независимо от конечного значения концентрации ксенона. Теперь можно представить задачу оптимального управления в следующем модифицированном виде.

По заданному начальному состоянию $[x(0), y(0)]$ найти закон оптимального управления $u(t)$, при котором целевое множество, описанное уравнением (6.8), достигалось бы за минимальное время.

Чтобы применить формулировку, приведенную в гл. 4, надо теперь уравнения состояния (6.1) переписать в следующей дискретной форме и рассматривать их как огра-

ничения в форме равенств в задаче нелинейного программирования:

$$\left. \begin{aligned} x_{i+1} - x_i &= -(\omega + r_0 u_i) x_i T_{i+1} + g_1 y_i T_{i+1} + g_2 u_i T_{i+1}, \\ y_{i+1} - y_i &= -y_i T_{i+1} + u_i T_{i+1}, \quad i=0, 1, \dots, N-1. \end{aligned} \right\} \quad (6.9)$$

Показатель качества будет теперь следующий:

$$\text{минимизировать } J = \sum_{i=1}^N T_i \quad (6.10)$$

при условии выполнения равенства (6.9) и следующих ограничений в форме неравенств:

$$x_i \leq x_{\max}, \quad i=1, \dots, N, \quad (6.11)$$

$$u_i \leq u_{\max}, \quad i=0, \dots, N-1. \quad (6.12)$$

Далее следуют решения двух численных примеров.

Пример 1 [3]. Исходные данные: $x(0) = y(0) = 1$, $u_{\max} = 2$, $x_{\max} = 4,8$, $\gamma_x = 0,003$, $\gamma_y = 0,056$, $\lambda_1 = 2,9 \times 10^{-5} \text{ сек}^{-1}$ (I), $\lambda_2 = 2,1 \times 10^{-5} \text{ сек}^{-1}$ (Xe), $\sigma = 3,5 \times 10^{-18} \text{ см}^2$, $\varphi_0 = 1,66 \times 10^{14} \text{ н/см}^2 \cdot \text{сек}$. Вычисленные значения: $r_0 = 20$, $\omega = 0,724$, $\gamma_1 = 0,949$, $\gamma_2 = 0,0508$, $g_1 = 19,67$, $g_2 = 1,054$, $g_w = 3,625$, $F = 0,014$, $G = 8,4$.

Решение задачи нелинейного программирования при $N=12$ дало следующие результаты:

	T_i (мин)	t_i (мин)	$u_{i-1} \times 10^{-3}$	x_i (Xe)	y_i (I)
1	0,032	0,032	4,67	1,001	1,000
2	0,032	0,064	4,67	1,002	1,000
3	0,032	0,096	4,67	1,003	1,000
4	0,032	0,128	4,67	1,004	1,000
5	0,032	0,160	4,67	1,005	1,000
6	0,032	0,192	4,67	1,006	1,000
7	0,032	0,224	4,67	1,007	1,000
8	115,035	115,259	0,07	4,796	0,799
9	0,032	115,291	2,14	4,737	0,799
10	0,032	115,323	2,14	4,797	0,799
11	0,032	115,355	2,14	4,798	0,799
12	533,921	659,276	129,05	4,800	0,177

Таким образом, минимальное время равно

$$T_{\min} = 649,276 \text{ мин} = 10,8 \text{ час.}$$

Значения T_i , фактически полученные машиной, были равны значениям T_i в таблице, деленным на множитель масштабирования, равный 578,8 мин.

Пример 2 [4]. Исходные данные: $x(0) = y(0) = 1$, $u_{\max} = 1$, $x_{\max} = 5,0$, $\gamma_x = 0,002$, $\gamma_y = 0,061$, $\lambda_1 = 2,9 \times 10^{-5} \text{ сек}^{-1}$, $\lambda_2 = 2,1 \times 10^{-5} \text{ сек}^{-1}$, $\sigma = 2,7 \times 10^{-18} \text{ см}^2$, $\varphi_0 = 10^{14} \text{ н/см}^2 \cdot \text{сек}$.

Вычисленные значения (отличающиеся от полученных в предыдущем примере): $r_0 = 9,47$, $\gamma_1 = 0,968$, $\gamma_2 = 0,0317$, $g_1 = 9,87$, $g_2 = 0,324$, $F = 0,02795$, $G = 4,27$.

Следующие результаты были получены при $N = 12$:

i	T_i (мин)	t_i (мин)	$u_{i-1} \times 10^{-3}$	x_i (Xe)	y_i (I)
1	117,000	117,000	0,47	2,86	0,796
2	0,272	117,272	12,00	2,86	0,796
3	0,270	117,542	12,00	2,86	0,795
4	0,266	117,808	12,00	2,86	0,795
5	0,271	118,079	12,00	2,87	0,794
6	0,270	118,349	12,00	2,87	0,794
7	0,270	118,619	12,00	2,88	0,793
8	0,272	118,891	12,00	2,88	0,793
9	0,273	119,164	12,00	2,88	0,792
10	0,275	119,439	12,00	2,88	0,792
11	0,274	119,713	12,00	2,89	0,701
12	343,000	462,713	80,78	5,00	0,367

Таким образом, минимальное время равно

$$T_{\min} = 462,713 \text{ мин} = 7,7 \text{ час.}$$

Чтобы оценить возможную ошибку дискретизации, были проведены вычисления для того же примера, но при $N = 20$. Конечные результаты оказались приблизительно те же. Минимальное время вычисления оказалось равным

$$T_{\min} = 463,4 \text{ мин.}$$

Учитывая малость разности и малость относительной разности

$$\frac{T_{\min}^{20} - T_{\min}^{12}}{T_{\min}^{20}} = \frac{0,7}{463} = 0,0015 = 0,15 \%,$$

можно утверждать, что ошибка дискретизации является допустимой для практики.

Вычисления для этих примеров были произведены на ЦВМ ИБМ 7094 с использованием метода последовательной минимизации без ограничений. Вновь обращаясь к дискретизированным уравнениям состояния (6.9), видим, что была использована простейшая форма дискретизации. Она дает аппроксимацию только первого порядка, поскольку используются первые два члена разложения Тейлора, до производной первого порядка. Чтобы получить большую точность при дискретизации нелинейных уравнений состояния с непрерывным временем, надо применить аппроксимацию более высокого порядка. Пример, иллюстрирующий это, будет приведен в 6.3.

6.3. Аппроксимация более высокого порядка; задача оптимизации траектории

Пирсон [8, 9] предложил применить разностную схему третьего порядка. Метод дискретизации основан на интерполяционных разностных уравнениях Адамса [10]. Исходя из системы уравнений состояния, например вида (4.1), делим интервал $t_0 \leq t \leq t_f$ на N равных частей. Тогда разностные уравнения можно записать следующим образом:

$$\left. \begin{aligned} y(1) - y(0) - 2b[f(0) + f(1)] &= 0, \\ y(2) - y(1) - 2b[-f(0) + 8f(1) + 5f(2)] &= 0, \\ y(3) - y(2) - b[f(0) - 5f(1) + 19f(2) + 9f(3)] &= 0, \\ y(N) - y(N-1) - b[\dot{f}(N-3) - 5\dot{f}(N-2) + \\ &+ 19\dot{f}(N-1) + 9\dot{f}(N)] = 0, \end{aligned} \right\} (6.13)$$

где

$$f(i) = f[y(i), u(i), i], \quad (6.14)$$

$$b = \frac{t_f}{24N}. \quad (6.15)$$

Опыт прежних вычислений [8, 9] показал, что такая разностная схема оказывается вполне удовлетворительной.

Применяемый в данном случае алгоритм вычислений аналогичен алгоритму последовательной минимизации без ограничений (см. гл. 3) в том смысле, что предусматривает решение последовательности задач без ограничений. Однако в данном алгоритме применен другой метод введения штрафа. Этот метод, использующий единичную функцию Хэвисайда, был предложен Курантом [11, 12]. В алгоритме используется следующий модифицированный показатель качества:

$$J^* = J + K_v \left[\sum_i h_i^2 + \sum_i H(-g_i) (\omega g_i)^4 \right], \quad (6.16)$$

где J — исходный показатель качества, h_i — ограничения в форме равенств $h_i = 0$, g_i — ограничения в форме неравенств $g_i \geq 0$, H — единичная ступенчатая функция Хэвисайда:

$$H(z) = \begin{cases} 1, & z > 0, \\ 0, & z \leq 0, \end{cases} \quad (6.17)$$

$K_v = C^{h-1}K_1$; $C > 1$; $K_1 > 0$, $k = 1, 2, \dots$ (K_v играет роль r_k в алгоритме последовательной минимизации без ограничений, описанном в гл. 3), ω — произвольная весовая постоянная.

Метод минимизации без ограничений, примененный в нашем случае, принадлежит к классу методов сопряженных градиентов (см. 3.4). Точнее говоря, был применен алгоритм Флетчера и Ривза [13]. При применении этого метода новый направляющий вектор \mathbf{r}_i выбирается согласно следующей формуле:

$$\mathbf{r}_i = -\nabla f_{i+1} + \mathbf{r}_{i+1} \frac{(\nabla f_i^T \nabla f_i)}{(\nabla f_{i-1}^T \nabla f_{i-1})}. \quad (6.18)$$

Итерации возобновляются через каждые $M+1$ шагов, где M — общее число переменных задачи минимизации. Это положение основывается главным образом на опыте вычислений, связанных с применением метода [8]. Для определения длины шага d_i на каждой итерации решается одномерная задача минимизации, для чего применялось три различных метода поиска: модифицированный поиск Фибоначчи [14], модифицированный метод Ньютона [15] и кубическая интерполяция, предложенная Давидоном [16]. Последний метод оказался значительно более эффективным, чем любой из первых двух [8].

Пример 3. Упрощенная задача о ракете [8]. Рассматриваемая задача может быть сформулирована следующим образом.

Полагая, что гравитационное ускорение g_0 постоянно, что полет происходит в вакууме и что тяговое ускорение A ракеты постоянно, найти зависимость направления тяги от времени, необходимую, чтобы ракета достигла заданной высоты Y_f в заданное время t_f при нулевой вертикальной скорости V_y и максимальной горизонтальной скорости V_x . Дальность до конечной точки не задается.

Показатель качества этой задачи:

$$\left. \begin{array}{l} \text{максимизировать } V_x(t_f) \\ \text{или} \\ \text{минимизировать } J = -V_x(t_f). \end{array} \right\} \quad (6.19)$$

Вектор состояний в этой задаче равен

$$\mathbf{y} = \begin{bmatrix} X \\ Y \\ V_x \\ V_y \end{bmatrix} \quad \begin{array}{l} \text{горизонтальная дальность} \\ \text{высота} \\ \text{горизонтальная составляющая} \\ \text{скорости} \\ \text{вертикальная составляющая} \\ \text{скорости} \end{array}$$

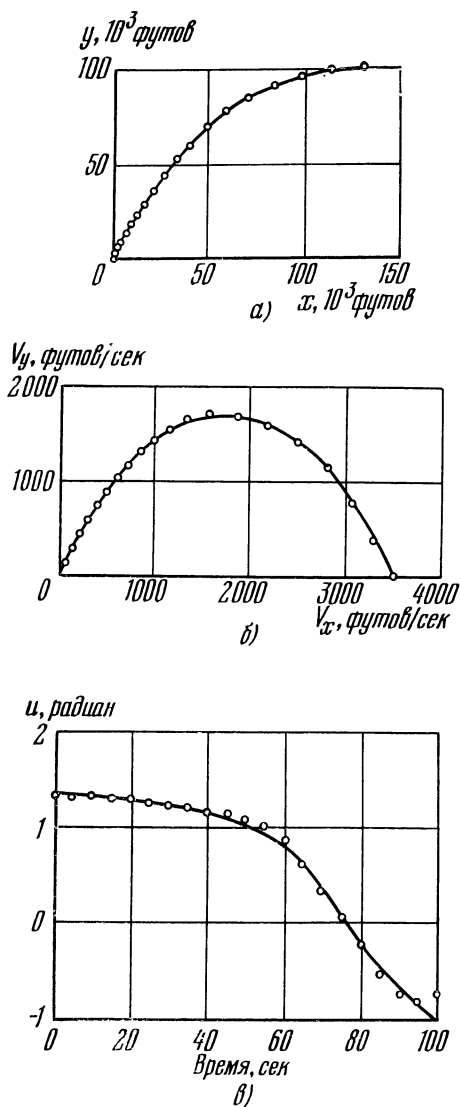


Рис. 6.1. Результаты решения задачи о ракете [8]: а) оптимальная траектория, б) годограф оптимальной траектории, в) зависимость оптимального управления от времени.

Уравнения состояния данной системы:

$$\left. \begin{aligned} \dot{y}_1 &= y_3, \\ \dot{y}_2 &= y_4, \\ \dot{y}_3 &= A \cos u, \\ \dot{y}_4 &= A \sin u - g_0. \end{aligned} \right\} \quad (6.20)$$

Управляющей переменной u является наклон вектора силы тяги по отношению к горизонтальной плоскости.

Начальные условия:

$$y_i(0) = 0, \quad i = 1, \dots, 4. \quad (6.21)$$

Конечные условия:

$$\left. \begin{aligned} y_2(t_f) &= Y_F = 10^5 \text{ фут}, \\ y_4(t_f) &= 0, \\ t_f &= 100 \text{ сек.} \end{aligned} \right\} \quad (6.22)$$

Другие данные:

$$g_0 = 32 \text{ фут/сек}^2, \quad A/g_0 = 2.$$

Задача была решена на ЦВМ ИБМ 7090 и дала следующие результаты:

N	K_v	Затраченное машинное время (сек)	Число итераций	Конечное значение V_x (фут/сек)
5	1,000		123	3688,5
5	4,000		19	3632,0
5	16,000		20	3113,4
5	64,000	10	18	3608,9
10	64,000		398	3517,6
10	256,000	56	37	3514,7
20	256,000	77	95	3500,0
40	256,000	342	654	3502,6

Как видно из таблицы, увеличение числа интервалов до $N=20$ не влияет существенно на конечное значение V_x . Полное решение для $N=20$ показано на рис. 6.1.

6.4. Вычисление с помощью ЦВМ оптимальных параметров нелинейных систем управления

Часто бывает, что система управления предназначается для работы в условиях, при которых качественные требования не остаются постоянными. Иногда оказывается более важным иметь возможно большее усиление,

в то время как на других этапах управляемого процесса более важным оказывается требование минимального времени установления заданного режима. Во всех случаях необходимо сохранять устойчивость системы. Обеспечить работу системы при меняющемся оптимальном показателе качества можно путем соответствующих изменений устанавливаемых параметров системы. Один из лучших способов управления системой в этих условиях заключается в применении ЦВМ для непосредственного или косвенного управления.

Для этой цели нужен алгоритм, который, объединяя требования динамики системы, критерия устойчивости и показателя качества, вычисляет оптимальные параметры системы. Когда изменяется показатель качества, информация об этом соответственно вводится в машину и вновь вычисляются оптимальные параметры. Результаты вычисления затем могут быть прямо или косвенно переданы в систему.

Был разработан алгоритм, выполняющий эту задачу для нелинейных систем управления [17]. Этот алгоритм использует критерий устойчивости Попова для нелинейных систем [18] и метод нелинейного программирования.

Работа алгоритма изучалась на модели с помощью машинной системы ИБМ 360/91.

Рассматриваемая система управления схематически представлена на рис. 6.2, *а*. Она состоит из линейной системы n -го порядка с передаточной функцией $G(s, \mathbf{p})$. Здесь \mathbf{p} — вектор параметров, компоненты которого являются устанавливаемыми параметрами системы. $G(s, \mathbf{p})$ в общем случае может объединять основной объект и регулятор. Предполагается, что система содержит единственную нелинейность, которая не входит в линейную часть системы. Общий вид нелинейной характеристики показан на рис. 6.2, *б*. На входе нелинейности действует сигнал ошибки e , а ее выходной сигнал $u = n(e)$ подается на вход линейной части системы. Относительно нелинейности дополнительно предполагается [18], что

1) $n(e)$ — известная непрерывная функция при всех значениях e ,

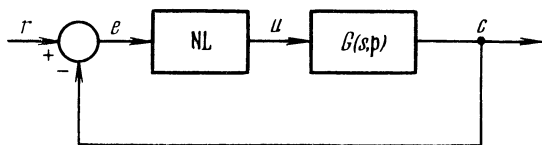
2) $n(0) = 0$, $en(e) > 0$ при всех $e \neq 0$.

Хотя это и ограничивает характер нелинейности, тем не менее широкий класс существующих на практике нелинейностей попадает в эту категорию или может быть таким способом приближенно представлен.

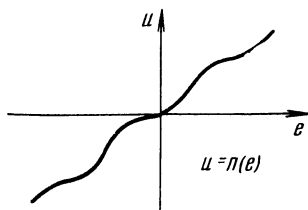
Частотную характеристику линейной части системы представим в виде

$$G(j\omega, p) = G_1(\omega, p) + jG_2(\omega, p). \quad (6.23)$$

Предполагается, что все полюсы и нули линейной части, т. е. функции $G(s, p)$, лежат в левой полуплоскости.



а)



б)

Рис. 6.2. Система управления [17]: а) схематическое изображение системы, б) нелинейность.

Теорема Попова [18] утверждает, что рассматриваемая система будет абсолютно устойчива, если при

$$0 \leq \frac{n(e)}{e} \leq K \quad \left(\text{или } 0 \leq \frac{u}{e} \leq K \right)$$

существует неотрицательное вещественное q такое, что

$$G_1(\omega, p) - q\omega G_2(\omega, p) - \frac{1}{K} > 0, \quad (6.24)$$

K фактически представляет собой максимально допустимое мгновенное усиление (отношение выходного сигнала

к входному) нелинейности. Как будет видно из нижеприводимых примеров, из неравенства (6.24) можно получить систему ограничений в форме неравенств, являющихся функциями \mathbf{p} , K и q :

$$g_j(\mathbf{p}, K, q) \geq 0, \quad q \geq 0, \quad (6.25)$$

где $j=1, \dots, M$. Общее число M ограничений в (6.25), полученных из условия устойчивости (6.24), зависит от порядка и структуры линейной части системы.

Для различных моментов времени можно приписать системе некоторые показатели качества, которые можно представить в следующем общем виде:

$$J = f(\mathbf{p}, K). \quad (6.26)$$

Тогда задача оптимизации заключается в минимизации (или максимизации) J при выполнении ограничений (6.25), а также при любых других ограничениях, которые могут быть наложены на систему по различным практическим соображениям:

$$g_i(\mathbf{p}, K) \geq 0, \quad i=1, \dots, I_g, \quad (6.27)$$

и

$$h_i(\mathbf{p}, K) = 0, \quad i=1, \dots, I_h. \quad (6.28)$$

Эта формулировка и определяет задачу нелинейного программирования.

Пример 4. Система третьего порядка [17]. Передаточную функцию для системы третьего порядка представим в виде

$$G(s, \mathbf{p}) = \frac{1}{(s+a)(s^2 + 2\delta\omega_n s + \omega_n^2)}, \quad a, \delta, \omega_n \geq 0. \quad (6.29)$$

В этом случае устанавливаемыми параметрами являются a , δ и ω_n и, следовательно, вектор параметров равен

$$\mathbf{p} = \begin{bmatrix} a \\ \delta \\ \omega_n \end{bmatrix}.$$

Для частотной области запишем

$$G(j\omega, \mathbf{p}) = G_1(\omega, \mathbf{p}) + jG_2(\omega, \mathbf{p}).$$

Подставляя $s=j\omega$ в (6.29), получим

$$G_1(\omega, \mathbf{p}) = \frac{a(\omega_n^2 - \omega^2) - 2\delta_n \omega^2}{(a^2 + \omega^2) [(\omega_n^2 - \omega^2)^2 + 4\delta^2 \omega_n^2 \omega^2]}, \quad (6.30)$$

$$G_2(\omega, \mathbf{p}) = \frac{\omega(\omega_n^2 - \omega^2) + 2\delta a \omega_n \omega}{(a^2 + \omega^2) [(\omega_n^2 - \omega^2)^2 + 4\delta^2 \omega_n^2 \omega^2]}. \quad (6.31)$$

Подставляя (6.30) и (6.31) в (6.24) и вводя

$$x = \frac{1}{K},$$

получим

$$x\omega^6 + [x(a^2 - 2\omega_n^2 + 4\delta^2\omega_n^2) - q]\omega^4 + [x(4\delta^2 a^2 \omega_n^2 - 2a^2 \omega_n^2 + \omega_n^4) + 2q\delta a \omega_n + q\omega_n^2 - 2\delta\omega_n - a]\omega^2 + (xa^2\omega_n^4 + a\omega_n^2) > 0. \quad (6.32)$$

Так как x и q неотрицательны и так как в (6.32) ω входит лишь в четной степени, то неравенство (6.32) будет выполняться при всех ω , если коэффициенты при ω^4 и ω^2 неотрицательны, т. е.

$$x(a^2 - 2\omega_n^2 + 4\delta^2\omega_n^2) - q \geq 0, \quad (6.33)$$

$$x(4\delta^2 a^2 \omega_n^2 - 2a^2 \omega_n^2 + \omega_n^4) + 2q\delta a \omega_n - 2\delta\omega_n - a \geq 0. \quad (6.34)$$

Ограничения (6.33) и (6.34) представляют собой условие устойчивости для системы третьего порядка, выраженное в общей форме в (6.25). Кроме ограничений (6.33) и (6.34) в нашем примере наложены следующие ограничения:

$$\delta_{\min} \leq \delta \leq \delta_{\max}, \quad a_{\min} \leq a \leq a_{\max}, \quad \omega_n < \omega_{n \max}, \quad q \leq q_{\max}. \quad (6.35)$$

Ограничения (6.35) линейны и очень просты. Однако ограничения (6.33) и (6.34) — нелинейны, причем (6.33) — неравенство пятой степени, а (6.34) — седьмой.

Показатель качества для нашего примера был выбран из следующих соображений. Во многих случаях оказывается желательным иметь параметры, обеспечивающие работоспособность при максимальном прямом усилении (в нашем случае равном K), сохраняя устойчивость системы. С другой стороны, может быть желательным иметь возможно малое время установления процесса, для чего надо иметь малый коэффициент затухания δ . Требование максимизации K (или минимизации $1/K$) вызывает необходимость увеличения δ . Таким образом, в задаче оптимизации действуют два противоречивых требования. В показатель качества приходится ввести веса для учета относительной важности этих двух требований. Так как эти две величины, x и δ , неотрицательны, то показатель качества может быть построен не в виде квадратичной, а в виде линейной формы:

$$\text{минимизировать } J = cx + \delta. \quad (6.36)$$

Дополнительно к условиям устойчивости (6.33) и (6.34) наложены следующие ограничения:

$$\left. \begin{aligned} 0,5 \leq \delta \leq 0,707, \\ 0,1 \leq a \leq 1, \\ \omega \leq 3 \text{ рад/сек}, \\ q \leq 10, \end{aligned} \right\} \quad (6.37)$$

Соотношение между переменными принято следующее:

Переменные задачи управления	Переменные задачи математического программирования	Переменные задачи управления	Переменные задачи математического программирования
x	$x(1)$	ω_n	$x(4)$
a	$x(2)$	q	$x(5)$
δ	$x(3)$		

Подставляя в (6.33), (6.34), (6.36) и (6.37) обозначения, принятые для задачи математического программирования, приходим к следующей задаче нелинейного программирования:

$$\begin{aligned} \min \{ & cx(1) + x(3) \mid x(3) - \delta_{\min} \geq 0; x(2) - a_{\min} \geq 0; a_{\max} - x(2) \geq 0; \\ & \omega_n \max - x(4) \geq 0; x(1) \{x^2(2) - [2 - 4x^2(3)]x^2(4)\} - x(5) \geq 0; \\ & x(1) \{[4x^2(3) - 2]x^2(2)x^2(4) + x^4(4)\} + 2x(2)x(3)x(4)x(5) + x(5)x^2(4) - \\ & - 2x(3)x(4) - x(2) \geq 0; \delta_{\max} - x(3) \geq 0; q_{\max} - x(5) \geq 0 \}. \end{aligned}$$

Задача нелинейного программирования содержит пять переменных и восемь ограничений в виде неравенств. Так как два ограничения нелинейны, то для решения нашей задачи целесообразно применить программу для метода последовательной минимизации без ограничений (гл. 3).

Задача была решена для различных значений весового коэффициента c .

Вычисления по программе метода последовательной минимизации без ограничений производились на вычислительной системе ИБМ 360/91. Результаты вычислений приведены в следующей таблице:

c	0,01	0,02	0,05	0,10	0,15	0,18	0,20	0,50	1,00
x_{\min}	4,00	4,00	0,910	0,312	0,128	0,081	0,066	0,064	0,064
K_{\max}	0,25	0,25	1,100	3,200	7,810	12,35	15,15	15,61	15,61
δ	0,5	0,5	0,612	0,654	0,676	0,684	0,687	0,687	0,687
ω_n	0,5	0,5	1,414	1,861	2,420	2,783	2,972	2,999	3,000
q	3,0	3,0	$1,46 \times 10^{-4}$	$4,92 \times 10^{-5}$	$1,82 \times 10^{-5}$	$1,50 \times 10^{-5}$	$1,95 \times 10^{-5}$	$6,40 \times 10^{-5}$	$2,49 \times 10^{-5}$
$a=1$ во всех случаях									

Для всех значений c машинное время, необходимое для решения задачи нелинейного программирования, составляло около 1,30 сек.

Теперь изложим более подробно вычисления по методу последовательной минимизации без ограничений для одного из примеров, а именно для случая $c=0,01$.

Результаты вычислений приводятся в следующей таблице (см. 3.3):

i	r_i	$f(x)$	$x=x(1)$	$a=x(2)$	$\delta=x(3)$	$\omega_n=x(4)$	$q=x(5)$
	Начальная точка	0,601	0,100	0,200	0,600	0,100	0,100
0	1,0000	3,736	30,406	0,859	0,695	2,273	4,901
1	0,2500	1,483	78,791	0,865	0,695	2,256	4,596
2	0,0625	0,913	21,817	0,881	0,695	2,186	3,402
3	0,0156	0,607	8,197	0,939	0,525	0,537	4,223
4	0,0039	0,556	5,057	0,975	0,505	0,497	3,375
5	$9,8 \times 10^{-4}$	0,544	4,262	0,992	0,501	0,498	3,096
6	$2,4 \times 10^{-4}$	0,541	4,066	0,998	0,500	0,499	3,024
7	$6,1 \times 10^{-5}$	0,540	4,016	0,999	0,500	0,500	3,006
8	$1,5 \times 10^{-5}$	0,540	4,004	1,000	0,500	0,500	3,001
9	$3,3 \times 10^{-6}$	0,540	4,002	1,000	0,500	0,500	3,001

Как видно из таблицы, алгоритм сходится после девяти итераций. Значение целевой функции, близкое к оптимальному, получается на пятой итерации. В качестве метода минимизации без ограничений использовался обобщенный метод Ньютона (см. 3.4).

Пример 5. Система четвертого порядка [17]. Передаточную функцию системы четвертого порядка запишем в следующем виде:

$$G(s, p) = \frac{1}{(s^2 + 2\delta_1\omega_{n1}s + \omega_{n1}^2)(s^2 + 2\delta_2\omega_{n2}s + \omega_{n2}^2)}, \quad (6.38)$$

$$G_1(\omega, p) = \frac{1}{D} [(\omega_{n1}^2 - \omega^2)(\omega_{n2}^2 - \omega^2) - 4\delta_1\delta_2\omega_{n1}\omega_{n2}\omega^2], \quad (6.39)$$

$$G_2(\omega, p) = -\frac{1}{D} [(2\delta_2\omega_{n2}\omega(\omega_{n1}^2 - \omega^2) + 2\delta_1\omega_{n1}\omega(\omega_{n2}^2 - \omega^2))], \quad (6.40)$$

где

$$D = (\omega_{n1}^2 - \omega^2)^2 (\omega_{n2}^2 - \omega^2)^2 + 4\delta^2\omega_{n2}^2\omega^2 (\omega_{n1}^2 - \omega^2)^2 + \\ + 4\delta_1^2\omega_{n1}^2\omega^2 (\omega_{n2}^2 - \omega^2)^2 + 16\delta_1^2\delta_2^2\omega_{n1}^2\omega_{n2}^2\omega^4.$$

Вектор устанавливаемых параметров в данном случае равен

$$p = \begin{bmatrix} \delta_1 \\ \omega_{n1} \\ \delta_2 \\ \omega_{n2} \end{bmatrix}.$$

Подставляя (6.39) и (6.40) в (6.38), так же как и в случае системы третьего порядка, получим условия устойчивости в виде ограничивающих неравенств

$$(2\delta_1^2 - 1) \omega_{n1}^2 + (2\delta_2^2 - 1) \omega_{n2}^2 \geq 0, \quad (6.41)$$

$$x \left[4\omega_{n1}^2 \omega_{n2}^2 (4\delta_1^2 \delta_2^2 - 2\delta_1^2 - 2\delta_2^2 + 1) + \omega_{n1}^4 + \omega_{n2}^4 \right] + 1 - 2q (\delta_1 \omega_{n1} + \delta_2 \omega_{n2}) \geq 0, \quad (6.42)$$

$$2x\omega_{n1}^2 \omega_{n2}^2 \left[(2\delta_1^2 - 1) \omega_{n2}^2 + (2\delta_2^2 - 1) \omega_{n1}^2 \right] + 2\omega_{n1} \omega_{n2} q (\delta_1 \omega_{n2} + \delta_2 \omega_{n1}) - \omega_{n1}^2 - \omega_{n2}^2 - 4\delta_1 \delta_2 \omega_{n1} \omega_{n2} \geq 0. \quad (6.43)$$

Показатель качества:

$$\text{минимизировать } J = cx + \delta_1 + \delta_2. \quad (6.44)$$

Дополнительные ограничения: $0,5 \leq \delta_1 \leq 2$, $0,5 \leq \delta_2 \leq 2$, $\omega_{n1} \leq 3$ рад/сек, $\omega_{n2} \leq 3$ рад/сек.

Получены следующие результаты:

c	0,01	0,05	0,10	0,20	1,0
x_{\min}	0,251	0,163	0,138	0,121	0,100
K_{\max}	3,98	6,14	7,25	8,26	10,0
δ_1	0,708	0,710	0,712	0,714	0,723
ω_{n1}	2,999	2,999	3,0	3,0	3,0
δ_2	0,5	0,5	0,5	0,5	0,5
ω_{n2}	0,236	0,398	0,496	0,611	0,916
q	4,748	3,043	2,562	2,204	1,728

Машинное время, необходимое для решения задачи нелинейного программирования, для каждого значения c составляло около 1,45 сек. Каждая задача имеет шесть переменных (x , δ_1 , ω_{n1} , δ_2 , ω_{n2} , q) и девять ограничений в форме неравенств.

Описанный выше алгоритм, объединяющий критерий устойчивости Попова для нелинейных систем и метод оптимизации путем нелинейного программирования, может эффективно применяться к нелинейным системам третьего и четвертого порядков при изменяющихся требованиях, предъявляемых к системе. Легко видеть, что тот же алгоритм можно легко распространить на системы более высокого порядка. Показатель качества при этом может быть любым; никаких ограничений здесь нет.

При наличии оборудования для работы в режиме разделения времени управление реальной системой может осуществляться следу-

ющим образом. Программа последовательной минимизации без ограничений, как и дополнительные подпрограммы, моделирующие ограничения, наложенные на систему, могут быть записаны на магнитной ленте. С помощью вынесенного пульта-оператора можно в любое

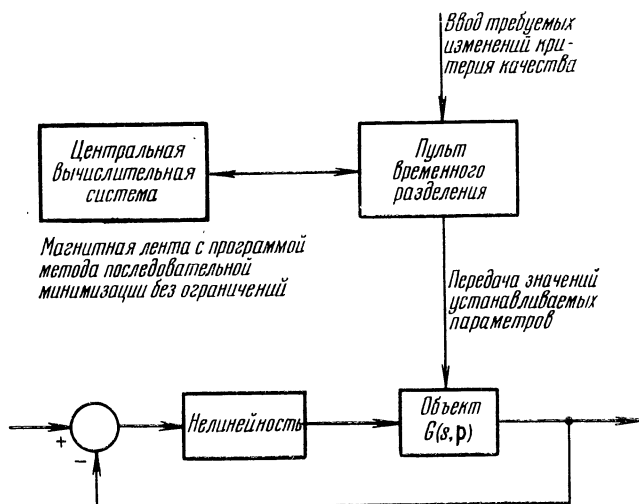


Рис. 6.3. Схема управления с помощью ЦВМ [17].

время передать входные данные, в том числе и изменения показателя качества. Фактическая работа программы занимает лишь несколько секунд.

Вычисленные значения устанавливаемых параметров, в случае непрерывного цифрового управления могут быть затем переданы из пульта в систему, или установлены оператором (рис. 6.3).

6.5. Метод внутреннего штрафа

Метод внутреннего штрафа для задачи оптимального управления с ограничениями предложили Лэсдон, Уорен, и Райс [19]. Главная идея этого метода основывается на алгоритме метода последовательной минимизации без ограничений Фиакко и Мак-Кормика (см. гл. 3).

Лэсдон и др. рассмотрели следующую задачу.

Дана система, описываемая следующими уравнениями состояния:

$$\dot{y} = f(y, u, t). \quad (6.45)$$

Это уравнение не отличается от (4.1). Предполагается, что вектор начального состояния $\mathbf{y}(t_0) = \mathbf{y}_0$ известен. Конечная область определяется системой ограничений в форме равенств

$$\mathbf{h}(\mathbf{y}_f, t_f) = \mathbf{0}, \quad (6.46)$$

где \mathbf{h} является r -мерной векторной функцией. При всех значениях времени система должна удовлетворять некоторым ограничениям типа неравенств

$$\mathbf{g}(\mathbf{y}, \mathbf{u}, t) \geq \mathbf{0}, \quad (6.47)$$

где \mathbf{g} — s -мерная векторная функция.

Рассматриваемый показатель качества относится к задачам терминального управления, так как он явно зависит лишь от конечного состояния и момента окончания:

$$\text{минимизировать } J = H(\mathbf{y}_f, t_f). \quad (6.48)$$

Чтобы решить задачу, формулируется следующий модифицированный показатель качества:

минимизировать $P(\mathbf{u}, r) =$

$$= H(\mathbf{y}_f, t_f) + r \sum_{i=1}^s \int_{t_0}^{t_f} \frac{dt}{g_i(\mathbf{y}, \mathbf{u}, t)}. \quad (6.49)$$

Это выражение аналогично выражению (3.129).

Как и в алгоритме метода последовательной минимизации без ограничений, здесь надо найти последовательность \mathbf{u}_i , минимизирующую $P(\mathbf{u}, r_i)$ при убывающих значениях коэффициентов r_i . На каждом шаге процесса для минимизации без ограничений применялся метод сопряженных градиентов [20]. Аналогичный способ решения задач оптимального управления с помощью метода сопряженных градиентов разработали Синнот и Люенбергер [21] (см. также [27]).

6.6. Решение двухточечных граничных задач

Известно, что метод принципа максимума (см. [12] в гл. 4) требует решения двухточечной граничной задачи (см. [1—3] в гл. 4). Эта задача особенно трудна, когда принцип максимума применяется к нелинейным систе-

мам оптимального управления. Розен и Майер [22, 23] предложили применение линейного программирования непосредственно к решению двухточечной граничной задачи. Типичная двухточечная граничная задача, получающаяся в результате применения принципа максимума к решению задач оптимального управления, может быть сформулирована следующим образом.

Дана система из n обыкновенных нелинейных дифференциальных уравнений первого порядка

$$\dot{y}_i(t) = f_i[y(t), t], \quad i=1, \dots, n, \quad (6.50)$$

во временном интервале $t_0 \leq t \leq t_f$, где $y_i(t)$ — искомые функции, а f_i — известные нелинейные непрерывные функции $y(t)$ и t ,

$$y(t) = [y_1(t), \dots, y_n(t)]^T.$$

Граничные условия заданы следующим образом:

$$y_i(t_0) = b_{i0}, \quad i=1, \dots, m; \quad m < n, \quad (6.51)$$

$$y_i(t_f) = b_{if}, \quad i=m+1, \dots, n. \quad (6.52)$$

Уравнение (6.50) можно дискретизировать, как это было описано в гл. 4, или в предыдущих параграфах настоящей главы.

Схема дискретизации, использованная Розеном и Майером [22, 23] приводится ниже применительно к уравнению (6.50):

$$y(j+1) - y(j) = \frac{1}{2} T \{f[y(j+1), j+1] + f[y(j), j]\}, \\ j = 0, 1, \dots, N-1, \quad (6.53)$$

где $T = \frac{t_f - t_0}{N}$, N — число дискретных временных подынтервалов на интервале $t_0 \leq t \leq t_f$ (подынтервалы предполагаются равными), $y(j) = y(t_j)$, $t_j = t_0 + jT$ и $f[y(j), j] = f[y(t_j), t_j]$.

Теперь мы имеем систему из nN нелинейных уравнений со следующими неизвестными переменными:

$y_i(1), y_i(2), \dots, y_i(N), \quad i=1, \dots, m; \quad mN$ переменных
 $y_i(0), y_i(1), \dots, y_i(N-1), \quad i=m+1, \dots, n; \quad (n-m)N$
 переменных

всего nN переменных

С помощью соответствующей формулировки показателя качества эта задача может быть представлена и решена как задача нелинейного программирования. Например, гамильтониан можно представить как функцию y_i из (6.50) и вновь представить в дискретной форме и максимизировать для каждого рассматриваемого момента времени. Розен и Майер [22, 23], используя линеаризацию уравнений (6.53), еще упростили задачу и получили возможность применить линейное программирование.

Литература

1. D. T a b a k, A Direct and Nonlinear Programming Approach to the Optimal Nuclear Reactor Shutdown Control, IEEE Trans. Nucl. Sci. NS-15, pp. 57—59 (1968).
2. D. T a b a k, Application of Mathematical Programming in the Design of Optimal Control Systems, Ph. D. Thesis, University of Illinois, Urbana, 1967.
3. M. A s h, Optimal Shutdown Control of Nuclear Reactor, Academic Press, New York, 1966.
4. J. J. R o b e r t s and H. P. S m i t h, Time Optimal Solution to the Reactivity-Xenon Shutdown Problem, Nucl. Sci. Eng. 22, pp. 470—478 (1965).
5. J. J. R o b e r t s and H. P. S m i t h, Equivalence of the Time Optimal and Minimax Solutions to the Xenon Shutdown Problem, Nucl. Sci. Eng. 23, pp. 397—399 (1965).
6. Z. R. R o s z t o c z y and L. E. W e a v e r, Optimum Reactor Shutdown Program for Minimum Xenon Buildup, Nucl. Sci. Eng. 20, pp. 318—323 (1964).
7. R. B e l l m a n, Dynamic Programming, Princeton University Press, Princeton, N. J., 1957. [Русский перевод: Р. Б е л л м а н, Динамическое программирование, ИЛ, 1960.]
8. B. L. P i e r s o n, A Discrete Variable Approximation to Optimal Flight Paths, Astronautica Acta 14, pp. 157—169 (1969).
9. B. L. P i e r s o n, Numerical Solution of Nonlinear Optimal Control Problems by Discrete Variable Approximation. SIAM 1968 Fall Meeting, Philadelphia, Pa., Oct. 1968.
10. L. C o l l a t z, The Numerical Treatment of Differential Equations, 3rd ed., Springer, Berlin, 1960, p. 85. [Русский перевод: Л. К о л л а т ц, Численные методы решения дифференциальных уравнений, ИЛ, 1953.]
11. R. C o u r a n t, Variational Methods for the Solution of Problems of Equilibrium and Vibrations, Bull. Am. Math. Soc. 49, pp. 1—23 (1943).
12. J. K o w a l i k and M. R. O s b o r n e, Methods for Unconstrained Optimization Problems, American Elsevier, New York, 1968.
13. R. F l e t c h e r and C. M. R e e v e s, Function Minimization by Conjugate Gradients, Computer J. 7, pp. 149—154 (1964).

14. D. J. Wilde and C. S. Beightler, *Foundations of Optimization* Prentice-Hall, Englewood Cliffs, N. J., 1967.
15. D. J. Wilde, *Optimum Seeking Methods*, Prentice-Hall, Englewood-Cliffs, N. J., 1964. [Русский перевод: Д. Дж. Уайлд, Методы поиска экстремума, «Наука», 1967.]
16. W. C. Davidson, *Variable Metric Method for Minimization*, Report ANL-5990, Argonne National Laboratory, Argonne, Ill., 1959.
17. D. Tabaak, *Computer Control of Nonlinear Systems with Varying Performance Criteria*, Intern. J. Control **11**, pp. 941—947 (1970).
18. В. М. Попов, Абсолютная устойчивость нелинейных систем автоматического регулирования, *Автоматика и телемеханика* **22**, № 8 (1961).
19. L. S. Lasdon, A. D. Waren and R. K. Rice, *An Interior Penalty Method for Inequality Constrained Optimal Control Problems*, IEEE Trans. Automatic Control **AC-12**, pp. 388—395 (1967).
20. L. S. Lasdon, S. K. Mitter and A. D. Waren, *The Conjugate Gradient Method for Optimal Control Problems*, IEEE Trans. Automatic Control **AC-12**, pp. 132—138 (1967).
21. J. F. Sinnott, Jr. and D. G. Luenberger, *Solution of Optimal Control Problems by the Method of Conjugate Gradients*, 1967, IACC, pp. 566—574, Philadelphia, Pa.
22. J. B. Rosen and R. Meyer, *Solution of Nonlinear Two-Point Boundary Value Problems by Linear Programming*, Computer Science Technical Report #1, University of Wisconsin, Madison, Jan. 1967.
23. J. B. Rosen and R. Meyer, *Solution of Nonlinear Two-Point Boundary Value Problems by Linear Programming*, in *Mathematical Theory of Control* (A. V. Balakrishnan and L. W. Neustadt, eds.), Academic Press, New York, 1967, pp. 71—84.
- 24*. Л. И. Шатровский, Об одном численном методе решения задач оптимального управления, *Журнал вычислительной математики и математической физики* **2**, № 3 (1962).
- 25*. Г. М. Островский, Об одном методе решения вариационных задач, *Автоматика и телемеханика* **23**, № 10 (1962).
- 26*. Р. П. Федоренко, Приближенное решение некоторых задач оптимального управления, *Журнал вычислительной математики и математической физики* **4**, № 6 (1965).
- 27*. В. С. Орлов, Б. Т. Поляк, В. А. Ребрий, Н. В. Третьяков, Опыт решения задач оптимального управления, в сб. «Вычислительные методы и программирование», **9**, Изд-во МГУ (1967).
- 28*. Ю. М. Ермольев, В. П. Гуленко, О численных методах решения задач оптимального управления, *Кибернетика*, № 1, 1966.
- 29*. И. О. Мельц, Применение методов нелинейного программирования для оптимизации динамических систем в функциональном пространстве, *Автоматика и телемеханика*, № 1, 1968.
- 30*. Г. М. Островский, Ю. М. Волин, Методы оптимизации химических реакторов, «Химия», 1967.
- 31*. Н. Н. Моисеев, Численные методы в теории оптимальных систем, «Наука», 1971.

7.1. Введение

Линейная система с дискретным временем при постоянном временном интервале может быть описана следующей системой разностных уравнений состояния:

$$\mathbf{y}(k+1) = A\mathbf{y}(k) + B\mathbf{u}(k), \quad (7.1)$$

где $\mathbf{y}(k)$ — n -мерный вектор состояния в дискретный момент $t=t_k$, $\mathbf{u}(k)$ — m -мерный вектор управления при $t=t_k$, A — $(n \times n)$ -мерная постоянная матрица, B — $(n \times m)$ -мерная постоянная матрица.

Вектор состояния при любом $t=t_k$ может быть представлен в виде функции начального состояния и всех предшествующих векторов управления [1]:

$$\mathbf{y}(k) = A^k \mathbf{y}(0) + \sum_{i=0}^{k-1} A^{k-i-1} B \mathbf{u}(i). \quad (7.2)$$

Рассмотрим теперь формулировку задачи оптимального управления в виде задачи математического программирования, подобную приведенной в (4.26). Из уравнения (7.2) видно, что ограничения в форме равенств в задаче (4.26), основанные на уравнениях состояния системы, теперь оказываются линейными. Эти ограничивающие уравнения являются строго линейными, так как никакие приближения не вводились. Кроме того, вектор состояния $\mathbf{y}(i)$ при любом $t=t_i$ ($i=1, \dots, N$) можно представить функцией вектора $\mathbf{y}(0)$ начального состояния, которое обычно задается, и последовательности векторов управления $\mathbf{u}(i)$ ($i=0, 1, \dots, N-1$). Это сокращает общее число рассматриваемых переменных задачи математического программирования на nN , что может существенно повысить эффективность численного решения. Если все дополнительные ограничения являются линейными функциями переменных состояния или управления, то задача оптимального управления, относящаяся к системе, описываемой уравнением (7.1)

или (7.2), оказывается фактически задачей линейного или квадратичного программирования в зависимости от формулировки показателя качества (см. гл. 2).

Поэтому совершенно естественно, что методы математического программирования должны быть применены в первую очередь к задачам управления [2, 3], принадлежащим к классу задач, рассматриваемых в настоящем параграфе. Многочисленные публикации, посвященные применению математического программирования в оптимальном управлении, рассматривают системы типа (7.1) или (7.2) [4—14]. Задача, относящаяся к аналогичному классу импульсных систем с неравными и априорно неизвестными интервалами времени, недавно была решена с помощью линейного и квадратичного программирования [15, 16].

В следующих параграфах приведено несколько характерных примеров применения методов линейного и квадратичного программирования к системам оптимального управления с дискретным временем.

7.2. Примеры задач минимизации

В этом параграфе будет рассмотрено несколько примеров задач оптимального управления линейными системами с равными временными интервалами дискретизации. Уравнения состояния для всех примеров описываются формулой (7.1), а ограничивающие равенства в формулировке математического программирования имеют вид (7.2). Различие между задачами определяется в основном показателем качества, что и используется для классификации задач.

Задачи минимального расхода топлива [2, 3, 6, 8]

В задачах минимального расхода топлива в системах с дискретным временем показатель качества может быть записан следующим образом:

$$\text{минимизировать } J = \sum_{i=0}^{N-1} \sum_{j=1}^m c_{ij} |u_j(i)|, \quad (7.3)$$

где c_{ij} — постоянный коэффициент, N — общее число

рассматриваемых интервалов времени, $u_j(i)$ — j -ая компонента вектора управления $\mathbf{u}(i)$ в момент $t=t_i$.

Как указывалось в гл. 5, показатель качества типа (7.3) не является линейным выражением. Поэтому нельзя непосредственно применить линейное программирование. В гл. 5 было описано несколько методов модификации этого показателя качества, которые делают его линейным. Аналогичные методы применялись также в работах, посвященных системам с дискретным временем.

Например, Вален [2] рассмотрел случай скалярного управления $u(i)$ ($i=0, \dots, N-1$). Показатель качества был взят следующий:

$$\text{минимизировать } J = \sum_{i=0}^{N-1} c_i |u(i)|. \quad (7.4)$$

Были введены новые переменные $g(0), \dots, g(N-1)$ так, что

$$\left. \begin{aligned} -g(i) &\leq u(i), \\ g(i) &\geq u(i), \end{aligned} \right\} \quad i = 0, \dots, N-1. \quad (7.5)$$

В результате новый показатель качества оказался линейным:

$$\text{минимизировать } J = \sum_{i=0}^{N-1} c_i g_i. \quad (7.6)$$

Очевидно, что здесь применен метод ограничивающих переменных, описанный в 5.4.

Другой подход предложил Торнг [6]. Торнг также рассматривал скалярный управляющий сигнал и исходил из показателя качества вида (7.4). Исходные переменные управления $u(0), u(1), \dots, u(N-1)$ заменялись последовательностью $u_m(0), u_m(1), \dots, u_m(N-1)$, где

$$u_m(i) = u(i) + 1.$$

Тогда ограничение $|u(i)| \leq 1$ принимало вид

$$0 \leq u_m(i) \leq 2. \quad (7.7)$$

Далее вводились дополнительные переменные $q(i) \geq 0$ так, что

$$u_m(i) + q(i) = 2, \quad i = 0, 1, \dots, N-1. \quad (7.8)$$

Показатель качества минимума расхода топлива получал следующий вид:

$$\text{максимизировать } J = \sum_{i=0}^{N-1} [c_{ui}u_m(i) + c_{qi}q(i)], \quad (7.9)$$

где

$$c_{ui} = \begin{cases} 1, & \text{если } u_m(i) < 1, \\ 0, & \text{если } u_m(i) = 1, \\ -1, & \text{если } u_m(i) > 1, \end{cases}$$

$$c_{qi} = \begin{cases} 1, & \text{если } q(i) < 1, \\ 0, & \text{если } q(i) = 1, \\ -1, & \text{если } q(i) > 1. \end{cases}$$

Задача минимума

квадратичного показателя качества [10, 11, 14]

Типичный вид квадратичного показателя качества для рассматриваемых задач следующий [14]:

$$\text{минимизировать } J = \sum_{i=1}^N [y^T(i) Q y(i) + u^T(i-1) R u(i-1)], \quad (7.10)$$

где Q и R — постоянные матрицы с размерами соответственно $n \times n$ и $m \times m$.

Рассмотрим, например, следующую систему второго порядка [14]:

$$\begin{bmatrix} y_1(k+1) \\ y_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 1 - e^{-T} \\ 0 & e^{-T} \end{bmatrix} \begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} + \begin{bmatrix} e^{-T} + T - 1 & 0 \\ 1 - e^{-T} & 0 \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix}. \quad (7.11)$$

Очевидно, что (7.11) имеет тот же вид, что и (7.1). В данном случае $n=m=2$. Максимальное число интервалов времени взято равным $N=16$, а каждый интервал равен $T=0,25 \text{ сек.}$ Дано, что начальное состояние равно

$$y(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad (7.12)$$

а конечное, целевое состояние есть начало координат:

$$y(16) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (7.13)$$

Кроме того, управляющие сигналы ограничены следующим образом:

$$-1 \leq u_j(i) \leq 1, \quad j=1, 2; \quad i=0, 1, \dots, 15. \quad (7.14)$$

Показатель качества:

$$\text{минимизировать } J = \frac{1}{2} \sum_{k=1}^{16} [y_1^2(k) + y_2^2(k) + u_1^2(k-1)]. \quad (7.15)$$

Легко видеть, что все ограничения, определяемые формулами от (7.11) до (7.12) — линейны, а показатель качества (7.15) — квадратичный. Следовательно, мы имеем задачу квадратичного программирования. Задача решалась на ЦВМ ИБМ 7094 с помощью алгоритма

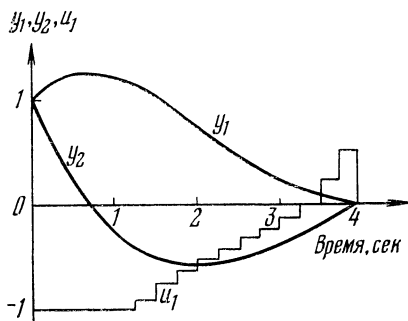


Рис. 7.1. Результаты решения задачи минимизации квадратичного показателя качества [14].

Катлера RSQPF 4 [17]. Результаты расчетов показаны на рис. 7.1.

Полученные значения $u_1(i)$ приведены в следующей таблице.

i	$u_1(i)$	i	$u_1(i)$
0	-1,000000	8	-0,402747
1	-1,000000	9	-0,411670
2	-1,000000	10	-0,324881
3	-1,000000	11	-0,230901
4	-1,000000	12	-0,116078
5	-0,905836	13	+0,036745
6	-0,739215	14	+0,249919
7	-0,608510	15	+0,553175

Задачи на быстродействие [6, 10, 11]

Аналогично решению задачи оптимизации по быстродействию, изложенному в 5.6, решим ту же задачу для систем рассматриваемого класса путем решения ряда последовательных подзадач, применяя методы линейного или квадратичного программирования. Для иллюстрации такого подхода рассмотрим решение Торнга [6]. Рассмотрим линейные системы с равными интервалами дискретного времени. Формулируя задачу, получим систему линейных ограничений следующего вида (см. 3.1):

$$\left. \begin{aligned} a_{11}x_1 + \dots + a_{1n}x_n + q_1 &= b_1, \\ \dots & \\ a_{k1}x_1 + \dots + a_{kn}x_n + q_k &= b_k, \\ a_{k+1,1}x_1 + \dots + a_{k+1,n}x_n &= b_{k+1}, \\ \dots & \\ a_{m1}x_1 + \dots + a_{mn}x_n &= b_m. \end{aligned} \right\} \quad (7.16)$$

Переменные q_1, \dots, q_k являются *дополнительными переменными* (см. 3.1), так как первые k уравнений (7.16) первоначально были неравенствами, в то время как последние $m-k$ уравнений сначала были равенствами. Чтобы получить единичную матрицу с размерами $m \times m$ в качестве базисного допустимого решения, надо ввести $m-k$ дополнительных искусственных переменных так, чтобы $m-k$ уравнений в (7.16) приняли вид

$$\left. \begin{aligned} a_{k+1,1}x_1 + \dots + a_{k+1,n}x_n + a_1 &= b_{k+1}, \\ \dots & \\ a_{m1}x_1 + \dots + a_{mn}x_n + a_{m-k} &= b_m. \end{aligned} \right\} \quad (7.17)$$

Первые k уравнений в (7.16) остаются без изменений.

Теперь формулируется следующая подзадача линейного программирования:

$$\text{максимизировать } J = - \sum_{i=1}^{m-k} a_i \quad (7.18)$$

при условии выполнения всех заданных ограничений. В конечном счете, искусственные переменные не будут содержаться в базисе.

Попробуем решить эту задачу для $N=1$. Если найдено допустимое решение, то это означает, что минимальное время равно

$$t_{\min} = T.$$

Если допустимое решение нельзя найти при исключении искусственных переменных, то это означает, что нельзя достигнуть конечного состояния за время $t=T$. Тогда пробуем $N=2$ и т. д. до тех пор, пока при некотором $N=N_0$ получим первое из допустимых решений, не содержащих искусственных переменных. В этом случае минимальное время равно

$$t_{\min} = N_0 T.$$

Непосредственное применение математического программирования к решению задач на быстродействие дано в [15, 18, 19] (см. гл. 6). Однако в этом случае пришлось применять нелинейное программирование.

В работах [6, 10, 11], на которые мы ссылались, отмечается возможность существования более одного допустимого управления, переводящего систему в заданное целевое множество за минимальное число N_0 интервалов дискретного времени. После получения N_0 решается дополнительная задача минимизации. Для выбора оптимального управления в [6] решается задача получения минимального расхода топлива, а в [10, 11] задача о минимуме энергии (квадратичный показатель качества).

Задачи минимальной ошибки [2, 4]

Задача минимизации ошибки, равной разности между конечным состоянием системы $y(N)$ и заданным состоянием y_a , была рассмотрена Валеном [2]. Вектор

ошибки e , размерность которого предполагается равной K , определяется следующим образом:

$$e = D[y(N) - y_d], \quad (7.19)$$

где D есть $(K \times n)$ -мерная постоянная матрица. Показатель качества формулируется следующим образом:

$$\text{минимизировать } J = \sum_{i=1}^K e_i. \quad (7.20)$$

Ясно, что показатель качества (7.20) является линейной функцией переменных состояния системы в конечный момент $t = t_N$. Применяя (7.2), можно видеть, что показатель качества также является линейной функцией последовательности векторов управления $u(i)$. Если все другие ограничения задачи линейны, то без труда может быть применен алгоритм линейного программирования.

Другой показатель качества применял Пропой [4]. Одна из переменных состояния y_1 рассматривается как выходная переменная. Ошибки системы для любого $t = t_i$ определяются следующим образом:

$$e_i = y_0(i) - y_1(i), \quad (7.21)$$

где $y_0(i)$ — заданное состояние, с которым производится сравнение. В общем случае оно может зависеть от времени.

Показатель качества формулируется следующим образом:

$$\text{минимизировать } J = \sum_{i=1}^N e_i. \quad (7.22)$$

Вновь получаем линейную функцию переменных состояния, а следовательно, и управления. Чтобы предотвратить возможность появления перелета, в обоих случаях должны быть наложены специальные ограничения

$$e_i \geq 0, \quad i = 1, \dots, N \text{ (или } i = 1, \dots, K). \quad (7.23)$$

Этих ограничений можно не делать, если применяется квадратичная целевая функция.

7.3. Расчет цифрового регулятора

Настоящий параграф посвящен применению линейного и квадратичного программирования к расчету цифрового регулятора для линейной импульсной системы управления с равными интервалами времени. Изложение основывается на работах Порчелли и Фегли [5, 7, 9].

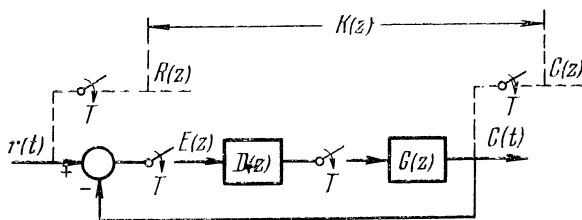


Рис. 7.2. Схема импульсной системы [9].

Структура рассматриваемой системы показана на рис. 7.2.

Объект представлен его z -преобразованной передаточной функцией $G(z)$. Аналогично, $D(z)$ представляет цифровой регулятор, а $K(z)$ — общий передаточный коэффициент замкнутой системы. Сигнал на входе системы задан в виде следующей последовательности импульсов:

$$R(z) = r_0 + r_1 z^{-1} + r_2 z^{-2} + \dots + r_n z^{-n} = \sum_{i=0}^n r_i z^{-i}, \quad (7.24)$$

где $r_i = r(t_i)$ обозначает известные амплитуды входных импульсов в моменты t_i ($i=0, 1, \dots, n$). Аналогично, общая передаточная функция системы определяется в виде

$$K(z) = A_m z^{-m} + A_{m+1} z^{-(m+1)} + \dots + A_{m+k} z^{-(m+k)}, \quad (7.25)$$

где m представляет неустранимую задержку в системе. Так как передаточная функция $D(z)$ регулятора не задана, то неизвестны и коэффициенты A_i ($i=m, \dots, m+k$) общей передаточной функции.

Основные уравнения системы, представленной на рис. 7.2, следующие:

$$C(z) = K(z)R(z), \quad (7.26)$$

$$E(z) = R(z) - C(z), \quad (7.27)$$

$$K(z) = \frac{C(z)}{R(z)} = \frac{D(z)G(z)}{1 + D(z)G(z)}. \quad (7.28)$$

Из (7.28) следует, что

$$D(z) = \frac{K(z)}{G(z)[1 - K(z)]}. \quad (7.29)$$

Обычно объект $G(z)$ задан. Если $K(z)$ известно, то с помощью (7.29) можно найти выражение $D(z)$ для цифрового регулятора. Ниже рассматривается метод определения с помощью линейного или квадратичного программирования коэффициентов A_i передаточной функции $K(z)$.

Объединяя (7.26) и (7.27) и используя (7.24) и (7.25) в полученном уравнении, можно, приравняв коэффициенты при одинаковых степенях z^{-1} , получить следующее матрично-векторное уравнение

$$RA + e = r, \quad (7.30)$$

где

$$e = \begin{bmatrix} e_m \\ e_{m+1} \\ \dots \\ e_n \end{bmatrix}, \quad r = \begin{bmatrix} r_m \\ r_{m+1} \\ \dots \\ r_n \end{bmatrix}, \quad A = \begin{bmatrix} A_m \\ A_{m+1} \\ \dots \\ A_n \end{bmatrix},$$

$$R = \begin{bmatrix} r_0 & 0 & 0 & \dots & 0 \\ r_1 & r_0 & 0 & \dots & 0 \\ r_2 & r_1 & r_0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ r_k & r_{k-1} & r_{k-2} & \dots & r_0 \\ \dots & \dots & \dots & \dots & \dots \\ r_{n-m} & r_{n-m-1} & r_{n-m-2} & \dots & r_{n-m-k} \end{bmatrix} -$$

$-\ [(n+1-m) \times (k+1)] = \text{матрица.}$

Компоненты ошибки e_i ($i=m, \dots, n$), как и коэффициенты A_i ($i=m, \dots, m+k$), являются неизвестными переменными задачи. Ясно, что уравнения (7.30) образуют систему линейных ограничений в форме равенств, состоящую из $(n+1-m)$ уравнений с $(n+k-m-2)$

переменными. В алгоритмах линейного программирования обычно предполагается, что все переменные неотрицательны (см. гл. 3). В данном же случае e_i и A_i могут быть как положительными, так и отрицательными. Чтобы фактические переменные задачи были неотрицательными, надо ввести следующие переменные:

$$A_i = A_i^+ - A_i^- \quad \text{для всех } i, \quad (7.31)$$

$$e_i = e_i^+ - e_i^- \quad \text{для всех } i, \quad (7.32)$$

где

$$A_i^+, A_i^-, e_i^+, e_i^- \geq 0 \quad \text{для всех } i. \quad (7.33)$$

Подставляя (7.31) и (7.32) в (7.30), получим

$$RA^+ - RA^- + e^+ - e^- = r. \quad (7.34)$$

Кроме ограничений (7.32) и (7.34) наложены еще следующие ограничения:

$$-E_m \leq (e_i^+ - e_i^-) \leq E_m \quad \text{для всех } i. \quad (7.35)$$

Чтобы применить алгоритм линейного программирования, нужно ввести линейную целевую функцию. Исходная целевая функция имела вид взвешенной суммы абсолютных значений ошибок:

$$\text{минимизировать } J = \sum_{i=m}^n c_i |e_i| = \sum_{i=m}^n c_i |e_i^+ - e_i^-|. \quad (7.36)$$

Нетрудно видеть, что в оптимальном решении либо e_j^+ , либо e_j^- , или обе эти величины равны нулю. Следовательно, $|e_j^+ - e_j^-|$ можно заменить на $e_j^+ + e_j^-$, и записать теперь целевую функцию в виде:

$$\text{минимизировать } J = \sum_{i=m}^n c_i (e_i^+ + e_i^-). \quad (7.37)$$

Выражение (7.37) является линейным относительно переменных задачи e_i^+ и e_i^- , и для численного решения этой задачи можно непосредственно применить алгоритм линейного программирования.

Можно сформулировать также квадратичный показатель качества для минимизации взвешенной суммы квадратов ошибок в следующем виде [9]:

$$\text{минимизировать } J = \sum_{i=m}^n c_i (e_i^+ - e_i^-)^2. \quad (7.38)$$

Для численного решения этой задачи был применен алгоритм квадратичного программирования RSQF 4 [17].

Ниже приводятся три примера, иллюстрирующие осуществление метода.

Пример 1. Объект цифровой системы управления, представленной на рис. 7.2, обладает передаточной функцией

$$G(z) = \frac{(1 + 2,34z^{-1})(1 + 0,16z^{-1})z^{-1}}{(1 - z^{-1})(1 - 0,368z^{-1})^2}.$$

Система должна удовлетворять следующим требованиям: (1) входной сигнал представляет собой единичную линейно растущую функцию; (2) все переменные не имеют ограничений по знаку; (3) расчет основывается на первых 15 дискретных моментах, а период равен 1 сек; (4) представление функции $K(z)$ ограничивается пятью членами, т. е. $K(z) = A_1z^{-1} + A_2z^{-2} + A_3z^{-3} + A_4z^{-4} + A_5z^{-5}$; (5) наложены следующие ограничения:

$$|e_2| \leq 1, \quad |e_j| \leq 0,5, \quad j=3, 4, \dots, 15.$$

В системе типа рис. 7.2, как показано в [1], при заданном $G(z)$ общая передаточная функция $K(z)$ должна обладать нулями, совпадающими со всеми нулями функции $G(z)$, которые в плоскости z лежат вне единичной окружности или на ней, а $1-K(z)$ должна иметь нули, совпадающие со всеми полюсами функции $G(z)$, лежащими в плоскости z вне единичной окружности или на ней. Как показано в [7], из этих двух необходимых условий следует ограничение в форме равенств, которое, применительно к нашему примеру, имеет вид

$$HA^+ - HA^- = \delta,$$

где

$$H = \begin{bmatrix} 4,16 & -1,34 & 1 & 0 & 0,4275 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

и

$$\delta = [0,3, 1].$$

Задача решается вначале при чисто линейной целевой функции вида

$$\min J = \min \sum_{i=1}^{15} i (e_i^+ + e_i^-)$$

и, далее, при чисто квадратичной целевой функции вида

$$\min J = \min \sum_{i=1}^{15} (e_i^+ - e_i^-)^2.$$

Для случая линейной целевой функции получены следующие результаты:

$$A_1 = 1,000000, \quad e_1 = 1,000000,$$

$$A_2 = 1,455090, \quad e_2 = 1,000000,$$

$$A_3 = -1,910180, \quad e_3 = -0,455090.$$

$$A_4 = 0,455090,$$

Все остальные переменные равны нулю. Целевая функция равна

$$J = 4,365279.$$

Общая импульсная передаточная функция равна

$$K(z) = z^{-1} + 1,455090z^{-2} + 1,910180z^{-3} + 0,455090z^{-4}.$$

Результаты, полученные для квадратичной целевой функции:

$$A_1 = 1,018104, \quad e_6 = -0,016549,$$

$$A_2 = 1,382101, \quad e_7 = -0,013629,$$

$$A_3 = -1,997295, \quad e_8 = -0,010708,$$

$$A_4 = 0,795343, \quad e_9 = -0,007788,$$

$$A_5 = -0,201173, \quad e_{10} = -0,004868,$$

$$e_1 = 1,000000 \quad e_{11} = -0,001948,$$

$$e_2 = 0,981896 \quad e_{12} = 0,000973,$$

$$e_3 = -0,418308 \quad e_{13} = 0,003893,$$

$$e_4 = 0,178783, \quad e_{14} = 0,006813,$$

$$e_5 = 0,019469, \quad e_{15} = 0,009733.$$

Целевая функция равна $J = 2,172283$.

Общая импульсная передаточная функция

$$K(z) = 1,018104z^{-1} + 1,382101z^{-2} - 1,997295z^{-3} + 0,795343z^{-4} - 0,201173z^{-5}.$$

Результаты для квадратичной целевой функции совершенно аналогичны результатам для линейной целевой функции. Однако сравнение результатов рис. 7.3 показывает, что в нашем примере линейная целевая функция приводит к несколько более быстрой реакции, чем квадратичная целевая функция.

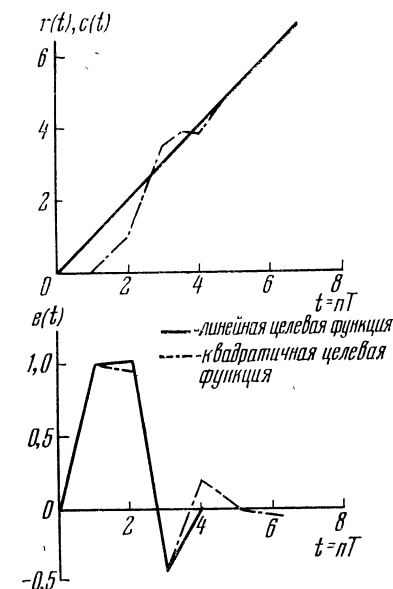


Рис. 7.3. Входной сигнал, импульсные сигналы выхода и ошибки для примера 1 [9].

Целевая функция приводит к несколько более быстрой реакции, чем квадратичная целевая функция.

Пример 2. Должна быть рассчитана система, объект которой не накладывает никаких ограничений на общую передаточную функцию. Должны быть выполнены следующие требования.

1) задан входной сигнал, представленный на рис. 7.4:

$$\begin{aligned} r(t) &= 0,1t^2, & 0 \leq t < 5T, \\ r(t) &= 0,1[50 - (t-10)^2], & 5T \leq t < 15T, \\ r(t) &= 0,1(t-20)^2, & 15T \leq t < 20T, \\ r(t) &= 0, & 20T \leq t; \end{aligned}$$

2) все переменные не имеют ограничений;

3) расчет основывается на первых 24 моментах отсчетов;

4) представление $K(z)$ ограничено тремя членами и имеет следующий вид:

$$K(z) = A_1 z^{-1} + A_2 z^{-2} + A_3 z^{-3}.$$

Должны быть получены два решения: первое — для чисто линейной целевой функции вида

$$\min J = \min \sum_{i=1}^{24} (e_i^+ + e_i^-)$$

и второе — для чисто квадратичной целевой функции вида

$$\min J = \min \sum_{i=1}^{24} (e_i^+ - e_i^-)^2.$$

Результаты для линейной целевой функции:

$$\begin{aligned} A_1 &= 2,999990, & e_{16} &= 0,199997, \\ A_2 &= -2,999982, & e_{17} &= 0,199998, \\ A_3 &= 0,999992, & e_{21} &= -0,099999, \\ & & e_{22} &= -0,099999, \end{aligned}$$

Все остальные ошибки равны нулю. Целевая функция равна $J = 1,200003$.

Общая импульсная передаточная функция

$$K(z) = 2,999990z^{-1} - 2,999982z^{-2} + 0,999992z^{-3}.$$

Результаты для квадратичной целевой функции:

$$\begin{aligned} A_1 &= 2,763373, & e_5 &= 0,102032, & e_{12} &= 0,014306, & e_{18} &= -0,036397, \\ A_2 &= -2,632494, & e_6 &= -0,074529, & e_{13} &= -0,034998, & e_{19} &= -0,019369, \\ A_3 &= 0,864540, & e_7 &= -0,097428, & e_{14} &= -0,056605, & e_{20} &= -0,001426, \\ e_1 &= 0,100000, & e_8 &= 0,059299, & e_{15} &= -0,079129, & e_{21} &= -0,082567, \\ e_2 &= 0,123663, & e_9 &= 0,042272, & e_{16} &= 0,097431, & e_{22} &= -0,086454, \\ e_3 &= 0,057900, & e_{10} &= 0,024329, & e_{17} &= 0,120400, & e_{23} &= e_{24} = 0, \\ e_4 &= 0,079508, & e_{11} &= 0,005469, & & & & \end{aligned}$$

Целевая функция $J = 0,117240$.

Общая импульсная передаточная функция

$$K(z) = 2,763373z^{-1} - 2,632494z^{-2} + 0,864540z^{-3}.$$

Оба результата представлены на рис. 7.4.

Пример 3. Данный пример аналогичен примеру 2; единственное отличие заключается в том, что теперь $K(z)$ имеет пять членов вместо

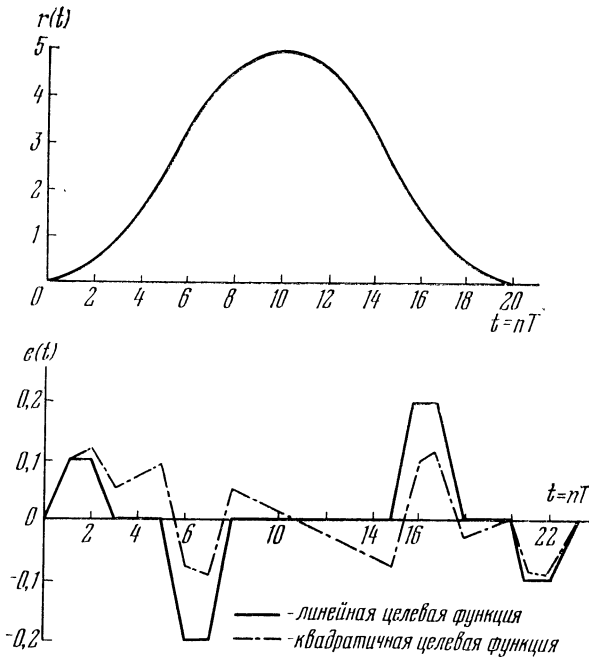


Рис. 7.4. Входной сигнал и ошибка для примера 2 [9].

трех, а линейная и квадратичная целевые функции содержат веса, зависящие от времени. Итак, имеем

$$K(z) = A_1 z^{-1} + A_2 z^{-2} + A_3 z^{-3} + A_4 z^{-4} + A_5 z^{-5},$$

$$\min J = \min \sum_{i=1}^{24} i (e_i^+ + e_i^-), \quad \min J = \min \sum_{i=1}^{24} i (e_i^+ - e_i^-)^2.$$

Результаты для линейной целевой функции:

$A_1 = 2,872985,$	$e_3 = 0,096601,$	$e_{11} = 0,022182,$	$e_{19} = 0,071556,$
$A_2 = -3,457953,$	$e_4 = 0,153489,$	$e_{13} = -0,025045,$	$e_{20} = 0,030054,$
$A_3 = 2,440063,$	$e_5 = 0,155278,$	$e_{14} = -0,052952,$	$e_{21} = -0,050626,$
$A_4 = -1,159208,$	$e_6 = -0,011091,$	$e_{15} = -0,083721,$	$e_{22} = -0,041145,$
$A_5 = 0,289802,$	$e_8 = 0,071556,$	$e_{16} = 0,082647,$	$e_{24} = -0,028980,$
$e_1 = 0,100000,$	$e_{10} = 0,041503,$	$e_{17} = 0,071556,$	$e_7 = e_9 = e_{12} =$
$e_2 = 0,112702,$			$= e_{18} = e_{23} = 0.$

Целевая функция $J=12,789630$.

$$K(z) = 2,872985z^{-1} - 3,457953z^{-2} + 2,440063z^{-3} - \\ - 1,159208z^{-4} + 0,289802z^{-5}.$$

Результаты для квадратичной целевой функции:

$A_1 = 3,067555,$	$e_3 = 0,058892,$	$e_{10} = 0,022914,$	$e_{17} = 0,056519,$
$A_2 = -3,859141,$	$e_4 = 0,120514,$	$e_{11} = 0,006429,$	$e_{18} = -0,041168,$
$A_3 = 2,623430,$	$e_5 = 0,121991,$	$e_{12} = -0,011841,$	$e_{19} = 0,051309,$
$A_4 = -1,062250,$	$e_6 = -0,052598,$	$e_{13} = -0,031897,$	$e_{20} = 0,021713,$
$A_5 = 0,221480,$	$e_7 = -0,011891,$	$e_{14} = -0,053738,$	$e_{21} = -0,061801,$
$e_1 = 0,100000,$	$e_8 = 0,085795,$	$e_{15} = -0,077364,$	$e_{22} = -0,036775,$
$e_2 = 0,093244,$	$e_9 = -0,006682,$	$e_{16} = 0,097225,$	$e_{23} = 0,017633,$
			$e_{24} = -0,022148.$

Целевая функция $J=0,822381$.

$$K(z) = 3,067555z^{-1} - 3,859141z^{-2} + 2,623430z^{-3} - \\ - 1,062250z^{-4} + 0,221480z^{-5}.$$

Ошибка показана на рис. 7.5 для обоих решений.

Сравнение рис. 7.4 и рис. 7.5 показывает, что большее число членов в выражении для $K(z)$ дает более сглаженную функцию ошибки и что зависящие от времени веса в целевых функциях дают более

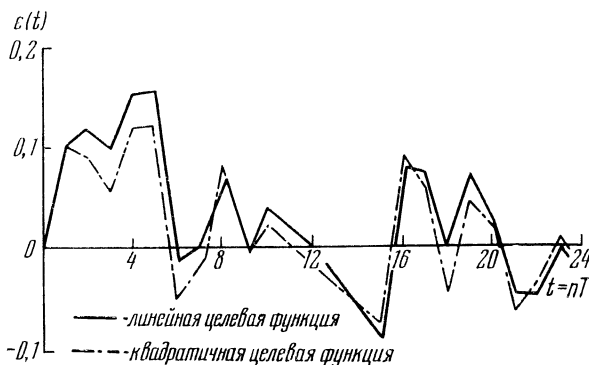


Рис. 7.5. Ошибка для примера 3 [9].

быстрое уменьшение амплитуды ошибки. Результаты этих двух примеров показывают, что различие между решениями с помощью линейного и квадратичного программирования уменьшается с ростом числа удерживаемых членов общей передаточной функции $K(z)$.

7.4. Системы, управляемые в дискретные моменты при неизвестных интервалах

В настоящем параграфе рассмотрим линейную систему, воздействие на которую осуществляется только в дискретные моменты времени, число которых конечно [15, 16]. Это значит, что система, вообще говоря, может рассматриваться как непрерывная, описываемая системой линейных уравнений состояния вида

$$\dot{\mathbf{x}} = A\mathbf{x}, \quad (7.39)$$

где \mathbf{x} — n -мерный вектор состояния, A — $(n \times n)$ -мерная постоянная матрица.

В некоторые дискретные моменты t_1, \dots, t_N происходят скачки (разрывы непрерывности) значений переменных состояния. Эти скачки рассматриваются как переменная управления, а все скачки всех компонент вектора \mathbf{x} образуют вектор управления, действующего в дискретные моменты времени t_1, \dots, t_N , которые априорно неизвестны.

Промежуток времени между появлением смежных разрывов непрерывности вектора состояния называется непрерывным периодом или циклом. Каждый цикл имеет определенную, но априорно неизвестную длительность T_i ($i=1, \dots, N$). Моменты t_1, \dots, t_N , когда воздействует управление, называются моментами управления или квантования. Очевидно, что

$$T_i = t_i - t_{i-1}, \quad i=1, \dots, N, \quad (7.40)$$

где N — максимальное число рассматриваемых непрерывных периодов или циклов.

Вектор состояния в момент $t=t_i$ обозначен $\mathbf{x}(i)$, вектор управления — $\mathbf{u}(i)$. j -ую компоненту векторов состояния и управления обозначим соответственно $x_j(i)$ и $u_j(i)$.

Зависимость от времени векторов состояния и управления схематически можно представить следующим образом.

Циклы	T_1		T_2		T_{N-1}		T_N
Момент времени	t_0	t_1	t_2	...	t_{N-2}	t_{N-1}	t_N
Вектор состояния	$x(0)$	$x(1)$	$x(2)$...	$x(N-2)$	$x(N-1)$	$x(N)$
Вектор управления	$u(0)$	$u(1)$	$u(2)$...	$u(N-2)$	$u(N-1)$	$u(N)$

Так как $t=t_N$ есть нефиксированное время окончания процесса и никакие дальнейшие изменения конечного состояния $x(N)$ не рассматриваются, то $u(N)=0$. Предполагается, что начальное состояние $x(0)$ известно. Однако по причинам, которые станут ясными из дальнейшего, предполагается, что $u(0)=x(0)$, вопреки общему определению вектора управления как дискретного изменения вектора состояния. Это определение вступит в силу, начиная только с $t=t_1$.

Решение уравнения (7.39) для любого t , начиная с $t=t_0$, равно

$$x(t) = e^{A(t-t_0)} x(t_0), \quad (7.41)$$

где $e^{A(t-t_0)}$ представляет собой $(n \times n)$ -мерную матрицу.

В конце первого цикла при $t=t_1$ вектор состояния принимает конечное значение

$$x_f(1) = x(t_1) = e^{A(t_1-t_0)} x(0) = e^{AT_1} x(0). \quad (7.42)$$

В этот момент подается управление $u(1)$, в результате чего состояние x получает свое фактическое значение при $t=t_1$:

$$x(1) = x_f(1) + u(1) = e^{AT_1} x(0) + u(1). \quad (7.43)$$

Таким образом, $u(1)$ вызывает дискретное изменение значения $x_f(1)$ вектора состояния. Этот разрыв непрерывности можно выразить следующим образом:

$$x(t_1-0) = x_f(1), \quad x(t_1+0) = x(1).$$

Аналогично обстоит дело во всех моментах управления, так что мы можем записать следующее общее разностное уравнение:

$$x(i) = e^{AT_i} x(i-1) + u(i) \quad i = 1, \dots, N. \quad (7.44)$$

Переменными разностного уравнения состояния (7.44) являются не только векторы управления и состояния, но и продолжительность цикла между моментами управления. Учитывая, что все периоды T_i ($i=1, \dots, N$) неизвестны, мы сталкиваемся с трудной задачей определения этих периодов. Мы предлагаем итерационный алгоритм для определения неизвестных периодов путем использования других свойств и ограничений, определяющих систему. Ниже, в этом параграфе, будет описан этот алгоритм и рассмотрены дополнительные свойства системы.

Как в любой задаче оптимального управления, должен быть задан показатель качества. В этой задаче надо минимизировать или максимизировать показатель качества следующего общего вида:

$$J = \sum_{i=1}^N F[x(i), u(i)]. \quad (7.45)$$

При этом должны выполняться определенные ограничения, зависящие от переменных состояния и управления. В частности, в начале каждого цикла должны удовлетворяться следующие ограничения начала:

$$g_{in}[x(i), u(i)] \geq 0, \quad i=0, \dots, N-1. \quad (7.46)$$

Ни один цикл не может начаться, пока не будет выполнено неравенство (7.46). С другой стороны, начавшийся цикл продолжается пока удовлетворено следующее ограничение окончания:

$$g_i[x_f(i)] \geq 0, \quad i=1, \dots, N. \quad (7.47)$$

На систему наложено еще условие, заключающееся в том, что она должна находиться в непрерывных циклах наибольшее время, возможное без нарушения ограничения (7.47). Поэтому в конце каждого периода ограничение фактически принимает форму равенства. Как только условие (7.47) нарушается, цикл заканчивается и происходит воздействие управления. Когда после подачи управления оказывается выполненным условие (7.46), начинается новый цикл.

Теперь, после того как завершено определение задачи, можно сформулировать алгоритм вычислений:

1. Решить численно уравнение (7.39), используя заданное начальное условие $\mathbf{x}(0)$. При каждом шаге интегрирования проверять, выполняется ли условие $g_t(\mathbf{x}) \geq 0$. Как только $g_t(\mathbf{x}) < 0$, остановить процесс интегрирования и записать конечное время T_1 , равное длительности первого периода.

2. Положить все периоды равными T_1 :

$$T_1 = T_2^{(1)} = T_3^{(1)} = \dots T_N^{(1)}.$$

Верхний индекс (1) указывает на первую итерацию.

3. Подставить $T_i^{(1)}$ ($i=1, \dots, N$) в разностные уравнения состояния (7.44). Используя (7.44), можно выразить каждый вектор состояния для любого момента t_i в виде функции предыдущего и текущего векторов управления $\mathbf{u}(j)$ ($j=1, \dots, i$) и начального вектора $\mathbf{x}(0) = \mathbf{u}(0)$ следующим образом:

$$\mathbf{x}(i) = \sum_{m=0}^{i-1} \left[\prod_{j=m+1}^i \exp(AT_j) \right] \mathbf{u}(m) + \mathbf{u}(i). \quad (7.48)$$

Далее значение $\mathbf{x}(i)$ (7.48) подставить в показатель качества (7.45) и в ограничения (7.46) и (7.47), и во все дополнительные ограничения.

4. Решается задача математического программирования для определения экстремума показателя качества (7.45) при выполнении всех ограничений. В результате получаем множество оптимальных векторов $\mathbf{u}^{(1)}(m)$ ($m=1, \dots, N-1$). В частности, если все ограничения и показатель качества линейны, то задача сводится к линейному программированию.

5. Теперь решается уравнение (7.39) для всех N периодов. Каждый период заканчивается, как только $g_t(\mathbf{x}) < 0$. Пусть $\mathbf{x}_f(i)$ есть значение в конце i -го периода. Тогда начальное условие для следующего интегрирования будет $\mathbf{x}(i) = \mathbf{x}_f(i) + \mathbf{u}(i)$ и т. д. После решения для всех циклов, получаем новое множество периодов $T_1^{(2)}, T_2^{(2)}, \dots, T_N^{(2)}$. Отметим, что $T_1^{(2)} = T_1$, так как начальный вектор $\mathbf{x}(0)$ предполагается постоянным. В общем случае можно изменять $\mathbf{x}(0)$, как и T_1 . При этом алгоритм остается прежним.

Используя новые $T_i^{(2)}$ ($i = 1, \dots, N$), повторить шаги 3—5. В результате получим $T_i^{(3)}$ и т. д. Процесс продолжать, пока не получим

$$|T_i^{(k)} - T_i^{(k+1)}| \leq \epsilon > 0,$$

где ϵ — заданный критерий сходимости. Вектор $u^{(k)}(i)$ ($i = 1, \dots, N-1$) будет являться окончательным законом управления.

Сходимость этого алгоритма была установлена экспериментально.

Пример. Предложенный метод можно применить к задаче оптимизации топливных циклов в ядерных реакторах. Для наглядности была выбрана простейшая модель ядерного реактора на тепловых

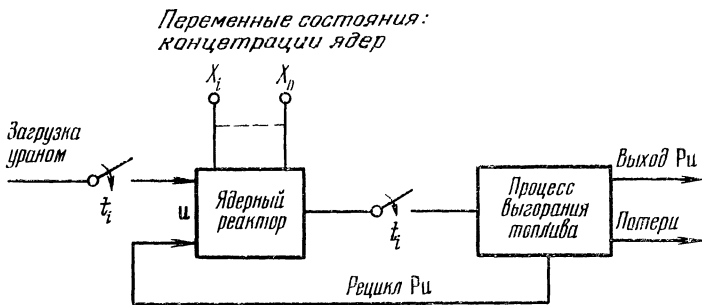


Рис. 7.6. Модель топливного цикла в ядерном реакторе [16].

нейтронах (рис. 7.6). Переменными состояниями являются теперь концентрации рассматриваемых материалов, а переменными управления являются дискретные изменения этих концентраций в моменты догрузки (рис. 7.7).

Уравнения состояния, с текущим временем в качестве независимой переменной, совершенно аналогично уравнению (7.39). Матрица A зависит от ядерных свойств рассматриваемых материалов, как например, от эффективного сечения.

Для успешной работы ядерного реактора необходимо, чтобы коэффициент размножения $k(x)$ был больше единицы. Обычно задают некоторое минимальное значение коэффициента k , необходимое для того, чтобы начался цикл облучения. В нашем случае, по аналогии с ограничением (7.46),

$$k_{in}(x) \geq 1,08. \quad (7.49)$$

Ограничением, определяющим конец каждого цикла облучения, будет неравенство

$$k_f(x) \geq 1,0001, \tag{7.50}$$

аналогичное ограничению (7.47). Как только неравенство (7.50) оказывается нарушенным, необходимо реактор остановить и произвести догрузку топлива.

Основным топливом, расщепляющимся в ядерном реакторе является U-235. Он входит в естественный уран в концентрации, составляющей 0,72%. Обогащение урана, для увеличения содержания U-235

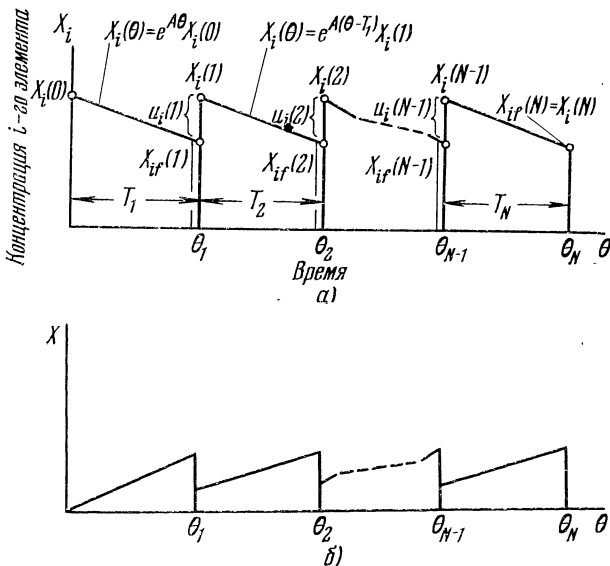


Рис. 7.7. Зависимость концентрации от времени [16]: а) элемент, подвергавшийся обеднению (например, U-235), б) образующейся элемент (например, Pu-239).

в нем, является очень дорогим процессом. Поэтому важно, чтобы реактор работал при минимальном расходе U-235. С другой стороны, в ядерном реакторе в процессе облучения, в качестве побочного продукта образуется Pu-239, который может служить расщепляющимся топливом в том же реакторе или может извлекаться и использоваться для других целей. Возникает задача, заключающаяся в определении оптимального сочетания количества вносимого в реактор U-235 и извлекаемого Pu-239, обеспечивающего снижение общих расходов.

Рассматриваются два частных показателя качества:

P1. Минимизация расхода U-235 по всему диапазону циклов при требовании, чтобы общий выход Pu-239 был бы не меньше заданного

количества C . Этот показатель качества может быть представлен линейной и квадратичной формами.

Линейная форма:

$$J = \sum_{i=0}^{N-1} [x_1(i) - x_{f1}(i+1)]. \quad (7.51)$$

Квадратичная форма:

$$J = \sum_{i=0}^{N-1} [x_1(i) - x_{f1}(i+1)]^2. \quad (7.52)$$

Индекс 1 относится к U-235 — первой компоненте вектора x .

P2. Максимизация выхода Pu-239 при ограничении обогащения U-235 заданным значением.

Показатель качества определен линейной формой

$$J = \sum_{i=1}^{N-1} u_3(i) + x_{f3}(N). \quad (7.53)$$

Индекс 3 относится к Pu-239 — третьей компоненте вектора x .

Всего рассматриваются четыре переменных состояния:

x_1 (U-235),

x_2 (U-238),

x_3 (Pu-239).

x_4 — осколки деления.

Матрица A (7.39) для этой модели будет равна

$$A = \begin{bmatrix} -\sigma_{a1} & 0 & 0 & 0 \\ -f_1 & -\sigma_{a2} & -f_3 & 0 \\ 0 & \sigma_{a2} & -\sigma_{a3} & 0 \\ \sigma_{f1} & 0 & \sigma_{f3} & \sigma_{a4} \end{bmatrix},$$

где σ_{ai} — эффективная площадь поглощения частицы i , σ_{fi} — эффективная площадь расщепления частицы i , f_i — концентрация нейтронов, полученных в результате расщепления частиц i , сохранившихся после утечки и поглощения ураном U-238.

Для коэффициента размножения запишем следующее линейризованное выражение:

$$k = \frac{\sum_{i=1}^4 \gamma_i \sigma_{fi} x_i}{\sum_{i=1}^4 \sigma_{ai} x_i + \sum_s}, \quad (7.54)$$

где γ_i — число нейтронов, образуемых при расщеплении частицы i , \sum_s — эффективное сечение, представляющее все материалы, имею-

щиеся в реакторе, за исключением определяющих переменные состояния.

Для каждого дискретного момента $i=1, \dots, N-1$ рассматривались следующие ограничения:

С1. Начальное ограничение. Используя (7.49) и (7.54), имеем

$$\sum_{j=1}^4 \gamma_j \sigma_{fj} x_j(i) - 1,08 \sum_{j=1}^4 \sigma_{aj} x_j(i) - 1,08 \sum_s \geq 0. \quad (7.55)$$

С2. Ограничение окончания. Используя (7.50) и (7.54), имеем

$$\sum_{j=1}^4 \gamma_j \sigma_{fj}(i+1) - 1,001 \sum_{j=1}^4 \sigma_{aj} x_{fj}(i+1) - 1,0001 \sum_s \geq 0. \quad (7.56)$$

С3. Извлекаемое в момент i количество Pu-239 не должно быть больше количества Pu-239, полученного к концу i -го цикла:

$$u_3(i) \leq x_{f3}(i) \quad (7.57)$$

С4. Обогащение урана к началу каждого цикла не должно превосходить E_n %:

$$\frac{x_1(i)}{x_1(i) + x_2(i)} \leq \frac{E_n}{100}. \quad (7.58)$$

С5. Суммарный выход Pu-239 должен быть не меньше C (только в задаче P1):

$$\sum_{i=1}^{N-1} u_3(i) + x_{f3}(N) \geq C. \quad (7.59)$$

Во всех заданных ограничениях и показателях качества переменные состояния выражаются через переменные управления с помощью (7.43), (7.44) и (7.48). Таким образом, все ограничения имеют форму линейных неравенств. Поэтому в зависимости от формы J , заданной уравнениями от (7.51) до (7.53), может быть применено линейное или квадратичное программирование.

Заметим, что осколками деления реактор не загружается и, следовательно, $u_4(i) = 0$ при всех i . Таким образом, в каждый дискретный

Т а б л и ц а 7.1

Ядра	σ_a (барнс)	σ_f (барнс)	γ
1. U-235	362,4	309	2,42
2. U-238	1,56	0	0
3. Pu-239	1103,0	683	2,90
4. Осколки деления	2,0	0	0

Примечание. Значения выбраны с иллюстрированными целями. Они не обязательно представляют фактические значения, характерные для реальных реакторов.

момент рассматриваются лишь три переменных управления. Поэтому для рассматриваемых N циклов имеем всего $3(N-1)$ переменных, при $4(N-1)+1$ ограничениях для задачи P1 и при $4(N-1)$ ограничениях для задачи P2. Так как обычно производится извлечение Рн-239, то в уравнениях, вводимых в ЦВМ, $u_3(i)$ всегда имеет отрицательный знак.

Было решено четыре различных примера. Общее число рассматриваемых непрерывных циклов было равно $N=5$ при максимальном обогащении в 4%. Исходные данные приведены в табл. 7.1, а результаты решения задач — в табл. 7.2.

Таблица 7.2

Задача №	1	2	3	4
J	P1, квадратичный	P1, линейный	P2, линейный	P2, линейный
$x(0) =$	3,35 68,11 0 0	3,42 81,90 0 0	3,35 68,11 0 0	3,42 81,90 0 0
$E_i \%$	4,92	4,00	4,92	4,00
N_i	5	1	8	11
$u_0(1) =$	0,64141 14,11885 -0,03982 0	0,61118 0,45596 -0,04996 0	0,64497 14,20418 -0,04197 0	0,71842 22,89882 -0,05208 0
$u_0(2) =$	0,61668 0,42683 -0,05321 0	0,61355 0,29003 -0,05088 0	0,64131 6,47378 -0,05207 0	0,74749 17,38654 -0,06932 0
$u_0(3) =$	0,61681 0,35913 -0,05225 0	0,60910 0,17468 -0,04773 0	0,64647 4,32040 -0,05673 0	0,76435 11,63549 -0,08330 0
$u_0(4) =$	0,61377 0,21292 -0,05248 0	0,60164 0 -0,04186 0	0,64659 2,16447 -0,05984 0	0,76732 5,87588 -0,09296 0
$ J_0 $	0,92538	0,91181	0,14091	0,22490
T_1	0,51989	0,54249	0,51989	0,54249
T_2	0,54039	0,54249	0,54039	0,57439
T_3	0,54179	0,54249	0,54959	0,60088
T_4	0,54379	0,54249	0,55559	0,61868
T_5	0,54039	0,54249	0,55839	0,62728
$\delta T_j \%$	5	0	7	13,5
C	0,25	0,25	—	—
$E_n(1)$	4,00	4,00	4,00	4,00
$E_n(2)$	4,00	4,00	3,76	2,88
$E_n(3)$	4,00	4,00	3,62	2,68
$E_n(4)$	4,00	4,00	3,56	2,60

В табл. 7.2 J — показатель качества, x_0 — начальный вектор состояния, E_i — начальное обогащение, N_i — число итераций до сходимости, $u_0(i)$ — вычисленное значение вектора оптимального управления при $t=t_i$ ($i=1, \dots, N-1$) (перед $u_0(i)$ стоит знак минус, так как Рн-239 извлекается), $|J_0|$ — абсолютное значение оптимального показателя качества, T_j ($j=1, \dots, 5$) — вычисленные длины периодов непрерывности (в единицах времени, масштабированные на 10^3), δT_j — максимальное изменение T_j , C — требуемый выход Рн-239, $E_n(i)$ — обогащение i -й загрузки.

Для решения задачи линейного и квадратичного программирования были использованы программы, разработанные Отделением статистической службы Иллинойского университета.

Эти программы были записаны на специальном языке SSUPAC для ЦВМ ИБМ 7094. Задача линейного программирования решалась усовершенствованным симплекс-методом (см. гл. 3), а задача квадратичного программирования методом Вулфа, описанным в гл. 3.

Дополнительно был проведен анализ чувствительности. С этой целью значения переменных изменялись вблизи оптимума и контролировались отклонения показателей качества от их оптимальных значений. Результаты этого анализа приведены на рис. 7.8.

7.5. Импульсные системы с квантованным управлением [12]

Во многих практических системах управления применяется квантованный управляющий сигнал, т. е. сигнал, величина которого может иметь только фиксированные дискретные значения. Например, в системах с цифровым управлением, управляющая ЦВМ имеет конечное число разрядов. Само собою разумеется, что в результате этого ее выходные данные оказываются квантованными. В данном параграфе будет рассмотрен метод оптимизации

7 Д. Табак, Б. Куо

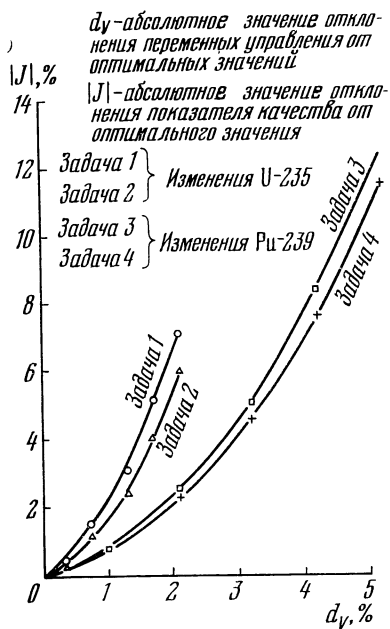


Рис. 7.8. Анализ чувствительности [16].

ции управления системами с дискретным временем и квантованными управляющими функциями, предложенный Кимом и Джаджури [12], а также его реализация с помощью математического программирования. Квантование предполагается равномерным, как показано на рис. 7.9. Амплитуда квантованного сигнала управления ограничена: $-u_m \leq u \leq u_m$.

Рассматриваемая система описывается следующими разностными уравнениями состояния:

$$\mathbf{y}(k) = \Phi(T)\mathbf{y}(k-1) + \mathbf{d}u(k), \quad k=1, \dots, N, \quad (7.60)$$

где $\mathbf{y}(k)$ — n -мерный вектор состояния при $t=t_k$, $\Phi(T)$ — $(n \times n)$ -мерная переходная матрица состояний, T — постоянный период дискретизации, \mathbf{d} — постоянный n -мерный вектор, $u(k)$ — скаляр, квантованный сигнал управления при $t=t_k$, удовлетворяющий условиям:

- 1) $u(k)$ постоянно на интервале $(k-1)T < t \leq kT$,
- 2) $u(k)$ может принимать только следующие значения (см. рис. 7.9): $-u_m, -u_m+r, \dots, -2r, -r, 0, r, 2r, \dots, u_m-r, u_m$.

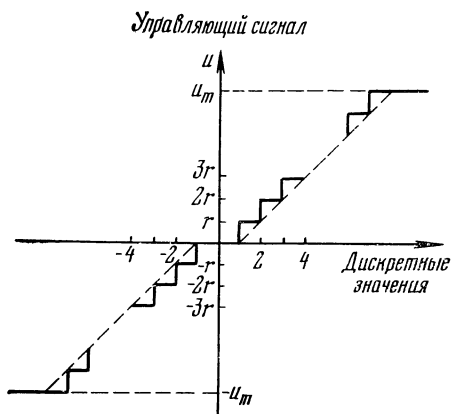


Рис. 7.9. Квантование сигнала управления [12].

Начальное состояние $\mathbf{y}(0)$ предполагается заданным. При любом $t=t_k$ состояние $\mathbf{y}(k)$ можно записать в виде

$$\mathbf{y}(k) = \Phi^k(T)\mathbf{y}(0) + \sum_{i=1}^k \Phi^{k-i}(T)\mathbf{d}u(i). \quad (7.61)$$

Чтобы упростить формулировку задачи, вводится следующая новая переменная:

$$f_k = \frac{[u(k) + u_m]}{r}, \quad (7.62)$$

или

$$u(k) = r f_k - u_m. \quad (7.63)$$

Таким образом, f_k всегда положительно и лежит в интервале

$$0 \leq f_k \leq \frac{2u_m}{r}. \quad (7.64)$$

Кроме того, f_k может принимать только целочисленные значения

$$f_k = 0, 1, 2, \dots, \frac{2u_m}{r}. \quad (7.65)$$

Рассмотрим эту задачу при двух типах показателя качества.

1. Минимум квадратичной ошибки. Минимизировать

$$J = \sum_{k=1}^N \{[y(k) - y_d(k)]^T Q(k) [y(k) - y_d(k)] + \omega u^2(k)\}, \quad (7.66)$$

где $y_d(k)$ — требуемый вектор состояния, $Q(k)$ — $(n \times n)$ -мерная симметричная положительно определенная матрица, ω — весовой коэффициент.

2. Задача минимального времени. В этом случае требуется найти последовательность $u(1), \dots, u(N)$, которая переводит систему из состояния $y(0)$ в нулевое состояние $y(N)$ за наименьшее число периодов.

Ниже будут подробно рассмотрены эти две задачи.

Задача минимальной квадратичной ошибки

В этой задаче задано число управляющих периодов N . Чтобы привести эту задачу к задаче целочисленного программирования, будем считать управляющим входным сигналом переменную f_k , удовлетворяющую уравнению (7.62) и ограничению (7.64).

Подставляя (7.63) в (7.61), получим

$$y(k) = \Phi^k y(0) - \sum_{i=1}^k \Phi^{k-i} \mathbf{d} u_m + \sum_{i=1}^k \Phi^{k-i} \mathbf{d} r f_i. \quad (7.67)$$

Показатель качества можно выразить через f_1, f_2, \dots, f_N , если подставить (7.67) и (7.62) в (7.66):

$$J = z_0 + z_1 f_1 + z_2 f_2 + \dots + z_N f_N + \\ + (z_1 + z_{11} f_1 + z_{12} f_2 + \dots + z_{1N} f_N) f_1 + \\ + (z_N + z_{N1} f_1 + z_{N2} f_2 + \dots + z_{NN} f_N) f_N, \quad (7.68)$$

где $z_{k1} = z_{1k}$, так как матрицы $Q(k)$ — симметричны.

Так как ω неотрицательно, а матрицы $Q(k)$ положительно определенные, то и J положительно определено. Показатель качества можно записать в следующей матричной форме:

$$J = z_0 + 2\mathbf{Z}_1^T \mathbf{F} + \mathbf{F}^T \mathbf{Z}_2 \mathbf{F}, \quad (7.69)$$

где

$$\mathbf{F} = \begin{bmatrix} f_1 \\ \dots \\ f_N \end{bmatrix}, \quad \mathbf{Z}_1 = \begin{bmatrix} z_1 \\ \dots \\ z_N \end{bmatrix}, \quad \mathbf{Z}_2 = \begin{bmatrix} z_{11} & z_{12} & \dots & z_{1N} \\ \dots & \dots & \dots & \dots \\ z_{N1} & z_{N2} & \dots & z_{NN} \end{bmatrix}. \quad (7.70)$$

Теперь задача заключается в том, чтобы найти неотрицательную последовательность управляющих входных сигналов (f_1, f_2, \dots, f_N), подчиненных ограничению (7.65), которая минимизирует показатель качества (7.69).

Изложенный в этом параграфе метод использует алгоритм отыскания собственных векторов матрицы. Согласно этому методу вначале определяется оптимальное решение без каких-либо ограничений, а из этого решения без ограничений выводится целочисленное решение, удовлетворяющее ограничению (7.65).

Процедуру удобно описать геометрически и иллюстрировать ее двумерным построением, приведенным на рис. 7.10. Пусть \mathbf{F}_0 представляет собой оптимальное решение без ограничений. Ограничение (7.64) представлено N -мерным кубом с ребром, равным $2u_m/r$. В двух измерениях куб представлен квадратом $ABCD$ на рис. 7.10, а, б. Размерность куба определяется числом всех периодов дискретизации N .

\mathbf{F}_0 должно удовлетворять N линейным алгебраическим уравнениям

$$\frac{\partial J}{\partial f_i} = 0, \quad (7.71)$$

или $z_i + z_{i1}f_1 + \dots + z_{iN}f_N = 0$ при $i=1, 2, \dots, N$. На рис. 7.10, а F_0 лежит внутри ограничивающего куба, а на рис. 7.10, б F_0 лежит вне ограничивающего куба.

Пусть точки на рис. 7.10 соответствуют целочисленным значениям, удовлетворяющим (7.64) и (7.65). Рас-

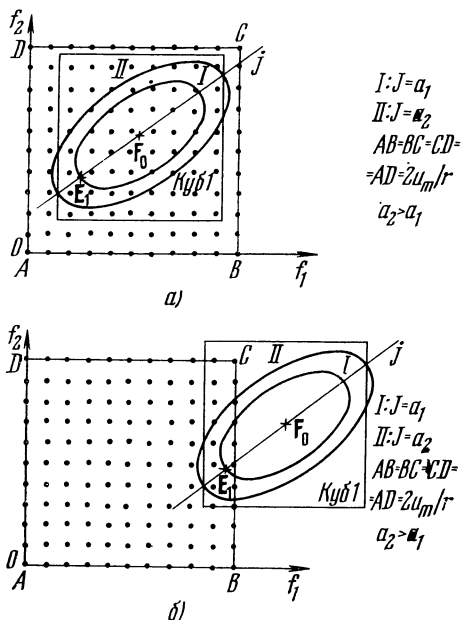


Рис. 7.10. Геометрическая иллюстрация алгоритма [12]: а) F_0 внутри ограничивающего куба, б) F_0 вне ограничивающего куба.

смотрим уравнения $J(f_1, \dots, f_N) = a_i$, где a_i — постоянные. Эти уравнения задают семейство эллипсоидов с центром в точке F_0 . Следовательно, задача сводится к отысканию наименьшего эллипсоида, проходящего через точку решетки.

Этот эллипсоид определяется следующим образом. Рассмотрим произвольный эллипсоид $J = a_1$, который обозначим — эллипсоид I.

Наиболее удаленная от F_0 точка на эллипсоиде есть точка E_1 , лежащая на его большой оси j (рис. 7.10). Направление прямой j совпадает с направлением соб-

ственного вектора, соответствующего наименьшему собственному значению матрицы Z_2 . Пусть Y_N обозначает этот собственный вектор. Пусть куб I , есть N -мерный куб, центр которого есть F_0 , а ребро есть расстояние между E_1 и F_0 . Рассмотрим точки решетки, лежащие внутри куба I и одновременно внутри ограничивающего куба. Пусть точка F_{s_1} является одной из тех точек, которой соответствует наименьший показатель качества J_{F_1} . Если F_{s_1} лежит внутри эллипсоида I или на нем, то $J_{F_1} \geq J_{E_1}$, где J_{E_1} показатель качества в точке E_1 . Тогда F_{s_1} является целочисленным оптимальным решением. Если $J_{F_1} > J_{E_1}$, то точка F_{s_1} лежит вне эллипсоида и не является решением. Увеличив эллипсоид, т. е. взяв большее значение a_2 и повторив изложенную выше процедуру, получим точку E_2 , куб 2 и точку F_{s_2} . Далее сравним соответствующие им показатели качества J_{F_2} и J_{E_2} . Если J_{F_2} все еще больше, чем J_{E_2} , повторяем процедуру, пока не будет найдена (после k повторений) точка решетки F_{s_k} , которой соответствует показатель качества $J_{F_k} \leq J_{E_k}$. Тогда точка F_{s_k} является решением.

Алгоритм представим в виде следующих шагов:

Шаг 1. Сформулировать, как указано было выше, задачу так, чтобы показатель качества был в виде (7.68).

Шаг 2. Найти F_0 — оптимальное решение без каких-либо ограничений, путем решения N линейных алгебраических уравнений (7.71). При малых N можно применить правило Крамера. При больших N полезен алгоритм Краута [20].

Шаг 3. Определить Y_N — собственный вектор, соответствующий наименьшему собственному значению матрицы Z_2 . При $N \leq 3$ применим прямой метод. Однако при больших N должен быть применен метод итераций [20]. При этом методе Y_N определяется как собственный вектор, соответствующий наибольшему собственному значению матрицы Z_2^{-1} . Сама матрица Z_2^{-1} может быть найдена с помощью алгоритма Краута.

Шаг 4. Пусть расстояние между F_0 и E_1 равно kd , где d — произвольное число. Начав с $k=1$, найти точку E_1 , для которой направление отрезка F_0E_1 совпадает с направлением Y_N , а расстояние F_0E_1 равно d . Вычислить показатель качества в точке E_1 , т. е. величину J_{E_1} .

Шаг 5. Определить координаты точек решетки, лежащих внутри куба I (сторона которого равна d) и внутри ограничивающего куба. Найти среди этих точек решетки точку F_{s1} , которой соответствует наименьший показатель качества J_{F1} . При малых N это можно получить непосредственными вычислениями и сравнениями или, при больших N , применяя динамическое программирование.

Шаг 6. Сравнить J_{F1} и J_{E1} . Если $J_{F1} \leq J_{E1}$, то F_{s1} есть оптимальное решение. Если $J_{F1} > J_{E1}$, повторить шаги 4, 5 и 6 при $k=2, 3, \dots$ до тех пор, пока не будет найдена точка F_{sk} , которой соответствует $J_{Fk} \leq J_{Ek}$. Точка F_{sk} теперь является оптимальным решением.

Иллюстрирующий пример. Передаточная функция рассматриваемого линейного объекта задана в виде

$$G(s) = \frac{1}{s(s+1)}. \quad (7.72)$$

Вектор состояния определен в виде

$$\mathbf{y}(t) = \begin{bmatrix} y(t) \\ \dot{y}(t) \end{bmatrix}. \quad (7.73)$$

Период дискретизации T равен 1 сек. Вычислены фундаментальная и входная матрицы:

$$\Phi(T) = \begin{bmatrix} 1 & 0,6321 \\ 0 & 0,3679 \end{bmatrix}$$

и

$$\mathbf{d} = \begin{bmatrix} 0,3679 \\ 0,6321 \end{bmatrix}. \quad (7.74)$$

Управляющая функция может принимать целочисленные значения и ограничена следующим образом:

$$u_k \text{ — целые числа, } -10 \leq u_k \leq 10.$$

Требуемое состояние $\mathbf{y}_d(k)$ является нулевым состоянием, а w равно нулю и

$$Q(k) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (7.75)$$

Начальное состояние равно

$$\mathbf{y}(0) = \begin{bmatrix} 5,0 \\ 0 \end{bmatrix} \quad (7.76)$$

Подставляя (7.75) в (7.66), получим показатель качества

$$J = \sum_{k=1}^N \mathbf{y}(k)^T \mathbf{y}(k),$$

или

$$J = \sum_{k=1}^N [y(k)^2 + \dot{y}(k)^2] \quad (7.77)$$

Вводим новые переменные

$$f_k = u_k + 10$$

или

$$u_k = f_k - 10, \quad k = 1, 2, \dots, N \quad (7.78)$$

Очевидно, что f_k может принимать только положительные целочисленные значения, а именно

$$f_k = 0, 1, 2, \dots, 19, 20. \quad (7.79)$$

Задача заключается в определении последовательности f_1, \dots, f_N , которая минимизирует показатель качества (7.77) при соблюдении ограничения (7.79).

С л у ч а й $N=2$

Подстановка (7.74) и (7.78) в (7.67) приводит к следующему:

$$\left. \begin{aligned} \mathbf{y}(1) &= \begin{bmatrix} 1,3212 + 0,3679f_1 \\ -6,3212 + 0,6321f_1 \end{bmatrix}, \\ \mathbf{y}(2) &= \begin{bmatrix} -6,3534 + 0,7675f_1 + 0,3679f_2 \\ -8,6466 + 0,2325f_1 + 0,6321f_2 \end{bmatrix}. \end{aligned} \right\} \quad (7.80)$$

Из (7.77) и (7.80) следует, что показатель качества равен

$$J = 156,8328 - 10,3964f_1 - 7,8030f_2 + (-10,3964 + 1,1780f_1 + 0,4293f_2)f_1 + \\ + (-7,8030 + 0,4293f_1 + 0,5349f_2)f_2. \quad (7.81)$$

Точка $\mathbf{F}_0 = (4,9600; 10,6065)$ представляет оптимальное решение без ограничений, которое удовлетворяет следующим двум уравнениям:

$$\left. \begin{aligned} 1,1780f_1 + 0,4293f_2 &= 10,3964, \\ 0,4293f_1 + 0,5349f_2 &= 7,8030. \end{aligned} \right\} \quad (7.82)$$

Собственные значения матрицы Z_2 определяются прямым методом, следующим образом. Определитель матрицы $Z_2 - \gamma I$ равен нулю, т. е.

$$\begin{vmatrix} 1,1790 - \gamma & 0,4293 \\ 0,4293 & 0,5349 - \gamma \end{vmatrix} = 0. \quad (7.83)$$

Решениями уравнения (7.83) будут $\gamma_1 = 1,3928$ и $\gamma_2 = 0,3201$. Пусть \mathbf{Y}_2 является собственным вектором, соответствующим γ_2 , который определяется из следующего уравнения:

$$Z_2 \mathbf{Y}_2 = \gamma_2 \mathbf{Y}_2. \quad (7.84)$$

Следовательно, собственный вектор равен

$$Y_2 = \begin{bmatrix} 1,0 \\ -1,9983 \end{bmatrix}.$$

Пусть e_1 и e_2 являются координатами точки E_1 , для которой расстояние PE_1 равно $k\sqrt{2}$ (в этом случае d равно $\sqrt{2}$). Начав с $k=1$, имеем расстояние, равное 2, и

$$E_1 = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \frac{F_0 + dY_2}{\|Y_2\|}, \quad (7.85)$$

где $\|Y_2\|$ — длина вектора Y_2 .

Таким образом, получаем точку E_1 (5,5928; 9,3418) и соответствующий ей показатель качества J_{E_1} , равный 23,1450. Для точек решетки, лежащих внутри квадрата 1, возможными значениями f_1 будут 4,0; 5,0 и 6,0, а для f_2 — 10,0; 11,0 и 12,0. Из этих точек, точка F_{S_1} (5,0; 11,0) имеет наименьший показатель качества, равный $J_{F_1} = 22,6031$. Точка F_{S_1} есть оптимальное целочисленное решение, так как $J_{F_1} < J_{E_1}$. Оптимальные управления равны $f_1=5,0$, $f_2=11,0$ или $u_1=5,0$, $u_2=1,0$ при минимальном показателе качества $J_2=22,6031$. Соответствующие векторы состояния равны

$$y(1) = \begin{bmatrix} 3,1606 \\ -3,1606 \end{bmatrix} \quad \text{и} \quad y(2) = \begin{bmatrix} 1,5306 \\ -0,5306 \end{bmatrix}. \quad (7.86)$$

Случай $N=5$.

Векторы состояния $y(1)$ и $y(2)$ определены в (7.80). Подставляя (7.74) и (7.78) в (7.67), получим

$$\begin{aligned} y(3) &= \begin{bmatrix} -15,4979 + 0,9145f_1 + 0,7675f_2 + 0,3679f_3 \\ -9,5021 + 0,0855f_1 + 0,2325f_2 + 0,6321f_3 \end{bmatrix}, \\ y(4) &= \begin{bmatrix} -25,1832 + 0,9685f_1 + 0,9145f_2 + 0,7675f_3 + 0,3679f_4 \\ -9,8168 + 0,0315f_1 + 0,0855f_2 + 0,2325f_3 + 0,6321f_4 \end{bmatrix}, \\ y(5) &= \begin{bmatrix} -35,0674 + 0,9884f_1 + 0,9685f_2 + 0,9145f_3 + 0,7675f_4 + 0,8679f_5 \\ -9,9326 + 0,0116f_1 + 0,0315f_2 + 0,0855f_3 + 0,2325f_4 + 0,6321f_5 \end{bmatrix}. \end{aligned} \quad (7.87)$$

Из уравнений (7.77), (7.80) и (7.87) получаем показатель качества:

$$\begin{aligned} J &= 2546,2471 - 84,8572f_1 - 80,0515f_2 - 66,2348f_3 - 44,6922f_4 - 19,1792f_5 + \\ &+ (-84,8572 + 3,9377f_1 + 2,9971f_2 + 2,0460f_3 + 1,1375f_4 + 0,3709f_5)f_1 + \\ &+ (-80,0515 + 2,9971f_1 + 2,9606f_2 + 2,0394f_3 + 1,4111f_4 + 0,3762f_5)f_2 + \\ &+ (-66,2348 + 2,0460f_1 + 2,0394f_2 + 2,0215f_3 + 1,1510f_4 + 0,3905f_5)f_3 + \\ &+ (-44,6922 + 1,1375f_1 + 1,1411f_2 + 1,1510f_3 + 1,1780f_4 + 0,4293f_5)f_4 + \\ &+ (-19,1792 + 0,3709f_1 - 0,3762f_2 + 0,3905f_3 + 0,4293f_4 + 0,5349f_5)f_5. \end{aligned} \quad (7.88)$$

Точка F_0 удовлетворяет следующим пяти уравнениям.

$$\frac{\partial J}{\partial f_i} = 0 \quad \text{при} \quad i = 1, \dots, 5 \quad (7.89)$$

7а Д. Табак, Б. Куо

При применении алгоритма Краута оптимальное решение без ограничений будет представлено точкой F_0 (4,7388; 10,1746; 10,0601; 10,0250; 10,0230). Обратная матрица для матрицы Z_2 , найденная с помощью алгоритма Краута, равна

$$Z_2^{-1} = \begin{bmatrix} 1,1075 & -1,1437 & 0,0251 & 0,0105 & 0,0096 \\ -1,1437 & 2,2896 & -1,1676 & 0,0204 & 0,0188 \\ 0,0251 & -1,1676 & 2,2941 & -1,1554 & 0,0563 \\ 0,0105 & 0,0204 & -1,1554 & 2,3301 & -1,0483 \\ 0,0096 & 0,0188 & 0,0563 & -1,0483 & 2,6498 \end{bmatrix}. \quad (7.90)$$

Собственный вектор Y_5 определяется итерационным методом [20]. Строится итерационная последовательность векторов V_n , W_n и скаляров g_n , причем

$$V_0 = W_0 = (1 \ 0 \ 0 \ 0)^T, \quad g_0 = 1.$$

n -я итерация связана со следующей итерацией равенством

$$W_{n+1} = Z_2^{-1} V_n = g_{n+1} V_{n+1}, \quad (7.91)$$

при этом первый элемент вектора V_{n+1} есть 1. Матричное умножение в (7.91) повторяется до тех пор, пока разность между данными и следующим коэффициентами g не начнет удовлетворять неравенству

$$\left| \frac{g_{n+1} - g_n}{g_n} \right| \leq 0,0001. \quad (7.92)$$

В данном случае это происходит после 32 итераций. Затем можно принять $Y_5 = V_{n+1}$. Для улучшения точности можно применить экстраполяцию Эйткена [20]. Пусть $v_{i,n-1}$, $v_{i,n}$ и $v_{i,n+1}$ являются i -ми элементами соответственно векторов V_{n-1} , V_n и V_{n+1} . Тогда i -й элемент вектора Y_5 приближенно равен

$$y_i = \frac{v_{i,n-1} \cdot v_{i,n+1} - v_{i,n}^2}{v_{i,n-1} - 2v_{i,n} + v_{i,n+1}}. \quad (7.93)$$

После использования (7.93) получаем собственный вектор

$$Y_5 = (1,0; -2,7107; 3,6311; -3,3858; 2,2701)^T. \quad (7.94)$$

Начинаем с $k=1$, когда расстояние $F_0 E_1$ равно 1,0, координаты точки E равны 4,9007; 9,7357; 10,6480; 9,4768 и 10,3905 и соответствующий показатель качества равен $J_{E1} = 23,2670$. Для точек решетки, находящихся внутри куба I и внутри ограничивающего куба, возможные значения f_1 равны 4,0; 5,0, а для f_2, f_3, f_4 и f_5 — 10,0; 11,0. Прямыми вычислениями и сравнениями находим, что точка F_{S1} (5,0; 10,0; 10,0; 10,0; 10,0) имеет наименьший показатель качества $J_{F1} = 23,1048$. Так как $J_{F1} < J_{E1}$, то оптимальное целочисленное решение будет $u_1 = -5,0$, $u_2 = u_3 = u_4 = u_5 = 0$ при минимальном показателе

качества $J_5=23,1048$. Соответствующие векторы состояния равны

$$\left. \begin{aligned} \mathbf{y}(1) &= \begin{bmatrix} 3,1606 \\ -3,1606 \end{bmatrix}, \quad \mathbf{y}(2) = \begin{bmatrix} 1,1627 \\ -1,1627 \end{bmatrix}, \quad \mathbf{y}(3) = \begin{bmatrix} 0,4277 \\ -0,4277 \end{bmatrix}, \\ \mathbf{y}(4) &= \begin{bmatrix} 0,1574 \\ -0,1574 \end{bmatrix}, \quad \mathbf{y}(5) = \begin{bmatrix} 0,0579 \\ -0,0579 \end{bmatrix}. \end{aligned} \right\} (7.95)$$

Задача минимального времени

В этой задаче подлежат определению наименьшее число периодов дискретизации N и соответствующая последовательность управлений, которая переводит объект в нулевое состояние. Чтобы привести задачу к задаче смешанно-целочисленного линейного программирования, рассматривается переменная управления $f(t)$, удовлетворяющая (7.63) и ограничению (7.64).

Вводя неотрицательные дополнительные переменные (q_1, q_2, \dots, q_n) в неравенство (7.64), получим

$$f_k + q_k = \frac{2u_m}{r} \quad \text{при } k = 1, 2, \dots, N. \quad (7.96)$$

Очевидно, что q_k являются неотрицательными целочисленными переменными. Подставляя (7.63) в (7.61), получим

$$\begin{aligned} \mathbf{y}(N) = \begin{bmatrix} y_1 \\ \dots \\ y_n \end{bmatrix} &= \Phi^N \mathbf{y}(0) - \sum_{m=1}^N \Phi^{N-m} \mathbf{d}u_m + \\ &+ \sum_{m=1}^N \Phi^{N-m} \mathbf{d}rf_m, \end{aligned} \quad (7.97)$$

где n — порядок системы. Если выполнить матричное умножение в (7.97), получим элементы вектора $\mathbf{y}(N)$:

$$\left. \begin{aligned} y_1 &= b_{1,0} + b_{1,1}f_1 + \dots + b_{1,N}f_N, \\ \dots &\dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \\ y_n &= b_{n,0} + b_{n,1}f_1 + \dots + b_{n,N}f_N. \end{aligned} \right\} \quad (7.98)$$

Требуется, чтобы элементы вектора $\mathbf{y}(N)$ удовлетворяли неравенству

$$|y_k| \leq y_0 \quad \text{при } k=1, 2, \dots, n, \quad (7.99)$$

где y_0 — заданное число. Введем дополнительные $7a^*$

переменные (g_1, g_2, \dots, g_n) , которые удовлетворяют условиям

$$g_k = y_k + y_0 \quad (7.100)$$

и

$$0 \leq g_k \leq 2y_0 \text{ при } k=1, 2, \dots, n. \quad (7.101)$$

При введении других неотрицательных переменных (p_1, p_2, \dots, p_n) неравенство (7.101) переписывается следующим образом:

$$g_k + p_k = 2y_0 \text{ при } k=1, 2, \dots, n. \quad (7.102)$$

Подставляя (7.98) и (7.100) в (7.102), получим

$$b_{k,1}f_1 + \dots + b_{k,N}f_N + g_k = -b_{k,0} - y_0 \quad \text{при } k=1, 2, \dots, n. \quad (7.103)$$

Так как все ограничения линейны относительно управлений f_k , то задачу можно рассматривать, как задачу линейного программирования. Чтобы получить целевую функцию, введем в (7.103) неотрицательные искусственные переменные (a_1, a_2, \dots, a_n) . Если $b_{k,0} + y_0 < 0$, перепишем (7.103) в виде

$$a_k + b_{k,1}f_1 + \dots + b_{k,N}f_N - g_k = -b_{k,0} - y_0 \quad \text{при } k=1, 2, \dots, n. \quad (7.104)$$

Если $b_{k,0} + y_0 > 0$, перепишем (7.103) в виде

$$a_k - b_{k,1}f_1 - \dots - b_{k,N}f_N + g_k = b_{k,0} + y_0 \quad \text{при } k=1, 2, \dots, n. \quad (7.105)$$

Рассмотрим функционал

$$Z = -a_1 - a_2 - \dots - a_n. \quad (7.106)$$

Так как к концу вычислений, при любом допустимом решении, искусственные переменные должны быть равны нулю, то Z можно рассматривать как целевую функцию, которая должна быть максимизирована. Это — смешанно-целочисленная задача, она сводится к отысканию наименьшего N и соответствующей неотрицательной целочисленной последовательности $(f_1, f_2, \dots, f_N; q_1, \dots, q_N)$ и неотрицательной последовательности $(g_1, g_2, \dots, g_n; p_1, p_2, \dots, p_n)$, которые максимизируют Z , делая его равным нулю, при соблюдении ограничений (7.96), (7.102), (7.104) и (7.105). Чтобы решить эту задачу, надо выполнить две фазы операций [21]. Первая фаза

заключается в решении нецелочисленной задачи линейного программирования для определения минимума N и последовательности управления, которая делает Z равным нулю, без ограничений целочисленности, накладываемых на f_h и q_h . Это эквивалентно отысканию базисного допустимого решения в линейном программировании (см. гл. 3).

Вторая фаза, основанная на решении первой фазы, заключается в отыскании последовательности целочисленных управлений, приводящих к $y(N)$, удовлетворяющему ограничению (7.99).

Так как N неизвестно, то должна быть использована итерационная процедура. Прежде всего, делается попытка найти базисное допустимое решение для $N=1$. Если при $N=1$ не существует никакого базисного допустимого решения, то делается подобная же попытка для $N=2$. Этот процесс продолжается до тех пор, пока не будет найдено наименьшее N , при котором оказываются выполненными ограничения (7.96), (7.102) и (7.103). Это означает, что первая фаза закончена. Используя полученное базисное допустимое решение, начинают вторую фазу.

Решение с помощью задачи квадратичной ошибки. Согласно этому методу задача минимального времени рассматривается и решается как задача квадратичной ошибки. В предыдущем параграфе указывалось, что состояние объекта $y(N)$ должно удовлетворять ограничению (7.99). Это эквивалентно следующему ограничению:

$$y(N)^T y(N) \leq \beta, \quad (7.107)$$

где β — заданное число. Определим новый показатель качества следующим образом:

$$J = y(N)^T y(N). \quad (7.108)$$

Тогда ограничение (7.107) получает вид

$$J \leq \beta. \quad (7.109)$$

Задача может быть решена следующим итерационным методом. Начиная с $N=1$, определяют оптимальное решение, подчиненное ограничению (7.64), которое минимизирует показатель качества (7.108). Если минимум показателя качества удовлетворяет (7.109), то задача

решена. Если нет, то повторяют процесс до тех пор, пока не будет найдено такое значение N , при котором соответствующий минимальный показатель качества удовлетворяет неравенству (7.109).

7.6. Заключительные замечания

Изучение некоторых случаев в этой главе показало, что для решения линейных задач оптимального управления с дискретным временем можно эффективно применять линейное и квадратичное программирование. Однако в некоторых случаях, особенно в системах большой размерности, мы получаем чрезмерно большие таблицы в алгоритмах линейного или квадратичного программирования. Это может привести к явлениям плохой обусловленности, когда делается попытка обращения матриц, близких к особым. Эти явления плохой обусловленности недавно исследовал Полак [22], который предложил некоторые алгоритмы, позволяющие избежать трудностей этого типа. Дальнейшие изыскания новых алгоритмов для широкого класса разнообразных задач оптимального управления представляет интересную задачу для будущих исследований.

Литература

1. B. C. Kuo, *Analysis and Synthesis of Sampled-Data Control System*, Prentice-Hall, Englewood Cliffs, N. J., 1963.
2. B. H. Whalen, *Linear Programming for Optimal Control*, Ph. D. Thesis, University of California, Berkeley, 1962.
3. L. A. Zadeh and B. H. Whalen, *On Optimal Control and Linear Programming*, IRE Trans. Automatic Control AC-7, pp. 45—46 (1962).
4. А. И. Пропой, Применение методов линейного программирования для синтеза импульсных автоматических систем, *Автоматика и телемеханика* 24, стр. 912—920 (1963).
5. K. A. Fegley, *Designing Sampled-Data Control Systems by Linear Programming*, IEEE Trans. Appl. Ind. 83, pp. 198—200 (1964).
6. H. C. Torg, *Optimization of Discrete Control Systems Through Linear Programming*, J. Franklin Inst. 277, pp. 28—44 (1964).
7. G. Porcelli and K. A. Fegley, *Linear Programming Design of Digitally Compensated Systems*, JACC, 1964, pp. 412—421, Stanford, Cal.
8. K. A. Fegley and M. I. Hsu, *Optimum Discrete Control by Linear Programming*, IEEE Trans. Automatic Control AC-10, pp. 114—115 (1965).

9. G. Porcelli and K. A. Fegley, Optimal Design of Digitally Compensated Systems by Quadratic Programming, *J. Franklin Inst.* **282**, pp. 303—317 (1966).
10. M. Kim, On Optimum Control of Discrete Systems: I. Theoretical Development, *ISA Trans.* **5**, pp. 93—98 (1966).
11. M. D. Cannon and J. H. Eaton, A New Algorithm for a Class of Quadratic Programming Problems with Application to Control, *J. SIAM Control* **4**, pp. 34—45 (1966).
12. M. Kim, and K. Djadjuri, Optimum Sampled Data Control Systems with Quantized Control Function, *ISA Trans.* **6**, pp. 65—73 (1967).
13. G. N. T. Lack and M. Enns, Optimal Control Trajectories with Minimax Objective Functions by Linear Programming, *IEEE Trans. Automatic Control* **AC-12**, pp. 749—752 (1967).
14. S. Blum and K. A. Fegley, A Quadratic Programming Solution of the Minimum Energy Control Problem, *IEEE Trans. Automatic Control* **AC-13**, pp. 206—207 (1968).
15. D. Tabak, Applications of Mathematical Programming in the Design of Optimal Control Systems, Ph. D. Thesis, University of Illinois, Urbana, 1967.
16. D. Tabak, Optimization of Nuclear Reactor Fuel Recycle via Linear and Quadratic Programming, *IEEE Trans. Nuclear Science* **NS-15**, pp. 60—64 (1968).
17. L. Cutler, Product Form Quadratic Programming Code RSQPF4, The RAND Corporation, Santa Monica, Cal., 1965.
18. D. Tabak and B. C. Kuo, Application of Mathematical Programming in the Design of Optimal Control System, *Intern. J. Control* **10**, pp. 545—552 (1969).
19. D. Tabak, A Direct and Nonlinear Programming Approach to the Optimal Nuclear Reactor Shutdown Control, *IEEE Trans. Nuclear Science* **NS-15**, pp. 57—59 (1968).
20. S. H. Crandall, *Engineering Analysis*, Mc Graw-Hill, New York, 1956, pp. 26—47 and 91—98.
21. R. E. Gomory, An Algorithm for Integer Solutions to Linear Programs, in *Recent Advances in Mathematical Programming* (R. L. Graves and P. Wolfe, eds.) McGraw-Hill, New York, 1963, 269—302.
22. E. Polak, On the Removal of Ill-Conditioning Effects in the Computation of Optimal Controls, *Automatica* **5**, pp. 607—614 (1969).
- 23*. Я. З. Цыпкин, Об оптимальных процессах в импульсных автоматических системах, Докл. АН СССР **134**, № 2 (1960).
- 24*. Я. З. Цыпкин, Оптимальные процессы в импульсных автоматических системах, Изв. АН СССР, Энергетика и автоматика, № 4, 1960.
- 25*. Е. Г. Гольштейн, Д. Б. Юдин, Методы расчета и синтеза импульсных автоматических систем, *Автоматика и телемеханика*, **24**, № 7, № 12 (1963).
- 26*. Б. Н. Пшеничный, Синтез линейных импульсных систем, *Автоматика и телемеханика*, № 5, 1966.

8.1. Введение

После опубликования принципа максимума для задач оптимального управления с непрерывным временем [1] появилось много работ, посвященных разработке варианта принципа максимума для систем с дискретным временем, т. е. так называемого дискретного (или цифрового) принципа максимума. Из этих работ сошлемся лишь на [2, часть III] и [3—9]. Почти во всех этих работах были получены необходимые условия оптимальности*). Однако во всех случаях возникает необходимость решения трудной двухточечной граничной задачи.

В настоящее время одним из наиболее эффективных путей получения численных результатов при расчете оптимальных систем управления с дискретным временем, и особенно в случае нелинейных систем, является применение математического программирования. В случае линейных дискретных систем малой размерности, как это показал Ту [10] для равномерной дискретизации, можно применить динамическое программирование. Субоптимальное решение для неравномерной дискретизации было предложено Брокстейном и Куо [11]. Хотя в нелинейном случае математическое программирование имеет свои вычислительные трудности, оно все же дает возможность относительно просто получать численные результаты. Аналогичные идеи были высказаны также Розеном [12].

Рассматриваемая динамическая система описывается следующими разностными уравнениями состояния:

$$\mathbf{y}(i+1) = \mathbf{f}[\mathbf{y}(i), \mathbf{u}(i), i], \quad i=0, 1, \dots, N-1, \quad (8.1)$$

где $\mathbf{y}(i)$ — n -мерный вектор состояния в дискретный момент $t=t_i$, $\mathbf{u}(i)$ — m -мерный вектор управления при $t=t_i$, \mathbf{f} — нелинейная n -мерная векторная функция, N — мак-

*) Точную формулировку дискретного принципа максимума и историю его доказательства можно найти в книге А. И. Пропося [28]. (Прим. ред.)

симальное число рассматриваемых интервалов времени, в общем случае не одинаковых. В каждый дискретный момент система подчинена дополнительной системе ограничений в форме равенств и неравенств:

$$h_j[\mathbf{y}(i), \mathbf{u}(i-1)] = 0, \quad j=1, \dots, p, \quad (8.2)$$

$$g_k[\mathbf{y}(i), \mathbf{u}(i-1)] \geq 0, \quad k=1, \dots, q, \quad (8.3)$$

где h_j и g_k в общем случае — нелинейные функции. Предполагается, что вектор начального состояния задан:

$$\mathbf{y}(0) = \mathbf{y}_0. \quad (8.4)$$

Общую задачу оптимального управления для рассматриваемой системы с дискретным временем можно сформулировать следующим образом.

Перевести систему, описанную уравнением (8.1), из начального состояния \mathbf{y}_0 в целевую область, определяемую неравенством

$$\mathbf{a}[\mathbf{y}(N)] \geq 0, \quad (8.5)$$

где \mathbf{a} — l -мерная, нелинейная, векторная функция, так, чтобы был минимизирован (или максимизирован) следующий показатель качества:

$$J = \sum_{i=1}^N F[\mathbf{y}(i), \mathbf{u}(i-1), i], \quad (8.6)$$

где F — нелинейная функция, при условии выполнения ограничений (8.2) и (8.3).

Очевидно, что задача оптимального управления для системы с дискретным временем при конечном числе N временных интервалов является конечно-мерной задачей математического программирования. По аналогии с формулировкой (4.26) приведенную выше задачу оптимального управления можно переформулировать следующим образом:

$$\min (\text{или } \max) \left\{ \sum_{i=1}^N F[\mathbf{y}(i), \mathbf{u}(i-1), i] \mid \mathbf{a}[\mathbf{y}(N)] \geq 0; \right. \\ \left. g_k[\mathbf{y}(i), \mathbf{u}(i-1)] \geq 0, \quad k=1, \dots, q, \quad h_j[\mathbf{y}(i), \mathbf{u}(i-1)] = 0, \right. \\ \left. j=1, \dots, p; \quad \mathbf{h}_{p+i+1} = \mathbf{y}(i+1) - \mathbf{f}[\mathbf{y}(i), \mathbf{u}(i), i] = 0, \right. \\ \left. i=0, 1, \dots, N-1 \right\}. \quad (8.7)$$

Так как функции f предполагаются нелинейными, то, очевидно, мы имеем задачу нелинейного программирования. В наиболее общем случае все остальные функции — F , a , g , h — являются нелинейными. Дискретное время может быть равномерным или неравномерным, интервалы могут быть известны или неизвестны.

В частном случае, когда F , $-a$, g выпуклы, а h линейны, к задаче математического программирования (8.7) можно применить теорему Куна — Таккера [13] (см. 2.6). Для согласования с первой формулировкой, данной в 2.6, вводится следующая система функций:

$$\begin{aligned} G_1 &\equiv -a_1, \\ &\cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \\ G_l &\equiv -a_l, \\ &\cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \\ G_{l+1} &\equiv -g_1, \\ &\cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \\ G_{l+q} &\equiv -g_q. \end{aligned}$$

Предполагается, что функции F , G_i ($i=1, \dots, l+q$) являются выпуклыми, а h_j ($j=1, \dots, p+nN$) линейны. Кроме того, для конкретности будем полагать, что мы имеем задачу минимизации:

$$\min \left\{ \sum_{i=1}^N F[y(i), u(i-1), i] \mid G_j[y(i), u(i-1)] \leq 0, \right. \\ \left. \begin{array}{l} j = 1, \dots, l+q; \quad h_k[y(i), u(i-1), i] = 0, \\ k = 1, \dots, p+nN \end{array} \right\}. \quad (8.8)$$

Применяя теорему Куна — Таккера, можно построить функцию, аналогичную функции Лагранжа (2.23):

$$\begin{aligned} H(y, u, \mu, \lambda) &= \sum_{i=1}^N F[y(i), u(i-1), i] + \\ &+ \sum_{j=1}^M \mu_j G_j[y(i), u(i-1)] + \sum_{k=1}^Q \lambda_k h_k[y(i), u(i-1), i], \\ & \quad i = 1, \dots, N, \quad (8.9) \end{aligned}$$

где $M=l+q$, $Q=p+nN$, $\mu_j \geq 0$ — множители Куна — Таккера, λ_k — множители Лагранжа.

Если обозначить оптимальные векторы состояния и управления соответственно через $y^0(i)$ и $u^0(i-1)$ ($i=$

$= 1, \dots, N)$, то, используя правую часть неравенства Куна — Таккера (2.24), получим

$$H[y^0, u^0, \mu^0, \lambda^0] \leq H[y, u, \mu^0, \lambda^0], \quad (8.10)$$

где $\mu^0 \geq 0$ и λ^0 являются оптимальными векторами соответственно коэффициентов Куна — Таккера μ_j и множителей Лагранжа λ_k .

Неравенство (8.10) можно рассматривать как некоторый вид формулировки принципа минимума задачи оптимального управления для системы с дискретным временем. Аналогичные результаты ранее опубликовали Розен [12], Пирсон и Сридхар [14] и Мангасарьян и Фромовиц [15]. Необходимые условия для общей задачи оптимизации получили Кэнон, Каллам и Полак [16]. В качестве частного случая этих условий, эти авторы получили теорему Куна — Таккера и дискретный принцип максимума. Эти результаты подтверждают высказанное в гл. 4 соображение о том, что задача оптимального управления является фактически задачей математического программирования. Возможно, что математическое программирование следует рассматривать как вычислительную реализацию необходимых условий оптимальности, вытекающих из принципа максимума [1].

8.2. Пример: нелинейная система второго порядка [17, 18]

В следующем примере рассматривается система второго порядка с одним управлением. Система описывается следующими разностными уравнениями состояния:

$$\begin{aligned} y_1(i+1) - y_1(i) - T_{i+1}[-y_1^2(i) + y_2(i) + u(i)] &= 0, \\ y_2(i+1) - y_2(i) - T_{i+1}y_1(i) &= 0, \quad i=0, 1, \dots, N-1, \end{aligned} \quad (8.11)$$

где $y_j(i)$ — j -я компонента вектора состояния в дискретный момент t_i , а $T_j = t_j - t_{j-1}$.

Начальное состояние системы

$$y(0) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Задача заключается в том, чтобы перейти из начального состояния в целевую область:

$$[y_1(N) - 10]^2 - y_2^2(N) - 1 \leq 0 \quad (8.12)$$

при минимуме показателя качества

$$J = \sum_{i=1}^N [y_1^2(i) + y_2^2(i) + 0,1u^2(i-1)]. \quad (8.13)$$

Переменная управления ограничена:

$$0 \leq u(i) \leq 1, \quad i=0, 1, \dots, N-1. \quad (8.14)$$

Переменные состояния должны быть неотрицательными:

$$y_j(i) \geq 0, \quad i=1, \dots, N; \quad j=1, 2. \quad (8.15)$$

Отметим, что целевая область представляет собой круг с радиусом 1 с центром в точке (10, 0). Вычисления произведены для двух случаев. В обоих случаях число периодов дискретизации взято равным $N=12$. Различие между двумя случаями заключается в том, что в первом случае общее время

$$T = \sum_{i=1}^N T_i$$

предполагается нефиксированным, а во втором T фиксировано и равно 10.

Для примера с нефиксированным временем получены следующие результаты:

i	T_i (сек)	t_i (сек)	$u(i-1)$	$y_1(i)$	$y_2(i)$
1	0,00291	0,00291	0,0553	0,00265	0,964
2	0,00196	0,00487	0,5000	0,00375	0,930
3	0,00176	0,00663	0,5000	0,00434	0,898
4	0,00169	0,00832	0,0466	0,00467	0,867
5	0,00164	0,00996	0,0435	0,00479	0,839
6	0,00162	0,01158	0,0419	0,00473	0,812
7	0,00159	0,01317	0,0406	0,0048	0,786
8	0,00156	0,01473	0,0395	0,00402	0,762
9	0,00151	0,01624	0,0383	0,00330	0,740
10	0,00144	0,01768	0,0370	1,00218	0,719
11	0,00130	0,01898	0,0356	0,00012	0,699
12	13,18224	13,20122	0,0339	9,26978	0,683

Вычисленный оптимальный показатель качества равен

$$J_{\min} = 93,86.$$

Зависимость от времени переменных y_1 , y_2 и u представлена на рис. 8.1.

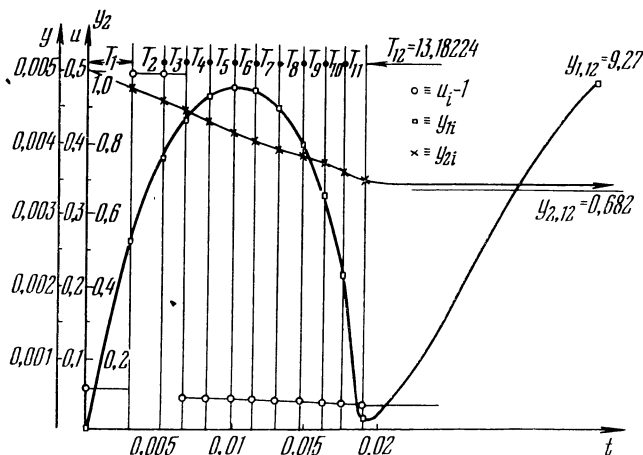


Рис. 8.1. Случай нефиксированного времени [18].

Для случая фиксированного времени получены следующие результаты:

i	T_i (сек)	t_i (сек)	$u(i-1)$	$y_1(i)$	$y_2(i)$
1	0,00146	0,00146	0,404	0,00205	0,973
2	0,07106	0,07252	0,500	0,14261	0,973
3	0,00042	0,07234	0,500	0,12606	0,925
4	0,00043	0,07337	0,992	0,10976	0,903
5	0,00044	0,07381	0,023	0,09369	0,884
6	0,00046	0,07427	0,023	0,07781	0,866
7	0,00047	0,07474	0,022	0,06208	0,850
8	0,00048	0,07522	0,022	0,04648	0,832
9	0,00048	0,07570	0,022	0,3099	0,822
10	0,00049	0,07619	0,022	0,01555	0,811
11	0,00050	0,07669	0,022	0,00012	0,801
12	0,92503	10,00172	0,022	0,36986	0,776

$$J_{\min} = 96,91.$$

Приведенные результаты показывают, что меньший показатель качества получается в случае нефиксированного времени. Результаты вычислений для второго примера представлены на рис. 8.2.

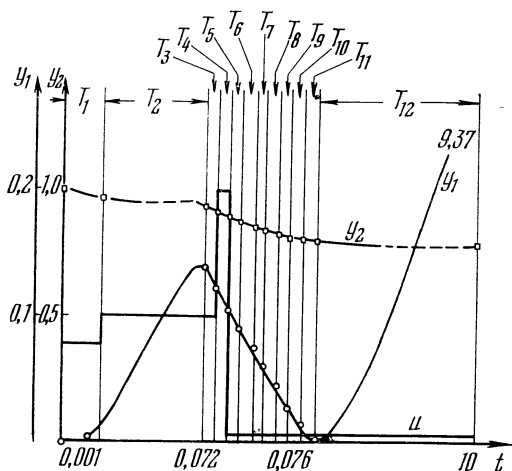


Рис. 8.2. Случай фиксированного времени [18].

Зависимости переменных от времени в этих двух случаях имеют аналогичный характер. Как видно из рис. 8.1 и 8.2, управление действует главным образом в течение первых трех или четырех интервалов. После этого подается очень небольшой сигнал управления. Переменная y_1 вначале растет, затем спадает до очень малого значения и, далее, в течение длительного интервала растет к целевому значению. Переменная y_2 медленно уменьшается в течение процесса до величины, составляющей 68—77% от своего начального значения.

В обоих случаях задача нелинейного программирования имеет 48 переменных и 37 ограничений. Время вычислений каждой задачи на ЦВМ ИБМ 7094 составляло приблизительно 8 мин. Для данного примера была использована программа метода последовательной минимизации без ограничений, разработанная Фиакко и МакКормиком [19].

8.3. Нелинейная система управления антенной [27]

Рассматривается часть системы управления антенной, предназначенной для сопровождения целей. Система состоит из линейной и нелинейной подсистем. Динамика системы может быть представлена следующим семейством разностных уравнений:

$$\mathbf{y}(i+1) = A\mathbf{y}(i) + \mathbf{b}NL[u(i)], \quad (8.16)$$

где A — $(n \times n)$ -мерная постоянная матрица, \mathbf{b} — n -мерный постоянный вектор, $u(i)$ — скалярная переменная управления при $t = t_i$, NL — нелинейная скалярная функция.

В рассматриваемой системе интервалы времени равны (равномерная дискретизация). Учитывая это, вектор состояния в любой момент времени $t = t_N$ можно выразить следующим образом [10]:

$$\mathbf{y}(N) = A^N \mathbf{y}(0) + \sum_{i=0}^{N-1} A^{N-i-1} \mathbf{b}NL[u(i)], \quad (8.17)$$

где $\mathbf{y}(0)$ — начальный вектор состояния, который предполагается известным.

Выходная переменная (положение антенны) системы представлена первой компонентой вектора состояния y_1 . Цель управления заключается в том, чтобы выходная переменная y_1 приняла заданное значение y_R . Этот процесс должен происходить при минимальной затрате энергии. В соответствии с этими требованиями сформулирован показатель качества:

$$\text{минимизировать } J = \sum_{i=1}^N \{ [y_1(i) - y_R]^2 + W u_1^2(i-1) \}, \quad (8.18)$$

где W — весовой коэффициент и N — максимальное число рассматриваемых дискретных моментов.

Первый член в выражении (8.18) для показателя качества определяется требованием минимизации ошибки, равной разности между фактическим и требуемым положением системы. Второй член выражает требование минимума энергии.

Из (8.17) следует, что

$$y_1(i) = A_1^i y(0) + \sum_{j=1}^{i-1} A_1^{i-j-1} \mathbf{b} NL[u(j)], \quad (8.19)$$

где A_1^i представляет собой первую строку матрицы A^i . Подставляя (8.19) в (8.18), получим

$$J = \sum_{i=1}^N \left\{ \left[A_1^i y(0) + \sum_{j=0}^{i-1} A_1^{i-j-1} \mathbf{b} NL[u(j)] - y_R \right]^2 + Wu^2(i-1) \right\}. \quad (8.20)$$

Ясно, что уравнение (8.20) определяет нелинейный показатель качества (нелинейную целевую функцию).

Следующие дополнительные ограничения в форме неравенства выражают ограничение амплитуды сигнала управления:

$$-u_m \leq u(i) \leq u_m. \quad (8.21)$$

Целевая функция (8.20), вместе с ограничениями (8.21), определяет задачу нелинейного программирования с N переменными и $2N$ ограничениями в форме неравенств.

Для решения этой задачи на машинной системе ИБМ 360/91 был использован алгоритм метода последовательной минимизации без ограничений.

Рассматриваемая реальная система моделировалась системой третьего порядка, для которой

$$A = \begin{bmatrix} -0,114 & 1,000 & 0 \\ 0,233 & -0,114 & 1 \\ -0,275 & 0,233 & 0 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0,179 \\ 0,550 \\ 0,115 \end{bmatrix}.$$

Рассматриваемая нелинейность, с явно выраженным насыщением, моделировалась гиперболическим тангенсом:

$$NL(u) = S \operatorname{th} \frac{u}{S}, \quad (8.22)$$

где S — сигнал на выходе нелинейности при насыщении, т. е. в нашем случае при $u \rightarrow \infty$. Часто насыщение моделируют резким изломом характеристики при переходе из

линейной области в насыщение. Во многих практических системах, как например в сервоусилителях, характеристика имеет плавный переход и функция (8.22) оказывается подходящей моделью. В нашем примере принято $S=10$.

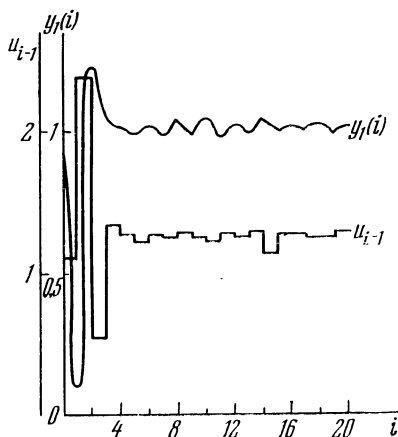


Рис. 8.3. Выходной и управляющий сигналы [27].

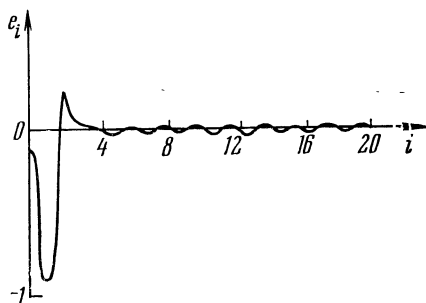


Рис. 8.4. Ошибка [27].

Максимально возможная амплитуда сигнала управления фиксирована на уровне $u_m=10$. Вектор начального состояния равен

$$\mathbf{y}(0) = \begin{bmatrix} 0,9 \\ 0 \\ 0 \end{bmatrix}.$$

Таблица 8.1 [27]

Дискретный момент i	Выход $y_i(t)$	Управляющий сигнал u_{i-1}	Ошибка $e_i = y_i(t) - y_R$
1	0,09494	1,1075	-0,905063
2	1,22355	2,3787	0,223548
3	1,04979	0,5408	0,049791
4	1,00955	1,3428	0,009553
5	0,97008	1,2791	-0,029922
6	1,01657	1,2225	0,016570
7	0,96834	1,2601	-0,031656
8	1,02376	1,2402	0,023765
9	0,97652	1,2814	-0,023479
10	1,03857	1,2390	0,038575
11	0,96528	1,2212	-0,034724
12	1,01545	1,2390	0,015446
13	0,97425	1,2528	-0,025753
14	1,02848	1,2791	0,028779
15	0,98529	1,1200	-0,014708
16	0,99706	1,2555	-0,002943
17	0,98556	1,2476	-0,014439
18	1,00052	1,2287	0,00521
19	0,97809	1,2304	-0,021913
20	0,99744	1,2589	-0,002564

Конечное значение $y_R=1,0$. Общее число периодов дискретизации было равно $N=20$. Результаты вычислений приведены в табл. 8.1 и на рис. 8.3 и 8.4.

Вычисления на вычислительной системе ИБМ 360/91 потребовали 12,12 сек. Величина показателя качества оказалась равной

$$J_0=0,8803.$$

8.4. Нелинейная система с цифровым управлением [20]

Рассматриваемая система представлена на схеме рис. 8.5. Регулятор системы является частью ЦВМ. Точнее говоря, регулятор реализован в виде подпрограммы ЦВМ, которая используется для управления. Это может быть специализированная ЦВМ, или пульт временного разделения, связанный с центральной вычислительной системой. Например, в рассматриваемом ниже случае системы цифрового управления антенной для сопровож-

дения целей применяется ЦВМ SDS Sigma 5, а регулятор осуществляется программой, записанной на языке ассемблера.

Рассматриваемая система (рис. 8.5) является нелинейной импульсной системой. В этом случае, вместо

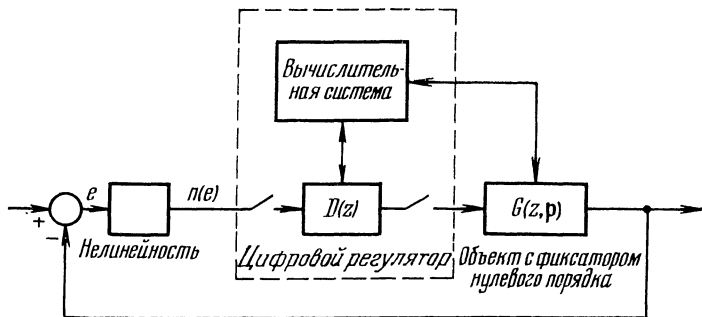


Рис. 8.5. Система с цифровым управлением [20].

критерия устойчивости, использованного в 6.4, должен быть применен критерий Джюри и Ли [21].

На нелинейность наложены следующие условия:

$$\left. \begin{aligned}
 1) \quad & n(e) \text{ — непрерывная функция,} \\
 2) \quad & n(0) = 0, \\
 3) \quad & K > \frac{n(e)}{e} > 0 \text{ при } e \neq 0, \\
 4) \quad & \left| \frac{dn(e)}{de} \right| < K'.
 \end{aligned} \right\} \quad (8.23)$$

Пусть $G^*(z)$ обозначает передаточную функцию всей линейной части системы, включая регулятор. Теорема Джюри и Ли утверждает, что система абсолютно устойчива, если существует такое q , что на единичной окружности $z = \exp(j\omega T)$, где T — период дискретизации, выполняется условие

$$\begin{aligned}
 JL(z) \equiv & \operatorname{Re} \{ G^*(z) [1 + q(z - 1)] \} + \\
 & + \frac{1}{K} - \frac{K' |q|}{2} |(z - 1) G^*(z)|^2 \geq 0. \quad (8.24)
 \end{aligned}$$

В этом случае алгоритм вычислений может быть построен в виде следующих двух шагов:

1. Установить при каком значении z левая часть неравенства (8.24) имеет минимальное значение.

2. Подставить полученное в шаге 1 значение z в неравенство (8.24) и использовать его в задаче нелинейного программирования, чтобы установить оптимальные параметры, обеспечивающие устойчивость системы.

Само собой разумеется, что должен быть сформулирован соответствующий показатель качества и учтены возможные дополнительные ограничения.

Этот алгоритм может применяться для различных цифровых систем управления. Например, его можно использовать для стабилизации и управления антенной, предназначенное для сопровождения целей. На рис. 8.6 показана блок-схема предлагаемого алгоритма.

В качестве иллюстрирующего примера применения алгоритма стабилизации выбран случай машинного управления 40-футовой антенной системой сопровождения целей. Структура системы схематически показана на рис. 8.7. Передаточная функция линейной части объекта, включая фиксатор нулевого порядка, равна

$$G(z) = \frac{0,179z^2 + 0,55z + 0,155}{z^3 - 0,114z^2 + 0,233z - 0,275}. \quad (8.25)$$

Предполагается, что нелинейность удовлетворяет условиям (8.23). Кроме этого, не делается никаких предположений относительно нелинейности, что делает решение применимым к широкому классу систем.

Цифровой регулятор взят второго порядка с передаточной функцией следующего вида:

$$D(z) = \frac{a_0z^2 + a_1z + a_2}{z^2 + b_1z + b_2} = \frac{a_0 + a_1z^{-1} + a_2z^{-2}}{1 + b_1z^{-1} + b_2z^{-2}}. \quad (8.26)$$

Параметры цифрового регулятора, a_0 , a_1 , a_2 , b_1 и b_2 , являются неизвестными и подлежат определению в процессе решения. Цифровой регулятор может быть запрограммирован для ЦВМ на основе хорошо разработанных методов [22]. Выбор регулятора второго порядка в некоторой степени случаен и, в какой-то мере, основан на опыте. Этот выбор, однако, не ограничивает общности, так как тот же метод может быть применен к $D(z)$ лю-

бого порядка. Единственное отличие будет заключаться в числе переменных. Само собой разумеется, что надо

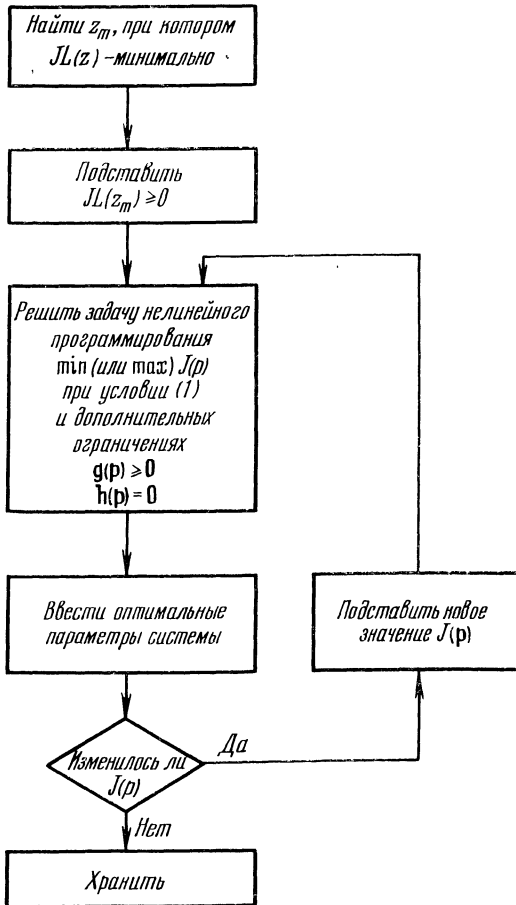


Рис. 8.6. Алгоритм для системы с цифровым управлением [20].

пробовать решать задачу при минимальном числе неизвестных, так как нежелательно иметь $D(z)$ излишне высокого порядка. Так как наш объект имеет передаточную функцию третьего порядка, то целесообразно

порядок регулятора выбирать на единицу меньше. Во всяком случае, выбор порядка $D(z)$ для нашего примера не играет существенной роли. Важным здесь является метод определения реальных значений параметров. Эти параметры фактически являются входными данными

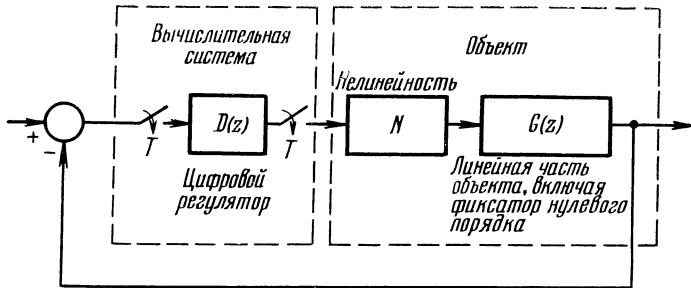


Рис. 8.7. Структура системы управления антенной [20].

подпрограммы, реализующей $D(z)$. Связь между регулятором и непрерывным объектом осуществляется через преобразователи аналог-цифра и цифра-аналог.

Передаточная функция всей линейной части системы, которая включает объект и регулятор, равна

$$\begin{aligned}
 G^*(z) &= D(z)G(z) = \\
 &= \frac{(a_0 + a_1z^{-1} + a_2z^{-2})(0,179z^{-1} + 0,55z^{-2} + 0,115z^{-3})}{(1 + b_1z^{-1} + b_2z^{-2})(1 - 0,114z^{-1} + 0,233z^{-2} + 1,275z^{-3})}.
 \end{aligned}
 \tag{8.27}$$

Как указывалось ранее, первая часть алгоритма относится к определению значения z , при котором минимальна левая часть неравенства (8.24). Другими словами, производится поиск значения z для худшего случая, при котором устойчивость системы является наиболее «угрожающей». Следует напомнить, что левая часть (8.24), $JL(z)$, должна быть неотрицательной, чтобы удовлетворять критерию устойчивости. В соответствии с этим критерием z должно лежать на единичной окружности в комплексной плоскости z , т. е.

$$z = e^{j\omega T}, \tag{8.28}$$

или

$$|z| = 1. \quad (8.29)$$

Одним из простейших путей определения значения z_m , минимизирующего $JL(z)$, является прямой поиск, поскольку имеется лишь одна независимая переменная z . Само собой разумеется, что должны быть соблюдены ограничения (8.28) и (8.29). Это можно сделать, выполняя поиск для вещественной части z , т. е. $\text{Re}(z)$, в качестве независимой переменной и вычисляя мнимую часть z по формуле

$$\text{Im}(z) = \sqrt{1 - \text{Re}^2(z)}. \quad (8.30)$$

Для данного примера поиск выполнялся прямым сканированием по значениям $\text{Re}(z)$ на фиксированном интервале. Само собой разумеется, что, в случае необходимости, можно воспользоваться более совершенными методами поиска. На вычислительной системе ИБМ 360/91 поиск занял 0,3 сек и дал следующие результаты:

$$\left. \begin{aligned} z_m &= -0,270 + j0,963, \\ JL(z_m) &= -7110,0. \end{aligned} \right\} \quad (8.31)$$

Та же задача была решена и как задача нелинейного программирования с использованием метода последовательной минимизации без ограничений [19]. Задача нелинейного программирования формулируется следующим образом:

$$\min \{JL(z, K, K', q) \mid |K| \leq K_{\max}; |K'| \leq K'_{\max}; |q| \leq q_{\max}; |z| = 1\}. \quad (8.32)$$

Заметим, что в этой задаче показатель качества (целевая функция) явно зависит от комплексной переменной z и от комплексной функции $G(z)$ (см. (8.24)). В этой задаче вместо $G^*(z)$ в (8.24) используется $G(z)$. Для числовых расчетов вместо z используются две переменные x_1 и x_2 , являющиеся соответственно вещественной и мнимой частями z , т. е.

$$z = x_1 + jx_2. \quad (8.33)$$

Комплексность z и соотношение $x = \text{Re}(z)$ обуславливается в программе на языке Фортран IV следующими

операторами:

COMPLEX Z
EQUIVALENCE ($Z, X(1)$).

Это автоматически влечет за собой то, что $X(2) = x_2$ будет мнимой частью z .

Условия (8.28) и (8.29) учитываются ограничением в форме равенства

$$x_1^2 + x_2^2 - 1 = 0. \quad (8.34)$$

В то время как в этом случае ограничения (8.32) и (8.34) очень просты, целевая функция $JL(x_1, x_2, K, K', q)$ представляет собой очень сложную функцию всех пяти переменных задачи. Хотя сама по себе целевая функция JL вещественна, она представляет собой явную зависимость от комплексных величин z и $G(z)$, что делает весьма трудным численное решение задачи нелинейного программирования. Для этого случая подходит специальный метод минимизации без ограничений, разработанный Фиакко и Мак-Кормиком [19]. Этот метод не требует вычислений производных используемых функций. Здесь это важно в силу сложности функции JL . Для данного случая получены следующие решения:

$$\left. \begin{aligned} z_m &= -0,288 + j0,958, \\ JL(z_m) &= -7101,0, \\ K &= 48,4, \\ K' = q &= 100,0. \end{aligned} \right\} \quad (8.35)$$

На той же вычислительной системе решение этого примера заняло около 28 сек. Может вызвать сомнения необходимость использования программы для вычислений методом последовательной минимизации без ограничений. Однако без соответствующих значений K , K' и q , полученных программой последовательной минимизации без ограничений, нельзя было бы проводить поиск. Сам поиск необходим, так как известно, что алгоритм, используемый в связи с методом последовательной минимизации без ограничений, теряет точность при сложных функциях. Все же, сравнение результатов, приведенных в (8.31) и в (8.35), показывает, что они не сильно отличаются друг от друга (0,13% для $JL(z_m)$). Очевидно,

следует выбрать результат поиска (8.31), так как он дает меньшее значение $JL(z_m)$. Поиск показал, что в этой области $JL(z)$ мало изменяется:

$$JL[\operatorname{Re}(z) = -0,30] = -7074,2,$$

$$JL[\operatorname{Re}(z) = -0,25] = -7078,5.$$

Поэтому точность второго значащего разряда в $\operatorname{Re}(z)$ не критична.

Значение z_m в (8.31) использовано в (8.24) и (8.27), а значение $G^*(z)$ в (8.27) подставлено в выражение (8.24) для JL . Теперь формулируется новая задача нелинейного программирования:

$$\min \{J(K, K') \mid JL(K, K', q, a_0, a_1, a_2, b_1, b_2) \geq 0; \mathbf{p} \in P\}, \quad (8.36)$$

где $\mathbf{p} = [K, K', q, a_0, a_1, a_2, b_1, b_2]$ — вектор параметров нашего примера. P является замкнутым множеством; в данном случае величины всех параметров ограничены:

$$|p_i| \leq 100,0, \quad i = 1, \dots, 8. \quad (8.37)$$

В этом случае показатель качества имеет вид

$$\max J = K + \omega K'. \quad (8.38)$$

где ω — весовой коэффициент, взятый в данном случае равным $\omega = 1$. Сущность задачи заключается в том, чтобы найти совокупность параметров \mathbf{p} , позволяющих работать системе при максимальных значениях K и K' , не теряя устойчивости.

Так как программа метода последовательной минимизации без ограничений приспособлена для решения задач минимизации, то формулировку показателя качества изменим следующим образом:

$$\min J(K, K') = -K - \omega K'. \quad (8.39)$$

Решение задачи на вычислительной системе ИБМ 360/91 заняло 5 сек и дало следующие результаты:

$$K = K' = 100,0 \text{ (работа на пределе),}$$

$$q = 15,0 \quad a_0 = 0,131,$$

$$a_1 = 0,063, \quad b_1 = 0,100,$$

$$a_2 = 0,129, \quad b_2 = 0,094.$$

Таким образом, цифровой регулятор должен обладать следующей передаточной функцией:

$$D(z) = \frac{0,131 + 0,063z^{-1} + 0,129z^{-2}}{1 + 0,100z^{-1} + 0,094z^{-2}}. \quad (8.40)$$

Фактическая величина показателя качества JL при указанных параметрах равна 0,08. Это достаточно близко к пределу устойчивости, равному нулю. Поэтому, для реальной работы должны быть взяты несколько меньшие значения K и K' . При решении задач, подобных этой, следует устанавливать предельные значения K и K' , превышающие процентов на 20 требуемую величину. Следует подчеркнуть, что поиск z_m производится редко и только при существенных изменениях параметров объекта. Так как для расчета параметров цифрового регулятора требуется мало машинного времени, то этот алгоритм определенно может применяться для управления процессами в натуральном времени, в тех случаях, когда изменения в требованиях к системе происходят не чаще, чем, скажем, один раз в 10 сек.

Таким образом, алгоритм может применяться в очень широком классе процессов, управляемых цифровыми машинами.

8.5. Расчет цифрового регулятора для нелинейной импульсной системы [17, 23]

В настоящем параграфе излагаются работы, в которых результаты исследований, описанных в 7.3, распространены на нелинейные импульсные системы. Речь идет о подходе, сущность которого сводится к составлению последовательности уравнений, устанавливающих связь между переменными системы для каждого момента дискретного времени. Эти уравнения обычно оказываются нелинейными не только вследствие наличия нелинейностей в системе, но и, как это будет далее показано, в результате наличия взаимных произведений переменных, соответствующих различным моментам дискретного времени. Критерий качества вводится в виде функции от ошибок системы при различных моментах дискретного времени.

Таким образом будет определена задача нелинейного программирования.

Структура рассматриваемой импульсной системы с равномерно дискретизированным временем показана на

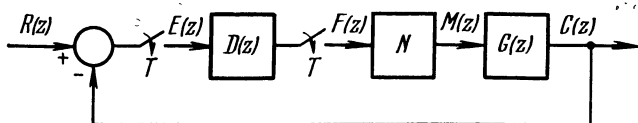


Рис. 8.8. Нелинейная импульсная система управления [23].

рис. 8.8. В этой системе $D(z)$ — импульсная передаточная функция объекта, $G(z)$ — импульсная передаточная функция объекта вместе с фиксатором (нулевого порядка), N — нелинейность.

Обычно фиксатор нулевого порядка стоит перед нелинейностью в прямой цепи. Тем не менее, поменяв местами фиксатор нулевого порядка и нелинейность N , с точки зрения математического описания не изменим системы, но значительно упростим расчетные уравнения.

Уравнения системы имеют следующий вид:

$$\left. \begin{aligned} E(z) &= R(z) - C(z), \\ C(z) &= G(z)M(z), \\ F(z) &= D(z)E(z), \\ M(z) &= N[F(z)], \end{aligned} \right\} \quad (8.41)$$

где N соотношение между сигналами на выходе и входе нелинейности.

Обычно $G(z)$ представляет собой отношение двух полиномов относительно z^{-1} :

$$G(z) = \frac{P(z)}{Q(z)}, \quad (8.42)$$

где

$$\left. \begin{aligned} P(z) &= p_k z^{-k} + p_{k+1} z^{-(k+1)} + \dots + p_m z^{-m}, \\ Q(z) &= q_0 + q_1 z^{-1} + \dots + q_l z^{-l} \end{aligned} \right\} \quad (8.43)$$

и k — задержка объекта, измеренная числом интервалов дискретизации [22].

Импульсная передаточная функция цифрового регулятора также представляет собой отношение полиномов относительно z^{-1} [22]:

$$D(z) = \frac{A(z)}{B(z)}, \quad (8.44)$$

где

$$\left. \begin{aligned} A(z) &= a_0 + a_1 z^{-1} + \dots + a_u z^{-u}, \\ B(z) &= 1 + b_1 z^{-1} + \dots + b_v z^{-v}. \end{aligned} \right\} \quad (8.45)$$

Подставляя (8.42) и (8.44) в (8.41), получим

$$R(z)Q(z) = E(z)Q(z) + P(z)M(z), \quad (8.46)$$

$$A(z)E(z) - B(z)F(z) = 0. \quad (8.47)$$

Допустим, что входной и другие сигналы в системе можно представить в виде следующих последовательностей:

$$\left. \begin{aligned} R(z) &= \sum_{i=0}^n r_i z^{-i}, \\ E(z) &= \sum_{i=0}^n \theta_i z^{-i}, \\ F(z) &= \sum_{i=0}^n f_i z^{-i}, \\ M(z) &= \sum_{i=0}^n m_i z^{-i}, \end{aligned} \right\} \quad (8.48)$$

где $r_i = [r(t)]_{t=iT}$, T — период дискретизации (аналогично для θ_i , f_i и m_i), а n — общее число рассматриваемых моментов дискретного времени.

Используя (8.42), (8.45) и (8.48) в уравнениях (8.46), (8.42) и в последнем уравнении из (8.41), раскрывая произведения и приравнивая коэффициенты при равных степенях z^{-1} , получим следующую систему

уравнений:

$$\left. \begin{aligned}
 r_0 q_0 &= e_0 q_0, \\
 r_1 q_0 + r_0 q_1 &= e_1 q_0 + e_0 q_1, \\
 r_k q_0 + \dots + r_0 q_k &= e_k q_0 + \dots + e_0 q_k + m_0 p_k, \\
 r_l q_0 + \dots + r_0 q_l &= e_l q_0 + \dots + e_0 q_l + \\
 &\quad + m_{l-k} p_k + \dots + m_0 p_l, \\
 r_m q_0 + \dots + r_{m-l} q_l &= e_m q_0 + \dots + e_{m-l} q_l + \\
 &\quad + m_{m-k} p_k + \dots + m_0 p_m, \\
 r_n q_0 + \dots + r_{n-l} q_l &= e_n q_0 + \dots + e_{n-l} q_l + \\
 &\quad + m_{n-k} p_k + \dots + m_{n-m} p_m.
 \end{aligned} \right\} (8.49)$$

Аналогично, подставляя (8.45) в (8.47), получим

$$\left. \begin{aligned}
 e_0 a_0 - f_0 &= 0, \\
 e_1 a_0 + e_0 a_1 - f_1 - f_0 b_1 &= 0, \\
 e_u a_0 + \dots + e_0 a_u - f_u - \dots - f_0 b_u &= 0, \\
 e_n a_0 + \dots + e_{n-u} a_n - f_n - \dots - f_{n-u} b_u &= 0,
 \end{aligned} \right\} (8.50)$$

где предполагается, что $u = v$,

$$\left. \begin{aligned}
 m_0 &= N(f_0), \\
 m_n &= N(f_n).
 \end{aligned} \right\} (8.51)$$

При раскрытии произведения в (8.49) для простоты предполагается, что $k \leq l \leq m$, что обычно выполняется в импульсной передаточной функции объекта. Заметим, что хотя уравнения (8.49) линейны относительно переменных e_i и m_i , уравнения (8.50) нелинейны из-за взаимных произведений, возникающих из (8.47). Таким образом, даже когда в рассматриваемой системе нет нелинейности, расчетные уравнения (8.49) и (8.50) при $m_i = f_i$ ($i = 0, 1, \dots, n$) все же оказываются нелинейными. Это является результатом только подхода, которому мы следуем, специально приспособленного к нелинейным системам. Расчет линейной системы при незначительно отличающемся подходе может быть осуществлен более эффективно с помощью линейного или квадратичного программирования (см. 7.3).

Системы уравнений от (8.49) до (8.51) образуют ограничения в форме равенств задачи нелинейного про-

граммирования. Чтобы учесть частные требования такие, например, как условия, ограничивающие по абсолютной величине дискретные ошибки, на переменные могут быть наложены дополнительные ограничения в форме неравенств.

Для определения задачи нелинейного программирования должен быть записан критерий оптимальности, основанный на выбранном показателе качества. В данном случае выбираем следующий критерий взвешенной, суммированной, среднеквадратичной ошибки:

$$\text{минимизировать } J = \sum_{i=0}^n c_i e_i^2, \quad (8.52)$$

где c_i — вес, отнесенный к i -й дискретной ошибке.

Тот же метод можно легко распространить на системы с несколькими нелинейностями, например на системы вида, показанного на рис. 8.9.

Аналогично уравнениям (8.41), уравнения такой системы имеют вид

$$\left. \begin{aligned} E(z) &= R(z) - C(z), \\ C(z) &= G_2(z) M_2(z), \\ M_2(z) &= N_2[F_2(z)], \\ F_2(z) &= G_1(z) M_1(z), \\ M_1(z) &= N_1[F_1(z)], \\ F_1(z) &= D(z) E(z). \end{aligned} \right\} \quad (8.53)$$

Действуя в том же порядке, как и в предыдущем случае, получим расчетные уравнения, совершенно аналогичные по структуре уравнениям (8.49) и (8.51).

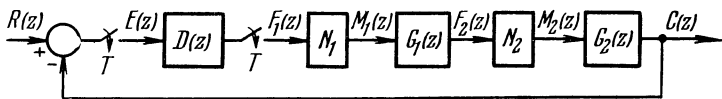


Рис. 8.9. Система с двумя нелинейностями [23].

Для иллюстрации предлагаемого расчетного метода рассмотрим систему с нелинейностью, имеющей насыщение. Как уже указывалось ранее, во многих практических случаях нелинейность описывается гладкой

кривой, а не линейно-ломаной. Поэтому нелинейность с насыщением моделируем гиперболическим тангенсом. Тогда (8.51) запишется в виде

$$m_i = S \operatorname{th} \frac{f_i}{S}, \quad i = 0, 1, \dots, n, \quad (8.54)$$

где S — значение m_i в области насыщения. При $f_i=0$ крутизна кривой равна 1.

Рассматриваемый объект обладает передаточной функцией

$$G(s) = \frac{0,0018(1 - e^{-Ts})}{s^2(1 + 200s)}, \quad (8.55)$$

которая включает фиксацию нулевого порядка, точное интегрирование и одну постоянную времени. При периоде дискретизации $T=50$ сек, z -преобразованная передаточная функция объекта равна

$$G(z) = \frac{0,014z^{-1}(1 + 0,93z^{-1})}{(1 - z^{-1})(1 - 0,78z^{-1})}. \quad (8.56)$$

Кроме того, были выбраны следующие дополнительные данные:

порог насыщения	$S=30$,
входной сигнал системы	
является ступенчатой функцией	$r(t) = 10u(t)$,
число рассматриваемых моментов	
дискретного времени	$n=20$,
вид цифрового регулятора	

$$D(z) = \frac{a_0 + a_1z^{-1} + a_2z^{-2}}{1 + b_1z^{-1} + b_2z^{-2}}. \quad (8.57)$$

Осуществляя соответствующие подстановки в уравнения от (8.49) до (8.51), получим следующие расчетные уравнения:

$$\left. \begin{aligned} e_0 &= 10, \\ e_1 &+ 0,0104m_0 = 10, \\ e_2 - 1,78e_1 &+ 0,0104m_1 + 0,00967m_0 = -7,8, \\ e_3 - 1,78e_2 + 0,78e_1 &+ 0,0104m_2 + 0,00967m_1 = 0, \\ &\dots\dots\dots \\ e_{20} - 1,78e_{19} + 0,78e_{18} &+ 0,0104m_{19} + 0,00967m_{18} = 0, \end{aligned} \right\} (8.58)$$

$$\left. \begin{aligned} 10a_0 & & -f_0 & & = 0, \\ (e_1 - 10b_1)a_0 + 10a_1 & & -f_1 & & = 0, \\ (e_2 - 10b_2)a_0 + e_1a_1 + 10a_2 - f_2 - f_1b_1 & & = 0, \\ e_3a_0 & + e_2a_1 + e_1a_2 - f_3 - f_2b_1 - f_1b_2 & = 0, \\ e_{20}a_0 & + e_{19}a_1 + e_{18}a_2 - f_{20} - f_{19}b_1 - f_{18}b_2 & = 0, \end{aligned} \right\} \quad (8.59)$$

$$m_i = 30 \operatorname{th} \frac{f_i}{30}, \quad i = 0, 1, \dots, 20. \quad (8.60)$$

Фактически используемый показатель качества равен:

$$\text{минимизировать } J = \sum_{i=1}^{20} 2^i e_i^2. \quad (8.61)$$

Вычисления для этого примера производились с помощью программы метода последовательной минимизации без ограничений [19], минимизировавшей (8.61), при условии соблюдения ограничений в форме равенств от (8.58) до (8.60). Для цифрового регулятора получено

$$D(z) = \frac{170,87 + 4,63z^{-1} + 5,26z^{-2}}{1 + 0,439z^{-1} + 0,131z^{-2}}.$$

i	e_i	f_i	m_i
0	10,000	1708,70	30,00
1	3,716	-69,18	-29,94
2	0,334	-66,16	-29,94
3	-0,106	42,43	26,68
4	0,013	-4,56	-4,52
5	0,021	1,06	1,05
6	0,017	4,10	4,13
7	0,013	1,32	1,32
8	$9,303 \times 10^{-3}$	1,17	1,17
9	$7,769 \times 10^{-3}$	1,19	1,19
10	$9,691 \times 10^{-3}$	1,43	1,43
11	$7,874 \times 10^{-3}$	1,16	1,16
12	$7,447 \times 10^{-3}$	1,22	1,22
13	$6,971 \times 10^{-3}$	1,25	1,25
14	$5,656 \times 10^{-3}$	1,11	1,11
15	$4,169 \times 10^{-3}$	1,06	1,06
16	$2,720 \times 10^{-3}$	1,00	1,00
17	$1,572 \times 10^{-3}$	0,92	0,92
18	$8,527 \times 10^{-4}$	0,90	0,90
19	$2,056 \times 10^{-4}$	0,87	0,87
20	$4,301 \times 10^{-4}$	0,80	0,80

Значения сигналов в системе в моменты времени $i=1, \dots, 20$ приведены в табл. на стр. 232. Оптимальный показатель качества оказался равным $J_0=31,34$.

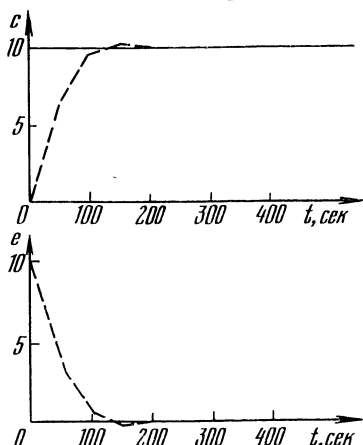


Рис. 8.10. Выход и ошибка в примере [23].

Зависимость от времени выходного сигнала и ошибки показана на рис. 8.10.

8.6. Итерационное решение

Решение задачи управления для одного класса нелинейных систем с дискретным временем было предложено Розеном [24]. Алгоритм состоит из последовательности решений задач математического программирования с линейными ограничениями. Розен доказал сходимость предложенной им итерационной процедуры [24].

Рассматривается следующая задача.

Дана нелинейная система с дискретным временем, разностные динамические уравнения состояния которой имеют следующий вид:

$$\mathbf{x}(i+1) - \mathbf{x}(i) = \mathbf{f}[\mathbf{x}(i), \mathbf{u}(i)]. \quad (8.62)$$

Найти оптимальную последовательность векторов управления $\mathbf{u}(i)$ такую, чтобы был минимизирован показатель качества:

$$J = \sum_{i=1}^{N-1} F[\mathbf{x}(i), \mathbf{u}(i)], \quad (8.63)$$

где $\mathbf{x}(i)$ — n -мерный вектор состояния при $t=t_i$, $\mathbf{u}(i)$ — r -мерный вектор управления при $t=t_i$, \mathbf{f} — n -мерная нелинейная функция, F — скалярная нелинейная функция.

Кроме того, предполагается, что на состояние и управление наложены следующие ограничения:

$$\mathbf{u}(i) \in U_i, \quad i=0, 1, \dots, N-1, \quad (8.64)$$

$$\mathbf{x}(i) \in X_i, \quad i=0, 1, \dots, N, \quad (8.65)$$

где U_i и X_i — компактные выпуклые множества. Далее, предполагается, что F — выпуклая функция, определенная на каждом из прямых произведений $U_i \times X_i$ со значениями в E^1 , и что \mathbf{f} есть функция, определенная на каждом $X_i \times U_i$ со значениями в E^n . Чтобы обеспечить сходимость предлагаемой итерационной процедуры, Розен предположил, что компоненты вектора \mathbf{f} должны быть либо выпуклыми, либо вогнутыми на $X_i \times U_i$ [24].

Вводится следующий новый вектор, он включает в себя все неизвестные переменные задачи:

$$\mathbf{z} = [\mathbf{x}(0), \mathbf{x}(1), \dots, \mathbf{x}(N), \mathbf{u}(0), \mathbf{u}(1), \dots, \mathbf{u}(N-1)]^T. \quad (8.66)$$

Очевидно, что \mathbf{z} представляет собой $[n(N+1) + rN]$ -мерный вектор. Пусть $M = [n(N+1) + rN]$.

Вводится новое ограничивающее множество Z :

$$\mathbf{z} \in Z = \prod_{i=1}^N X_i \times \prod_{i=0}^{N-1} U_i. \quad (8.67)$$

Так как множества X_i и U_i — выпуклые и компактные при всех i , то Z — также выпуклое и компактное. После подстановки компонент вектора \mathbf{z} в выражение (8.63) для показателя качества J будет представлено в виде функции от \mathbf{z} : $J(\mathbf{z})$. Ограничения в форме равенств, образованные разностными уравнениями (8.62), заменим следующими:

$$\begin{aligned} v_{i,j} = f_j[\mathbf{x}(i), \mathbf{u}(i)] + x_j(i) - x_j(i+1) = 0, \\ i=0, 1, \dots, N-1; j=1, \dots, n. \end{aligned} \quad (8.68)$$

Новые величины $v_{i,j}$ образуют вектор $\mathbf{v}(\mathbf{z})$, так что ограничения (8.68) можно выразить в виде

$$\mathbf{v}(\mathbf{z}) = \mathbf{0}. \quad (8.69)$$

Очевидно, что \mathbf{v} есть Nn -мерный вектор. Пусть $L = Nn$.

Задача оптимального управления, сформулированная в начале этого параграфа, теперь может быть сформулирована в виде следующей задачи математического программирования:

$$\min_z \{J(z) | z \in Z; \mathbf{v}(z) = 0\}. \quad (8.70)$$

Эта формулировка, по существу, совпадает с данной в (8.7). Отметим, что исходная задача точно описывается уравнением (8.70). В данном случае общим и наиболее очевидным подходом является прямое применение к задаче (8.70) какого-либо алгоритма нелинейного программирования. Однако при этом уверенность в том, что минимум является глобальным, может быть только тогда, когда допустимая область — выпуклая. Ранее было установлено, что Z выпуклое, так что каждая компонента векторной функции $\mathbf{v}(z)$ также выпуклая. Допустимая область, определенная в (8.70), будет выпуклой, только в случае, если $\mathbf{v}(z)$ линейна относительно z . Чтобы получить выпуклость, Розен линеаризирует уравнения состояния системы с помощью первых членов ряда Тейлора в окрестности произвольного z_1 :

$$\mathbf{w}(z, z_1) = \mathbf{v}(z_1) + \left\| \frac{\partial \mathbf{v}}{\partial z} \right\|_{z_1} (z - z_1), \quad (8.71)$$

где $\left\| \frac{\partial \mathbf{v}}{\partial z} \right\|_{z_1}$ — $(L \times M)$ -мерная матрица якобиана векторной функции $\mathbf{v}(z)$ при $z = z_1$.

Для реализации вычислений согласно предлагаемому итерационному алгоритму определим следующие множества:

$$S = \{z \in Z, \mathbf{v}(z) \geq 0\}, \quad (8.72)$$

$$W(z_1) = \{z | \mathbf{w}(z, z_1) \geq 0\}. \quad (8.73)$$

Теперь определим следующее точно-множественное отображение

$$\Gamma: Z \rightarrow Z,$$

как пересечение $W(z_1)$ и Z :

$$\Gamma z_1 = W(z_1) \cap Z. \quad (8.74)$$

Розен показал, что множество Γz_1 компактное и выпуклое [24].

Итерационная процедура начинается с произвольной начальной точки $\mathbf{z}_0 \in S$ и решается следующая задача математического программирования:

$$J(\mathbf{z}^1) = \min_{\mathbf{z} \in \Gamma \mathbf{z}^0} J(\mathbf{z}). \quad (8.75)$$

Предполагается, что J содержит штрафной член, который определяет возмещение за выход из допустимой области. Эта итерационная процедура затем продолжает образование последовательности субоптимальных точек $\mathbf{z}^1, \mathbf{z}^2, \dots$ согласно соотношению

$$J(\mathbf{z}^{j+1}) = \min_{\mathbf{z} \in \Gamma \mathbf{z}^j} J(\mathbf{z}), \quad j = 0, 1, \dots \quad (8.76)$$

Выпуклость $J(\mathbf{z})$ и $\Gamma \mathbf{z}^j$ гарантирует, что глобальный минимум $J(\mathbf{z})$ для $\mathbf{z} \in \Gamma \mathbf{z}^j$ достигается при $\mathbf{z} = \mathbf{z}^{j+1}$. Розен доказал сходимость этой процедуры [24]. Метод был далее развит и обобщен Майером [25, 26].

Литература

1. Л. С. Понтрягин, В. Г. Болтянский, Р. В. Гамкрелидзе, Е. Ф. Мищенко, Математическая теория оптимальных процессов, Физматгиз, 1961.
2. Л. И. Розоноэр, Принцип максимума Л. С. Понтрягина в теории оптимальных систем, Автоматика и телемеханика **20**, часть I, 1288—1302; часть II, 1402—1421; часть III, 1517—1532 (1959).
3. S. S. L. Chang, Digitized Maximum Principle, IRE Proc. **48**, pp. 2030—2031 (1960).
4. S. Katz, A Discrete Version of Pontryagin's Maximum Principle, J. Electronics Control **13**, pp. 179—184 (1962).
5. L. T. Fan and C. S. Wang, The Discrete Maximum Principle, Wiley, New York, 1964. [Русский перевод: Фан Л.-Т., Вань Ч.-С., Дискретный принцип максимума, «Мир», 1967.]
6. H. Halkin, B. W. Jordan, E. Polak and J. B. Rosen, Theory of Optimum Discrete Time Systems, 3rd IFAC Conference, London, 1966.
7. J. D. Pearson, The Discrete Maximum Principle, Intern. J. Control **2**, pp. 117—124 (1965).
8. J. M. Holtzman, Convexity and the Maximum Principle for Discrete Systems, IEEE Trans. Automatic Control **AC-11**, pp. 30—35 (1966).
9. А. И. Пропой, Об одной задаче оптимального дискретного управления, Докл. АН СССР **159**, № 6 (1964).
10. J. T. Hou, Optimum Design of Digital Control Systems, Academic Press, New York, 1963. [Русский перевод: Ю. Ту, Цифровые и импульсные системы автоматического управления, «Машиностроение», 1964.]

11. A. J. Brockstein and B. C. Kuo, Optimum Control of Multi-variable Sampled Data Systems with Adaptive Sampling, 1967, JACC, pp. 366—372, Philadelphia, Pa.
12. J. B. Rosen, Optimal Control and Convex Programming, MRC Technical Report 547, University of Wisconsin, Madison, Feb. 1965.
13. H. W. Kuhn and A. W. Tucker, Nonlinear Programming, in Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, Berkeley, 1951, pp. 481—492.
14. J. B. Pearson and R. Sridhar, A Discrete Optimal Control Problem, IEEE Trans. Automatic Control **AC-11**, pp. 171—174 (1966).
15. O. L. Mangasarian and S. Fromovitz, A Maximum Principle in Mathematical Programming, in Mathematical Theory of Control (A. V. Balakrishnan and L. W. Neustadt, eds.), Academic Press, New York, 1967, pp. 85—95.
16. M. D. Canon, C. D. Cullum and E. Polak, Constrained Minimization Problems in Finite Dimensional Spaces, J. SIAM Control **4**, pp. 528—547 (1966).
17. D. Tabak, Application of Mathematical Programming in the Design of Optimal Control Systems, Ph. D. Thesis, University of Illinois, Urbana, 1967.
18. D. Tabak and B. C. Kuo, Application of Mathematical Programming in the Design of Optimal Control Systems, Intern. J. Control **10**, pp. 545—552 (1969).
19. A. V. Fiacco and G. P. McCormick, Nonlinear Programming: Sequential Unconstrained Minimization Techniques, Wiley, New York, 1968. [Русский перевод: А. Фиакко и Г. Мак-Кормик, Нелинейное программирование, Методы последовательной безусловной минимизации, «Мир», 1972.]
20. D. Tabak, An Algorithm for Nonlinear Process Stabilization and Control, IEEE Trans. Computers **C-19**, pp. 487—492 (1970).
21. E. I. Jury and B. W. Lee, On the Stability of a Certain Class of Nonlinear Sampled-Data Systems, IEEE Trans. Automatic Control **AC-9**, pp. 51—61 (Jan. 1964).
22. B. C. Kuo, Analysis and Synthesis of Sampled Data Control Systems, Prentice-Hall, Englewood Cliffs, N. J., 1963.
23. G. Porcelli, D. Tabak and K. A. Fegley, Design of Digital Controllers for Nonlinear Systems by Mathematical Programming, 1968 WESCON, Session 14, Los Angeles, Cal., Aug. 1968.
24. J. B. Rosen, Iterative Solution of Nonlinear Optimal Control Problems, J. SIAM Control **4**, pp. 223—244 (1966).
25. R. R. Meyer, The Solution of Non-Convex Optimization Problems by Iterative Convex Programming, Ph. D. Thesis, University of Wisconsin, Madison, 1968.
26. R. R. Meyer, The Validity of a Family of Optimization Methods, J. SIAM Control **8**, pp. 41—54 (1970).
27. D. Tabak, Optimal Control of Nonlinear Discrete-Time Systems by Mathematical Programming, J. Franklin Inst. **289**, pp. 111—119 (1970).
- 28*. А. И. Пропой, Элементы теории оптимальных дискретных процессов, «Наука», 1973.

9.1. Введение

Применению методов математического программирования к стохастическим задачам посвящено относительно мало работ. Однако имеется большая литература по стохастическим системам оптимального управления; здесь будет упомянута лишь незначительная ее часть [1—7].

В настоящей главе рассматриваются несколько попыток применения методов математического программирования к стохастическим системам оптимального управления. В 9.2 излагается метод совместного использования алгоритмов математического программирования и метода Монте-Карло. В 9.3 обсуждается расчет цифрового регулятора для импульсной системы с зашумленным входом. В 9.4 в общих чертах излагается метод применения математического программирования к задаче оптимальной оценки. Конечно, можно утверждать, что уже существуют очень эффективные методы решения задач оптимальной оценки [1—7]. Однако это справедливо для задач оценки без ограничений. В тех случаях, когда на переменные состояния и управления наложены ограничения в форме неравенств, попытки применения обычных методов сталкиваются с очень трудной вычислительной задачей [1—7]. В то же время, применение математического программирования значительно облегчает численное решение задач оптимальной оценки с ограничениями. В заключительном параграфе 9.5 описывается применение математического программирования к задаче идентификации импульсных систем.

9.2. Комбинированный метод Монте-Карло и математического программирования

Типичная стохастическая система оптимального управления схематически может быть представлена в виде, показанном на рис. 9.1 [1].

Рассматриваемая система является импульсной и может быть описана с помощью z -преобразования. Объект подчиняется следующей системе нелинейных разностных уравнений:

$$x_{i+1} = F_i(x_i, u_i, c_i), \quad i=0, 1, \dots, N-1, \quad (9.1)$$

где x_i — выходной сигнал объекта в i -й момент, u_i — управляющий входной сигнал объекта в i -й момент,

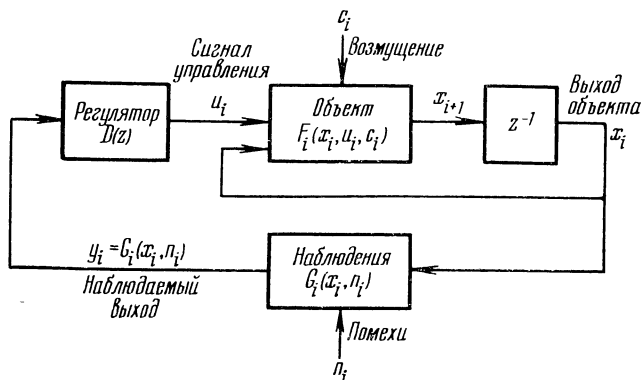


Рис. 9.1. Стохастическая система управления [1, 9].

c_i — параметр возмущения на входе объекта в i -й момент, F_i — нелинейная функция.

Для простоты рассматривается одномерная система. Однако x_i , u_i и c_i , вообще говоря, могут быть векторными переменными. Фактически наблюдаемый выходной сигнал y_i может быть представлен с помощью другой, в общем случае, нелинейной функции G_i :

$$y = G_i(x_i, n_i), \quad i=1, \dots, N, \quad (9.2)$$

где n_i — помеха в момент t_i .

Наблюдаемый сигнал y_i поступает в регулятор, который и должен быть рассчитан. Применяя z -преобразование, можно записать передаточную функцию регулятора в следующем общем виде:

$$D(z) = \frac{a_0 + a_1 z^{-1} + \dots + a_p z^{-p}}{1 + b_1 z^{-1} + \dots + b_p z^{-p}} = \frac{U(z)}{Y(z)}, \quad (3.9)$$

где p — порядок передаточной функции регулятора, а a_0, \dots, a_p и b_1, \dots, b_p — параметры регулятора, оценки которых должны быть найдены. Знание этих параметров позволит физически осуществить регулятор [10].

Из (9.3) следует, что сигнал управления в i -й момент можно выразить в виде функции текущего значения y_i и значений переменных y и u для p прошедших интервалов дискретизации [10].

Таким образом,

$$u_i = a_0 y_i + a_1 y_{i-1} + \dots + a_p y_{i-p} - b_1 u_{i-1} - \dots - b_p u_{i-p}. \quad (9.4)$$

Естественно предположить, что

$$y_i = u_i = 0 \quad \text{при } i < p.$$

Исходя из физических соображений и ограничений, относящихся к возможному сигналу на входе объекта, можно наложить на управления u_i следующие ограничения:

$$|u_i| \leq u_{\max}, \quad i = 0, 1, \dots, N-1, \quad (9.5)$$

Само собой разумеется, что при расчете системы надо применить определенный критерий качества. Например, можно следующим образом сформулировать задачу расчета.

Найти такие параметры $a_0 \dots a_p$ и $b_1 \dots b_p$ цифрового регулятора, при которых показатель качества

$$J = \sum_{i=1}^N W_i(x_i, u_{i-1}) \quad (9.6)$$

минимален и выполняются ограничения (9.1), (9.2), (9.4) и (9.5).

Из этих четырех уравнений видно, что они составляют задачу нелинейного программирования (гл. 2). Однако здесь есть существенное отличие. В то время как уравнения в гл. 2 содержали детерминированные переменные и ограничения, в уравнения данного параграфа дополнительно входят множества случайных переменных $c_i (i=0, 1, \dots, N-1)$ и $n_i (i=1, \dots, N)$.

Чтобы применить алгоритм нелинейного программирования, надо в (9.1) и (9.2) ввести вместо c_i и n_i некоторые определенные величины.

Если статистики и границы величин c_i и n_i известны, то можно применить генератор псевдослучайных чисел, вырабатывающий последовательности определенных величин c_0, c_1, \dots, c_{N-1} и n_1, n_2, \dots, n_N . Эти последовательности можно подставить в (9.1) и (9.2) и затем решить задачу нелинейного программирования. Естественно, что было бы желательно использовать наибольшее число (N) возможных сигналов.

Для такого рода задач можно применить метод Монте-Карло. После решения первой задачи нелинейного программирования вырабатываются две следующие последовательности из N псевдослучайных чисел c_i и n_i и решается вторая задача нелинейного программирования. Эта процедура повторяется M раз. Наконец, в качестве окончательного решения принимаются средние значения параметров регулятора a_i и b_i . Если не располагать очень быстро действующими средствами вычислений для задачи нелинейного программирования, то эта процедура может потребовать очень много времени. Однако при современных мультипроцессорных вычислительных машинах имеется возможность решать одновременно несколько задач нелинейного программирования, что облегчает применение предложенного метода. В тех случаях, когда статистики c_i и n_i неизвестны, значительно труднее образовать последовательности значений c_i и n_i . Необходимо использовать любую информацию, которую только можно получить о физической природе этих возмущений, и пытаться построить их модель, как можно более близкую к реальности. В этом случае вследствие большой неопределенности последовательностей c_i, n_i потребовалось бы значительно большее число экспериментов.

9.3. Статистический расчет импульсных систем управления

В этом параграфе рассматривается линейная импульсная система с зашумленным входным сигналом. Структура системы показана на рис. 9.2. Пусть основной входной сигнал есть $s(t)$, а сигнал помехи — $n(t)$.

Фактически сигнал на входе системы равен

$$r(t) = s(t) + n(t). \quad (9.7)$$

Выходной сигнал системы обозначим $c(t)$. Цель управления заключается в том, чтобы выходной сигнал $c(t)$ следил за изменениями основного входного сигнала $s(t)$.

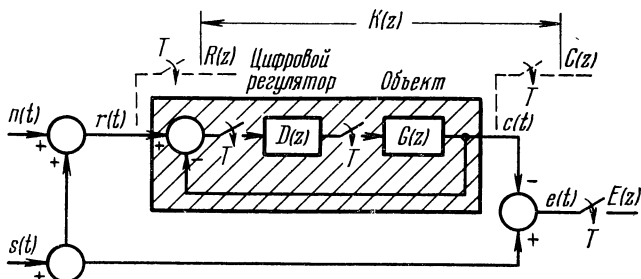


Рис. 9.2. Схема импульсной системы управления [11]; заштрихована рассчитываемая часть.

Наличие помехи затрудняет выполнение этой функции. Цель настоящего расчета заключается в построении такого цифрового регулятора $D(z)$, при котором ошибка

$$e(t) = s(t) - c(t) \quad (9.8)$$

была бы возможно малой. В качестве показателя качества в дальнейшем используется квадратичная ошибка.

Передаточная функция замкнутой системы управления равна

$$K(z) = \frac{D(z)G(z)}{1 + D(z)G(z)} = \sum_{i=1}^n K_i z^{-i}. \quad (9.9)$$

Здесь использованы те же обозначения, что и в гл. 7. Если $K(z)$ известно, то, полагая известной передаточную функцию объекта $G(z)$, можно найти передаточную функцию регулятора, равную

$$D(z) = \frac{K(z)}{G(z)[1 - K(z)]}. \quad (9.10)$$

В тех случаях, когда $G(z)$ имеет нули вне единичного круга на z -плоскости, этот подход может оказаться непригодным. Практически очень трудно достигнуть точной компенсации этих нулей, и так как они оказываются

полюсами функции $D(z)$, то могут сделать систему неустойчивой. Поэтому следует проявить осторожность при применении на практике излагаемого в данном параграфе подхода, как и любого другого подхода, основанного на уравнении (9.10) (см. гл. 7).

В дальнейшем предполагаются известными корреляционные функции сигнала и помехи. Среднеквадратичная ошибка, соответствующая случайному входному сигналу, может быть представлена в виде контурного интеграла в комплексной плоскости z для систем, запоминающих информацию неограниченной давности, или в виде квадратичной функции величин K_i для систем, запоминающих информацию конечной давности [12]. Показатель качества имеет следующий вид:

$$\text{минимизировать } J = e^2(nT) = \mathbf{KAK}^T, \quad (9.11)$$

где

$$\mathbf{K} = \begin{bmatrix} 1 \\ K_1 \\ K_2 \\ \dots \\ K_n \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} a_{00} & a_{10} & a_{20} & a_{30} & \dots & a_{n0} \\ a_{10} & a_{11} & a_{21} & a_{31} & \dots & a_{n1} \\ a_{20} & a_{21} & a_{11} & a_{21} & \dots & a_{(n-1)1} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n0} & a_{n1} & a_{(n-1)1} & a_{(n-2)1} & \dots & a_{11} \end{bmatrix} \quad (9.12)$$

\mathbf{A} представляет собой симметричную матрицу, элементы которой зависят от дискретов корреляционных функций $R_{ss}(\tau)$ и $R_{ns}(\tau)$ основного сигнала $s(t)$ и помехи $n(t)$:

$$\left. \begin{aligned} a_{00} &= R_{ss}(0), \\ a_{10} &= -R_{ss}(iT) - R_{ns}(iT), \quad i=1, 2, \dots, n, \\ a_{11} &= R_{ss}(0) + R_{nn}(0), \\ a_{ii} &= -a_{(i-1), 0}, \quad i=1, 2, \dots, n. \end{aligned} \right\} \quad (9.13)$$

Коэффициенты K_i являются переменными задачи. Обычно они имеют как положительные, так и отрицательные значения. Так как обычные алгоритмы линейного и квадратичного программирования требуют, чтобы переменные были неотрицательными, то производится следующее преобразование переменных:

$$K_i = K_{ia} - K_{ib}, \quad (9.14)$$

где K_{ia} и K_{ib} — неотрицательные величины.

После подстановки в (9.11) получим

$$\overline{e^2(nT)} = \hat{K} \hat{A} \hat{K}^T, \quad (9.15)$$

где

$$\hat{A} = \begin{array}{c} \overline{\left[\begin{array}{ccc|ccc} a_{n0} & a_{10} & \dots & a_{n0} & -a_{10} & \dots & -a_{n0} \\ a_{10} & & & & & & \\ \vdots & & & A_{00} & & & -A_{00} \\ a_{n0} & & & & & & \\ \hline -a_{10} & & & & & & \\ \vdots & & & -A_{00} & & & \\ -a_{n0} & & & & & & A_{00} \end{array} \right]} \end{array}, \quad (9.16)$$

$$\hat{K} = [1, K_{1a}, \dots, K_{na}, K_{1b}, K_{2b}, \dots, K_{nb}], \quad (9.17)$$

A_{00} — минор, образованный вычеркиванием первой строки и первого столбца матрицы A в (9.12).

Для статистического расчета импульсной системы управления минимизируемая среднеквадратичная ошибка $\overline{e^2(nT)}$ (9.15) является подходящей целевой функцией.

В общем случае ограничения системы определяются условиями физической осуществимости и требованиями, предъявляемыми к реакции системы на импульсный ступенчатый и линейно-изменяющийся входные сигналы. При квадратичном программировании допустимы только линейные ограничивающие уравнения. Когда объект обладает задержкой в m периодов дискретизации, то для физической осуществимости необходимо, чтобы общая импульсная передаточная функция $K(z)$ имела, по меньшей мере, ту же задержку. В общем случае заданная часть системы может быть представлена отношением двух полиномов:

$$G(z) = \frac{z^{-m} (h_0 + h_1 z^{-1} + \dots + h_b z^{-b})}{1 + g_1 z^{-1} + \dots + g_a z^{-a}}, \quad a \geq b. \quad (9.18)$$

Так как множитель z^{-m} в (9.16) выражает задержку в m периодов дискретизации, то для физической осуществимости необходимо, чтобы

$$K_i = 0, \quad i = 1, 2, \dots, m-1. \quad (9.19)$$

Пусть $K(z)$ представлено в виде ряда (9.9). Величины K_i представляют собой значения реакции замкнутой системы при $t=iT$ на единичный импульс при $i=0$. Если реакцию на единичный импульс надо ограничить

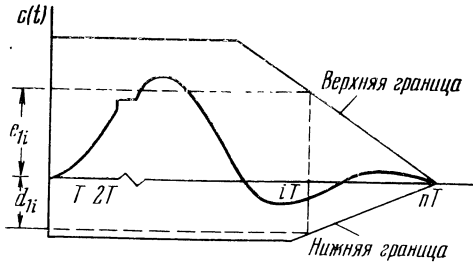


Рис. 9.3. Ограничения импульсной реакции системы [11].

пределами, показанными на рис. 9.3, то ограничивающие неравенства можно записать в следующем виде:

$$\left. \begin{aligned} K_i &\leq e_{1i}, & i = 1, 2, \dots, n-1, \\ K_i &\geq d_{1i}, & i = 1, 2, \dots, n-1, \\ K_i &= 0, & i = n, n+1, \dots \end{aligned} \right\} \quad (9.20)$$

Следовательно, если требуется, чтобы реакция на импульс прекратилась в n -й момент отсчета, то матрица A , определяемая уравнением (9.12), должна быть только $(n \times n)$ -мерной.

Если $K(z)$ является конечным полиномом, то разность $1-K(z)$ должна содержать множитель $(1-z^{-1})$, для того чтобы получить нулевую стационарную ошибку реакции на единичный ступенчатый входной сигнал.

Подставляя $z=1$ в уравнение (9.9), где верхний предел теперь равен n , получим ограничение в форме следующего равенства:

$$\sum_{i=1}^n K_i = 0, \quad K_i = 0, \quad i > n. \quad (9.21)$$

В этом случае A должно быть только $[(n+1)(n+1)]$ -мерной матрицей. Реакция на ступенчатый сигнал при

$t=iT$ для $i < n$ может быть ограничена неравенствами

$$\left. \begin{aligned} K_1 + K_2 + \dots + K_i &\leq (1 + e_{si}), \quad i = 1, 2, \dots, n-1, \\ K_1 + K_2 + \dots + K_i &\geq (1 - d_{si}), \quad i = 1, 2, \dots, n-1, \end{aligned} \right\} \quad (9.22)$$

где, как показано на рис. 9.4, e_{si} и d_{si} определяют диапазон значений, которые может иметь реакция на ступенчатый сигнал в i -й момент.

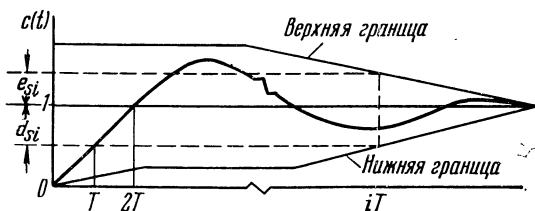


Рис. 9.4. Ограничения реакции на ступенчатый сигнал в системе [11].

Если система должна в заданных пределах следовать за линейно-изменяющимся входным сигналом, то ограничения будут определяться следующими неравенствами:

$$\begin{aligned} (i-1)K_1 + (i-2)K_2 + \dots + K_{(i-1)} &\leq i + e_{Ri}, \\ i &= 2, 3, \dots, n-1, \end{aligned} \quad (9.23)$$

$$\begin{aligned} (i-1)K_1 + (i-2)K_2 + \dots + K_{(i-1)} &\geq i - d_{Ri}, \\ i &= 2, 3, \dots, n-1, \end{aligned} \quad (9.24)$$

$$\begin{aligned} (i-1)K_1 + (i-2)K_2 + \dots + K_{(i-1)} &= i, \\ i &= n, n+1, \dots \end{aligned} \quad (9.25)$$

Уравнение (9.25) обеспечивает равную нулю стационарную ошибку и может быть удовлетворено, если потребовать, чтобы как разность $1 - K(z)$, так и ее производная содержали множитель $(1 - z^{-1})$. Тогда при $z=1$ $K(z)=1$ и $K'(z)=0$. Используя (9.9), можно записать ограничения в форме следующих равенств:

$$\sum_{i=1}^n K_i = 1, \quad \sum_{i=1}^n iK_i = 0, \quad (9.26)$$

$$K_i = 0, \quad i > n. \quad (9.27)$$

Уравнение (9.27) показывает, что размерность матрицы A должна быть не больше $[(n+1)(n+1)]$, а уравнения (9.23), (9.24) и (9.26) теперь образуют систему из $2n-2$ ограничений.

Заметим, что в тех случаях, когда объект нелинейно-фазовый или неустойчив, желательно, чтобы $K(z)$ содержало все нули функции $G(z)$, лежащие вне единичного круга, или $1-K(z)$ содержало все полюсы, лежащие вне единичного круга. Запишем $G(z)$ в следующем виде:

$$G(z) = \frac{z^{-m}(1-az^{-1})(h_0+h_1z^{-1}+\dots+h_s z^{-s})}{(1-bz^{-1})(1+g_1z^{-1}+\dots+g_r z^{-r})}, \quad r \geq s, \quad (9.28)$$

где a и b определяют положение нуля и полюса вне единичного круга. Тогда, полагая $K(z)=0$ при $z=a$, а $1-K(z)=0$ при $z=b$, получим соответствующие ограничивающие уравнения

$$\sum_{i=1}^n a^{n-i} K_i = 0, \quad \sum_{i=1}^n b^{n-i} K_i = b^n. \quad (9.29)$$

Следует также заметить, что, ограничивая число n коэффициентов в выражении $K(z)$, мы ограничиваем размерность матрицы A и создаем тенденцию к ограничению числа членов в передаточной функции $D(z)$ регулятора.

Для иллюстрации применения квадратичного программирования к статистическому расчету систем управления приведем теперь два примера.

Пример 1. Задача (рис. 9.2) заключается в расчете цифрового регулятора $D(z)$, который минимизирует средне-квадратичную ошибку $e^2(nT)$, если

$$G(s) = \frac{1 + e^{-Ts}}{s^2(s+1)}, \quad T = 0,2,$$

$$\Phi_{nn}(\omega) = 0,1, \quad \Phi_{ss}(\omega) = \frac{1}{1+\omega^2}, \quad \Phi_{ns}(\omega) = \Phi_{sn}(\omega) = 0.$$

Решение, полученное аналитически, было опубликовано Ту (Tou) [13]. Он получил следующие результаты:

$$K(z) = \frac{0,556}{(z-0,261)} = 0,556z^{-1} + 0,145z^{-2} + 0,0379z^{-3} + 0,0099z^{-4} + \dots,$$

$$D(z) = \frac{29,95(z^2 - 1,819z + 0,819)}{(z^2 - 0,08z - 0,73)}.$$

В общем случае, при подготовке задачи к решению с помощью квадратичного программирования, надо образовать целевую функцию и записать линейные ограничивающие уравнения. В данном примере нет линейных ограничивающих уравнений, а целевая функция дана уравнением (9.15). Таким образом, целью является

$$\text{минимизация } J = \hat{K} \hat{A} \hat{K}^T,$$

где \hat{K} должно быть найдено с помощью вычислений на ЦВМ, а \hat{A} дано уравнением (9.16). Элементы \hat{A} определяются путем разложения $\Phi_{ss}(z)$ и $\Phi_{nn}(z)$ в степенные ряды относительно z^{-1} .

Согласно таблице преобразований

$$\Phi_{ss}(z) = \frac{-0,2013z}{(z-0,8019)(z-1,221)} = \frac{1}{2} \left(\sum_{i=1}^{-\infty} e^{iT} z^{-i} + 1 + \sum_{i=1}^{\infty} e^{-iT} z^{-i} \right),$$

$$\Phi_{nn}(z) = 0,1.$$

Используя (9.13) для оценки элементов (61×61) -мерной матрицы \hat{A} , получаем

	0	1	2	30	31...60
0	0,500000	-0,409366	-0,335160	...	0,409366
1	-0,405366	0,600000	0,409366	...	
2	-0,335160	0,409366	0,600000	...	
3	-0,274406	0,335160	0,409366	...	
.	.	0,274406	0,335160	...	
.	.	.	0,274406	...	
.	
30	-0,001239	.	.	0,600000	
31	0,409366				
.	.				
.	.				
.	.				
60	0,001239				

Таблица 9.1

Граничные значения для реакции на ступенчатый сигнал в примере 2

Дискретный момент	ε_{st}	δ_{st}	Дискретный момент	ε_{st}	δ_{st}
T	0,40	0,70	$5T$	0,10	0,08
$2T$	0,50	0,45	$6T$	0,05	0,05
$3T$	0,30	0,30	$7T$	0,01	0,01
$4T$	0,20	0,16			

Пример был решен на ЦВМ общего назначения ИБМ 7040. Решение заняло 2,28 мин и дало следующие результаты:

$$\begin{aligned} K_1 &= 0,5548, & K_4 &= 0,0102, & K_7 &= 0,0002, \\ K_2 &= 0,1464, & K_5 &= 0,0027, & K_8 &= 0,001, \\ K_3 &= 0,0386, & K_6 &= 0,0007, & K_9 &= K_{10} = \dots = K_{30} = 0, \\ \overline{e^2(nT)} &= 0,2103. \end{aligned}$$

Эти значения очень близки к полученным аналитическим путем в виде ряда. Отметим, что большая часть элементов вектора \hat{K} равна нулю и что ряд, соответствующий $K(z)$, быстро сходится.

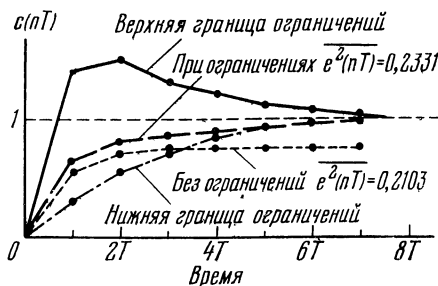


Рис. 9.5. Реакция систем, рассчитанных в примерах 1 и 2, на единственный ступенчатый входной сигнал [11].

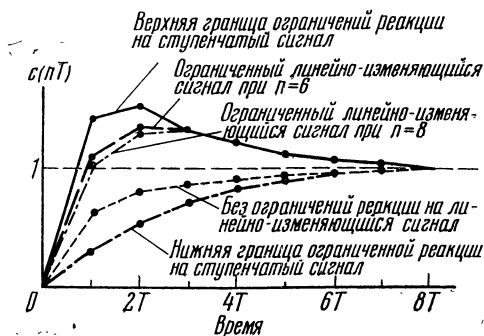


Рис. 9.6. Реакция систем, рассчитанных в примере 2, с ограничениями реакции на линейно-изменяющийся сигнал и без них [11].

Пример 2. Для задачи, рассмотренной в примере 1, потребуем теперь, чтобы реакция на ступенчатый сигнал была ограничена величинами, приведенными в табл. 9.1, и чтобы стационарная ошибка стала равной нулю на восьмом моменте отсчета. В примере 1 \hat{A} было взято в виде (61×61) -мерной матрицы. В данном примере \hat{A} опреде-

ляется ограничением, заключающемся в том, что стационарная ошибка при ступенчатом входе становится равной нулю на восьмом отсчете. В этом примере наложено 14 ограничений вида (9.22), обусловленных граничными значениями первых семи отсчетов реакции на ступенчатый сигнал, и дополнительное ограничение вида (9.21), обусловленное требованием нулевой стационарной ошибки на ступенчатый сигнал.

На рис. 9.5 показаны результаты вычислений, выполненных на ЦВМ. Заметим, что среднеквадратичная ошибка отсчетов $\overline{e^2(nT)}$, которая в примере 1 была равна 0,2103, вследствие ограничений возросла до 0,2331.

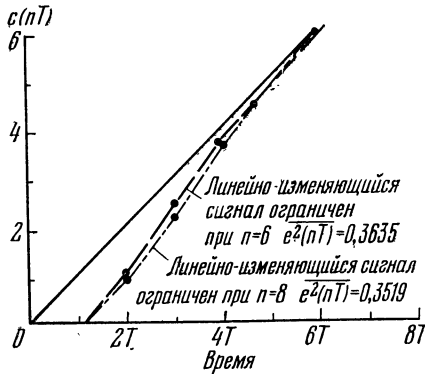


Рис. 9.7. Реакция систем, рассчитанных в примере 2, на единичный линейно-изменяющийся сигнал [11].

Дополнительно к приведенным выше ограничениям добавлены уравнения (9.26) и (9.27) для обеспечения нулевой стационарной ошибки реакции на линейно-изменяющийся сигнал. Результаты решения на ЦВМ при $n=6$ и $n=8$ приведены на рис. 9.6 и 9.7.

Эти исследования были развиты Маровицем [14], который применил методы математического программирования к задачам расчета оптимальных фильтров с ограничениями.

9.4. Применение к задачам оптимальной оценки [15]

Оптимальная оценка является, по существу, задачей оптимизации, которая обычно формулируется в виде минимизации среднеквадратичной ошибки. Как и любая другая задача оптимизации, естественно, она может быть сформулирована в виде задачи математического программирования. Такую формулировку предложили Айзекс и Лохри [15] для задачи рекуррентной оптималь-

ной оценки при ограничениях состояния в форме неравенств. Задача нелинейного программирования формулируется для каждого шага итерационного алгоритма.

Рассматриваемая система описывается следующими уравнениями динамического состояния:

$$\dot{\mathbf{y}}(t) = A(t)\mathbf{y}(t), \quad (9.30)$$

где $\mathbf{y}(t)$ — n -мерный вектор состояния и A — $(n \times n)$ -мерная зависящая от времени матрица.

Переходное уравнение состояния между двумя соседними моментами t_j, t_{j+1} ($t_j < t_{j+1}$) можно записать следующим образом:

$$\mathbf{y}(j+1) = \Phi(j+1, j)\mathbf{y}(j), \quad (9.31)$$

где $\mathbf{y}(j) = \mathbf{y}(t_j)$, а $\Phi(j+1, j) = \Phi(t_{j+1}, t_j)$ — переходная матрица, полученная в результате интегрирования следующего дифференциального уравнения:

$$\dot{\Phi}(t, j) = A(t)\Phi(t, j), \quad (9.32)$$

от $t = t_j$ до $t = t_{j+1}$ при начальном условии $\Phi(j, j) = I$.

Переменные состояния в каждый дискретный момент подчинены ограничениям в форме неравенств

$$\mathbf{g}[\mathbf{y}(j)] \leq \mathbf{b}, \quad (9.33)$$

где \mathbf{g} — выпуклая векторная функция, \mathbf{b} — постоянный вектор. Тогда неравенство (9.33) определяет замкнутое выпуклое множество в n -мерном пространстве.

Уравнение измерений имеет следующий вид:

$$\mathbf{z}(j) = H(j)\mathbf{y}(j) + \mathbf{v}(j), \quad (9.34)$$

где $\mathbf{z}(j)$ — m -мерный вектор измерений при $t = t_j$, $H(j)$ — $(m \times n)$ -мерная матрица измерений при $t = t_j$, $\mathbf{v}(j)$ — m -мерный вектор измерений шума. Предполагается, что $\mathbf{v}(j)$ — белый шум с нулевым средним значением.

На каждом шаге, т. е. при $t = t_j$, получают оценку вектора состояния $\hat{\mathbf{y}}(j)$. Задача заключается в том, чтобы, используя измерения $\mathbf{z}(j)$, обновлять оценку состояния $\hat{\mathbf{y}}(j)$ так, чтобы минимизировались квадратичные ошибки наблюдений при соблюдении ограничений (9.33).

Состояние $\mathbf{y}(j+1)$ оценивается рекуррентно, а именно: если дана оценка $\hat{\mathbf{y}}(j)$, основанная на $\mathbf{z}(1), \dots, \mathbf{z}(j)$, то новая оценка $\hat{\mathbf{y}}(j+1)$ вычисляется только на основе $\mathbf{z}(j+1)$ и $\hat{\mathbf{y}}(j)$. Чтобы это выполнить, надо определить

$\hat{y}(j+1)$, используя всю информацию $z(1), \dots, z(j+1)$. Математическое ожидание измерения при $t=t_j$ равно

$$\hat{z}(i) = H(i) \Phi(i, j+1) \hat{y}(j+1). \quad (9.35)$$

Введем следующие обозначения:

$$M_j = \begin{bmatrix} H(1) \Phi(1, j) \\ \vdots \\ H(j) \Phi(j, j) \end{bmatrix}, \quad \hat{z}_j^* = \begin{bmatrix} \hat{z}(1) \\ \vdots \\ \hat{z}(j) \end{bmatrix}, \quad z_j^* = \begin{bmatrix} z(1) \\ \vdots \\ z(j) \end{bmatrix}, \quad (9.36)$$

где все $z(i)$ ($i=1, \dots, j$) обозначают векторы фактических измерений.

Рассматривая все измерения до $t=t_{j+1}$, имеем

$$\begin{bmatrix} M_j \Phi^{-1}(j+1, j) \\ H(j+1) \end{bmatrix} \hat{y}(j+1) = \begin{bmatrix} \hat{z}_j^* \\ \hat{z}(j+1) \end{bmatrix}. \quad (9.37)$$

Уравнение (9.37) не удовлетворяется точно, так как измерения искажены шумом v . Поэтому $\hat{y}(j+1)$ надо определить так, чтобы был минимизирован следующий функционал:

$$J_{j+1} = \left\| \begin{bmatrix} M_j \Phi^{-1}(j+1, j) \\ H(j+1) \end{bmatrix} \hat{y}(j+1) - \begin{bmatrix} \hat{z}_j^* \\ \hat{z}^*(j+1) \end{bmatrix} \right\|_{Q_{j+1}^*}^2 \quad (9.38)$$

при выполнении ограничений

$$g[\hat{y}(j+1)] \leq b. \quad (9.39)$$

Обозначение $\|x\|_{Q_{j+1}^*}^2$ определяет квадратичную форму $x^T Q_{j+1}^* x$, а Q_{j+1}^* обозначает положительно определенную матрицу. Предположим, что Q_{j+1}^* равно

$$Q_{j+1}^* = \begin{bmatrix} Q_j^* & 0 \\ 0 & Q_{j+1} \end{bmatrix}. \quad (9.40)$$

Развернув квадратичную форму (9.38), получим

$$J_{j+1} = \|M_j \Phi^{-1}(j+1, j) \hat{y}(j+1) - z_j^*\|_{Q_j^*}^2 + \|H(j+1) \hat{y}(j+1) - z(j+1)\|_{Q_{j+1}}^2. \quad (9.41)$$

Развернув J_{j+1} , исключая члены, не содержащие $\hat{y}(j+1)$, и обозначая остающиеся члены функционалом L_{j+1} ,

получаем

$$L_{j+1} = \hat{\mathbf{y}}^T(j+1) \{ [\Phi^{-1}(j+1, j)]^T M_j^T Q_j^* M_j \Phi^{-1}(j+1, j) + H^T(j+1) Q_{j+1} H(j+1) \} \hat{\mathbf{y}}(j+1) - 2[z_j^{*T} Q_j^* M_j \Phi^{-1}(j+1, j) + \mathbf{z}^T(j+1) Q_{j+1} H_j(j+1)] \hat{\mathbf{y}}(j+1). \quad (9.42)$$

Теперь для упрощения записи введем следующие обозначения:

$$D_j = M_j^T Q_j^* M_j, \quad (9.43)$$

$$D_{j+1} = [\Phi^{-1}(j+1, j)]^T D_j \Phi^{-1}(j+1, j) + H^T(j+1) Q_{j+1} H(j+1), \quad (9.44)$$

$$\mathbf{c}_j^T = 2\mathbf{z}_j^{*T} Q_j^* M_j, \quad (9.45)$$

$$\mathbf{c}_{j+1}^T = \mathbf{c}_j^T \Phi^{-1}(j+1, j) + 2\mathbf{z}^T(j+1) Q_{j+1} H(j+1). \quad (9.46)$$

Используя уравнения от (9.42) до (9.46), можно сформулировать следующую задачу нелинейного программирования:

$$\min \{ \hat{\mathbf{y}}^T(j+1) D_{j+1} \hat{\mathbf{y}}(j+1) - \mathbf{c}_{j+1}^T \hat{\mathbf{y}}(j+1) \mid \mathbf{g}[\hat{\mathbf{y}}(j+1)] \leq \mathbf{b} \}. \quad (9.47)$$

Рассмотрим теперь вычисление матрицы $\Phi^{-1}(j+1, j)$, которая входит в (9.44) и (9.46). Для удобства введем присоединенную матрицу, определяемую следующим образом:

$$\Psi^T(t, t_j) = \Phi^{-1}(t, t_j). \quad (9.48)$$

Если учесть, что

$$\Psi^T(t, t_j) \Phi(t, t_j) = I, \quad (9.49)$$

и использовать (9.32), можно получить следующее дифференциальное уравнение, которому удовлетворяет присоединенная система:

$$\dot{\Psi}(t, t_j) = -A^T(t) \Psi(t, t_j), \text{ при этом } \Psi(t_j, t_j) = I. \quad (9.50)$$

Алгоритм для решения задачи рекуррентной оценки с ограниченными переменными состояния формулируется следующим образом:

1. Вычислить $\Psi(j+1, j)$ путем интегрирования (9.50) в пределах от $t=t_j$ до $t=t_{j+1}$.

2. Пересчитать показатель качества L_{j+1} :

$$(a) D_{j+1} = \Psi(j+1, j) D_j \Psi^T(j+1, j) + H^T(j+1) Q_{j+1} H(j+1)$$

при

$$D_0 = \{\text{cov}[\hat{y}(0)]\}^{-1}, \quad (9.51)$$

где $\text{cov}(x) = E[xx^T]$, а E обозначает среднее значение или, что то же самое, математическое ожидание.

$$(b) c_{j+1} = \Psi(j+1, j) c_j + 2H^T(j+1) Q_{j+1} z(j+1) \quad (9.52)$$

при $c_0 = 2D_0 \hat{y}(0)$.

$$(c) L_{j+1} = \hat{y}^T(j+1) D_{j+1} \hat{y}(j+1) - c_{j+1}^T \hat{y}(j+1). \quad (9.53)$$

3. Решить задачу нелинейного программирования

$$\min \{L_{j+1} | g[\hat{y}(j+1)] \leq b\}, \quad (9.54)$$

причем начальное приближение можно взять равным $\Phi(j+1, j) \hat{y}(j)$.

9.5. Идентификация [16]

Одна из наиболее важных задач расчета систем управления заключается в определении динамических характеристик управляемого процесса или объекта. В этом параграфе излагается метод идентификации линейной системы по измерениям входа и выхода в дискретные моменты времени. Метод использует итерационную процедуру и линейное программирование [16].

Рассматриваемая задача идентификации (оценки параметров) заключается в определении математической модели, отображающей соотношение между последовательностями дискретных значений входных и выходных сигналов системы. Эта модель использует рациональные выражения в z -преобразованиях, причем предполагается, что система линейна, инвариантна относительно времени и имеет по одному входу и выходу.

Модель $G(z)$ определяется отношением рациональных полиномов

$$G(z) = \frac{N(z)}{D(z)}, \quad (9.55)$$

где

$$N(z) = \alpha_0 + \alpha_1 z^{-1} + \dots + \alpha_p z^{-p} \quad (9.56)$$

и

$$D(z) = 1 + \beta_1 z^{-1} + \dots + \beta_q z^{-q}. \quad (9.57)$$

Входная и выходная последовательности — это функции времени, обозначенные соответственно $x(t)$ и $y(t)$, значения которых определены в равностоящие моменты времени с периодом дискретизации T . Применяя z -преобразование, эти последовательности можно представить в виде

$$X(z) = x_0 + x_1 z^{-1} + \dots + x_{n-1} z^{-(n-1)} = \sum_{j=0}^{n-1} x_j z^{-j}, \quad (9.58)$$

$$Y(z) = y_0 + y_1 z^{-1} + \dots + y_{n-1} z^{-(n-1)} = \sum_{j=0}^{n-1} y_j z^{-j}, \quad (9.59)$$

где $[x_j]$ и $[y_j]$ — значения входной и выходной последовательностей в дискретные моменты и n — число дискретов.

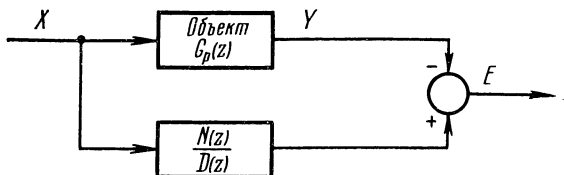


Рис. 9.8. Модель идентификации [16].

На рис. 9.8 показан объект и модель. Ошибка между реакцией модели и объекта (моделируемой системой) равна

$$E(z) = \frac{X(z) N(z)}{D(z)} - Y(z). \quad (9.60)$$

Полиномы $N(z)$ и $D(z)$ должны быть выбраны так, чтобы минимизировать

$$J = \sum_{j=0}^{n-1} |e(jT)|. \quad (9.61)$$

Это можно осуществить итерационным методом, который включает в себя линейное программирование.

Уравнение (9.60) можно приближенно представить следующим образом [17]:

$$E_i(z) = \frac{X(z) N_i(z)}{D_{i-1}(z)} - \frac{Y(z) D_i(z)}{D_{i-1}(z)}, \quad (9.62)$$

где i — номер итерации.

Деление на $D_{i-1}(z)$ действует на входную и выходную (рис. 9.9) последовательности $[x_j]$ и $[y_j]$ подобно цифровому фильтру, образуя фильтрованные входную и выходную последовательности $[x'_j]$ и $[y'_j]$. Соответственно этому уравнение (9.62) можно записать в виде

$$E_i(z) = X'_i(z) N_i(z) - Y'_i(z) D_i(z), \quad (9.63)$$

где

$$X'_i(z) = \frac{X(z)}{D_{i-1}(z)}, \quad Y'_i(z) = \frac{Y(z)}{D_{i-1}(z)}. \quad (9.64)$$

Решение уравнения (9.62) начинают, задаваясь некоторым начальным решением $D_0(z)$ и, далее, используя (9.64), преобразуют исходные входную и выходные последовательности $[x_j]$ и $[y_j]$ в фильтрованные последовательности $[x'_j]$ и $[y'_j]$. Затем эти фильтрованные последовательности подставляют в (9.63), что приводит к

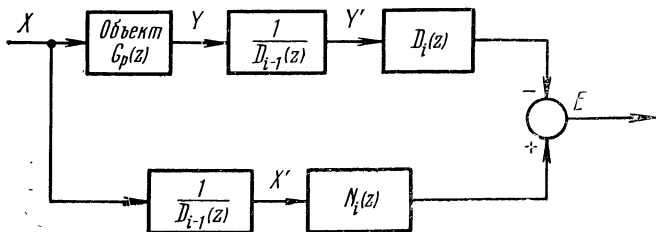


Рис. 9.9. Итерационная модель идентификации [16].

формуле, в которой коэффициенты в выражениях $N_1(z)$, $D_1(z)$ и $E_1(z)$ не определены. Эта формула представляет собой систему линейных ограничивающих уравнений, выражающих ошибку, т. е. разность между реакцией модели и требуемой реакцией в каждый дискретный момент.

Коэффициенты выражений для $N(z)$ и $D(z)$, представленных в виде (9.56) и (9.57), могут быть вычисле-

ны с помощью линейного программирования так, что при этом J в (9.61) будет минимальным.

Первое применение линейного программирования дает $D_1(z)$. Полученное $D_1(z)$ используется далее для вычисления новых входной и выходной последовательностей $[x'_j]$ и $[y'_j]$. Последние вновь подставляют в (9.63) для получения новых ограничивающих уравнений. Вновь применяя линейное программирование, получаем $D_2(z)$ (а попутно и $N_2(z)$ и $E_2(z)$).

Сходимость достигается, когда

$$D_i(z) \approx D_{i-1}(z). \quad (9.65)$$

Уравнение (9.63) можно переписать в виде

$$E_i(z) = \left[\frac{X(z)N_i(z)}{D_i(z)} - Y(z) \right] \frac{D_i(z)}{D_{i-1}(z)}. \quad (9.66)$$

Из (9.66) видно, что при выполнении (9.65) выражение (9.66) переходит в (9.60). Таким образом, итерационный метод идентификации является средством, обеспечивающим решение нелинейных вычислительных задач типа (9.60) путем многократного решения некоторой линейной задачи.

Ошибку системы, определяемую формулой (9.63), можно представить в следующей матричной форме:

$$X'n - Y'd - e = 0, \quad (9.67)$$

где матрицы X' и Y' равны

$$\left. \begin{aligned} X' &= \begin{bmatrix} x'_0 & 0 & \dots & 0 \\ x'_1 & x'_0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ x'_{n-1} & x'_{n-2} & \dots & x'_{n-1-q} \end{bmatrix}, \\ Y' &= \begin{bmatrix} y'_0 & 0 & \dots & 0 \\ y'_1 & y'_0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ y'_{n-1} & y'_n & \dots & y'_{n-1-q} \end{bmatrix}. \end{aligned} \right\} \quad (9.68)$$

Вектор-столбцы \mathbf{n} , \mathbf{d} и \mathbf{e} равны соответственно

$$\mathbf{n} = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_p \end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix} 1 \\ \beta_1 \\ \vdots \\ \beta_q \end{bmatrix}, \quad \mathbf{e} = \begin{bmatrix} e_0 \\ e_1 \\ \vdots \\ e_{n-1} \end{bmatrix}. \quad (9.69)$$

Индекс i в (9.63) подразумевается.

Матрицы X' и Y' известны для каждой итерации, а векторы \mathbf{n} , \mathbf{d} и \mathbf{e} неизвестны и их надо найти.

Так как компоненты вектора \mathbf{d} постоянные, то можно записать следующее соотношение:

$$\mathbf{d} = \mathbf{d}' + \mathbf{r},$$

где

$$\mathbf{d}' = \begin{bmatrix} 0 \\ \beta_1 \\ \vdots \\ \beta_q \end{bmatrix} \quad \text{и} \quad \mathbf{r} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \quad (9.70)$$

Подставляя (9.70) в (9.67), получим

$$X'\mathbf{n} - Y'(\mathbf{d}' + \mathbf{r}) - \mathbf{e} = \mathbf{0}. \quad (9.71)$$

Так как \mathbf{r} является единичным вектором, то (9.71) можно переписать в виде

$$X'\mathbf{n} - Y'\mathbf{d}' - \mathbf{y}' - \mathbf{e}' = \mathbf{0}, \quad (9.72)$$

где \mathbf{y}' — известный выходной вектор, равный

$$\mathbf{y}' = Y'\mathbf{r} = \begin{bmatrix} y'_0 \\ y'_1 \\ \vdots \\ y'_{n-1} \end{bmatrix}. \quad (9.73)$$

Так как \mathbf{y}' является известным выходным вектором, то (9.72) удобно переписать в виде

$$X'\mathbf{n} - Y'\mathbf{d}' - \mathbf{e} = \mathbf{y}'. \quad (9.74)$$

Требуется также, чтобы неизвестные векторы \mathbf{n} , \mathbf{d}' и \mathbf{e} не были бы ограничены по знаку. Неизвестные векторы можно представить в виде разности неотрицательных векторов (см. (7.3)):

$$\mathbf{n} = \mathbf{n}^+ - \mathbf{n}^-, \quad \mathbf{d}' = \mathbf{d}'^+ - \mathbf{d}'^- \quad \text{и} \quad \mathbf{e} = \mathbf{e}^+ - \mathbf{e}^-. \quad (9.75)$$

Подставляя (9.75) в (9.74), получим

$$X'n^+ - X'n^- - Y'd'^+ + Y'd'^- - e^+ + e^- = y'. \quad (9.76)$$

Это выражение определяет систему в каждый дискретный момент. Так как уравнение (9.76) линейно, то получить решения для неизвестных коэффициентов можно с помощью линейного программирования.

Чтобы удовлетворить различным требованиям, предъявляемым к системе, могут понадобиться дополнительные уравнения. Например, может потребоваться, чтобы на ошибки системы были наложены определенные ограничения для заданных дискретных моментов. Ограничения этого типа могут быть, например, такими:

$$-\xi_j' \leq e_j \leq \xi_j'' \quad (9.77)$$

В большинстве случаев ограничения, накладываемые на ошибки — симметричные, и, следовательно, задача заключается в ограничении абсолютной величины ошибки:

$$|e_j| \leq \xi_j \quad (9.78)$$

С учетом (9.75) ограничению (9.78) можно придать вид

$$e_j^+ \leq \xi_j \quad \text{и} \quad e_j^- \leq \xi_j \quad (9.79)$$

С помощью дополнительных переменных неравенства можно переписать в виде равенств (см. гл. 3). Представив их в матричной форме, запишем

$$\left. \begin{aligned} e^+ + s' &= \xi, \\ e^- + s'' &= \xi, \end{aligned} \right\} \quad (9.80)$$

где векторы s' , s'' неотрицательные.

Условия (9.76) и (9.80) определяют множество ограничивающих уравнений, используемых в итерационном методе идентификации. Уравнение (9.76) справедливо для каждого дискретного момента, а (9.80) только для тех моментов, для которых на ошибки наложены ограничения.

То обстоятельство, что (9.76) справедливо для каждого момента, ограничивает объем задачи, которые можно решать с помощью линейного программирования. Однако необязательно все эти ограничивающие уравнения должны использоваться в задаче минимизации.

Операцию фильтрации входных и выходных последовательностей, определенную уравнением (9.64), можно записать в следующем общем виде:

$$A'(z) = D'_{i-1}(z) A(z), \quad (9.81)$$

где

$$D'_{i-1}(z) = \frac{1}{D_{i-1}(z)} = 1 + d'_1(z) + \dots + d'_n(z) + \dots \quad (9.82)$$

Ряд, представляющий $D'_{i-1}(z)$ —бесконечный (но при практическом применении он усекается), за исключением тривиального случая, когда $D_{i-1}(z) = 1$. Входная последовательность $A(z)$ определяется следующим образом:

$$A(z) = a_0 + a_1 z^{-1} + \dots + a_{n-1} z^{-(n-1)}. \quad (9.83)$$

Вычисляя с помощью (9.81) фильтрованную последовательность $A'(z)$, получим

$$A'(z) = a_0 + (a_0 d'_1 + a_1) z^{-1} + \dots \\ \dots + (a_0 d'_{n-1} + \dots + a_{n-1}) z^{-(n-1)}. \quad (9.84)$$

Как видно из (9.84), члены более высоких степеней z^{-1} содержат составляющие с входными данными всех предыдущих дискретных моментов. Таким образом, можно выбрать те моменты, при которых реакция должна быть подобрана наиболее точно без потери информации о промежуточных моментах.

Это не относится к случаю, когда $D_{i-1}(z) = 1$. В этом случае необходимо рассматривать передаточную функцию, коэффициенты которой должны быть определены. Это следующее выражение:

$$A'(z) B(z), \quad (9.85)$$

где

$$A'(z) = a_0 + a'_1 z^{-1} + \dots + a'_{n-1} z^{-(n-1)}, \quad (9.86)$$

$$B(z) = b_0 + b_1 z^{-1} + \dots + b_m z^{-m} \quad (9.87)$$

и

$$m < n. \quad (9.88)$$

Надо определить коэффициенты b передаточной функции. С учетом уравнений (9.86) и (9.87) уравнение

(9.81) дает

$$a'_0 b_0 + (a'_0 b_1 + a'_1 b_0) z^{-1} + \dots \\ \dots + (a'_{n-1-m} b_m + a'_{n-m} b_{m-1} + \dots + a'_{n-1} b_0) z^{-(n-1)}. \quad (9.89)$$

Из (9.89) следует, что если припускать больше чем $m+1$ управление, то информация теряется. Так как решение $D_{i-1}(z) = 1$ должно быть вычислено, то при выборе интервала между моментами, которые должны быть согласованы, надо следить за указанным ограничением.

Применение линейного программирования требует, чтобы целевая функция была линейной. Большинство требуемых показателей качества можно получить, используя общий критерий минимизации суммы взвешенных абсолютных ошибок следующего вида:

$$\text{минимизировать } J = \sum_{j=0}^{n-1} c_j |e_j|. \quad (9.90)$$

Как уже неоднократно указывалось, эта целевая функция может быть (с учетом (9.75)) сведена к линейной:

$$\text{минимизировать } J = \sum_{j=0}^{n-1} c_j (e_j^+ + e_j^-). \quad (9.91)$$

В частном случае показатель качества, используемый в этом рассмотрении, есть сумма абсолютных ошибок (САО), которая получается если положить все c равными единице. Этот показатель качества будет иметь следующий вид:

$$\text{минимизировать } J = \sum_{j=0}^{n-1} (e_j^+ + e_j^-). \quad (9.92)$$

Однако этот показатель качества (9.92), используемый в качестве целевой функции линейного программирования, не представляет собой истинной суммы абсолютных ошибок (САО) решаемой задачи. Причина этого в том, что выражение (9.92) определяет ошибку системы, представленной на рис. 9.9, а не истинную ошибку системы, показанной на рис. 9.8. Истинную сумму абсолютных ошибок (САО) можно вычислить, используя заданные

входную и выходную последовательности $[x_j]$ и $[y_j]$ и вычисленные коэффициенты.

При использовании вычисленной суммы абсолютных ошибок можно считать, что сходимость наступила, если

$$\frac{|CAO_i - CAO_{i-1}|}{CAO_i} \leq V, \quad (9.93)$$

где CAO_i и CAO_{i-1} — вычисленные текущие и предыдущие значения суммы абсолютных ошибок, а V — любой заданный предел.

Дополнительные подробности, относящиеся к машинной программе, реализующей этот метод, описаны в диссертации Харисона [18]. Эту работу развил Барнс [19], который разработал специальный комплект программ для ЦВМ, реализующих описанный метод идентификации с использованием математического программирования.

Программа записана на Фотране IV и была реализована на машинной системе ИБМ 360 в Пенсильванском университете.

Литература

1. M. Aoki, Optimization of Stochastic Systems, Academic Press, New York, 1967. [Русский перевод: М. Аоки, Оптимизация стохастических систем, «Наука», 1971.]
2. A. P. Sage, Optimum Systems Control, Prentice-Hall, Englewood Cliffs, N. J., 1968, chaps. 8—11.
3. H. W. Sorenson, Kalman Filtering Techniques, Advan. Control Systems 3, pp. 219—232, C. T. Leondes, ed., Academic Press, New York (1966).
4. R. S. Bucy and P. D. Joseph, Filtering for Stochastic Processes with Applications to Guidance, Wiley-Interscience, New York, 1968.
5. N. E. Nahi, Estimation Theory and Applications, Wiley, New York, 1969.
6. A. E. Bryson and Y. C. Ho, Applied Optimal Control, Blaisdell, Waltham, Mass., 1969, chaps. 10—14. [Русский перевод: А. Брайсон, Хо Ю-ши, Прикладная теория оптимального управления, «Мир», 1972.]
7. J. R. Fisher, Optimal Nonlinear Filtering, Advan. Control Systems 5, pp. 197—300, C. T. Leondes, ed., Academic Press, New York (1967).
8. D. Tabak, Application of Mathematical Programming in the Design of Optimal Control Systems, Ph. D. Thesis. University of Illinois, Urbana, 1967.

9. D. T a b a k, Optimal Control of Nonlinear Discrete Time Systems by Mathematical Programming, J. Franklin Inst. **289**, pp. 111—119 (1970).
10. B. C. K u o, Analysis and Synthesis of Sampled-Data Control Systems, Prentice-Hall, Englewood Cliffs, N. J., 1963.
11. A. J. C a l i s e and K. A. F e g l e y, Quadratic Programming in the Statistical Design of Sampled-Data Control Systems, IEEE Trans. Automatic Control **AC-13**, pp. 77—80 (1968).
12. Л. А. В а й н ш т е й н, Б. Д. З у б а к о в. Выделение сигналов на фоне случайных помех, «Советское Радио», 1960.
13. J. T. T o u, Statistical Design of Digital Control Systems, IRE Trans. Automatic Control **AC-5**, pp. 290—297 (1960).
14. J. M a r o w i t z, Optimum Stochastic and Deterministic Control System Synthesis Using Quadratic Programming, Ph. D. Thesis, University of Pennsylvania, Philadelphia, 1969.
15. D. I s a a c s and W. D. L o c h r i e, Nonlinear Programming and Recursive Optimal Estimation, Allerton Conference of Circuits and Systems, Allerton, Ill., Oct., 1968.
16. R. W. H a r r i s o n and K. A. F e g l e y, Identification of Linear Systems using Mathematical Programming, 1968, WESCON, Session 14, Los Angeles, Cal.
17. K. S t e i g l i t z and L. E. M c B r i d e, A Technique for the Identification of Linear Systems, IEEE Trans. Automatic Control **AC-9**, pp. 461—464 (1965).
18. R. W. H a r r i s o n, Identification of Linear Systems Using Mathematical Programming, M. S. Thesis, University of Pennsylvania, Philadelphia, May 1968.
19. J. F. B u r n s, A Mathematical Programming Identification System (MPIS) Ph. D. Thesis, University of Pennsylvania, Philadelphia, June 1969.
- 20*. Я. З. Ц ы п к и н, Адаптация и обучение в автоматических системах, «Наука», 1968.
- 21*. Я. З. Ц ы п к и н, Основы теории обучающихся систем, «Наука», 1970.
- 22*. Р. Л и, Оптимальные оценки, определение характеристик и управление, «Наука», 1966.

СИСТЕМЫ С РАСПРЕДЕЛЕННЫМИ ПАРАМЕТРАМИ

10.1. Введение

Системы управления с распределенными параметрами представляют собой одну из областей, привлекающих наибольшее внимание исследователей [1—4]. Многие практические системы, как, например, ядерные реакторы [5] и космические корабли с нежесткой конструкцией [6, 7], представляют собой системы с распределенными параметрами. В течение нескольких последних лет было выполнено большое число работ, посвященных теоретическим и практическим вопросам систем оптимального управления с распределенными параметрами. Упомянутые здесь работы [8—17, 22—25] составляют небольшую долю имеющейся литературы. Подробный список литературы можно найти в [1—4]. Опубликовано несколько работ [18—20, 25—26], посвященных применению методов математического программирования к этому типу задач.

Сакава [18] применил квадратичное и линейное программирование к задаче об одномерном уравнении теплопроводности. Основные положения его работы изложены в 10.2. Эту работу развили Лорчирачункул и Пьер [19, 20] применительно к многомерным импульсным системам с распределенными параметрами. Их подход рассматривается в 10.3.

10.2. Задача оптимизации
для уравнения теплопроводности

Рассматриваемая система описывается следующим уравнением в частных производных [18]:

$$\frac{\partial^2 \tau(x, t)}{\partial x^2} = \frac{\partial \tau(x, t)}{\partial t}, \quad (10.1)$$

где $\tau(x, t)$ — распределение температуры в металле, x — пространственная координата, $0 \leq x \leq 1$, t — время, $0 \leq t \leq T$.

Время нормализовано таким образом, что коэффициент температуропроводности в (10.1) равен единице.

Начальные и граничные условия следующие:

$$\tau(x, 0) = 0, \quad (10.2)$$

$$\left. \frac{\partial \tau(x, t)}{\partial x} \right|_{x=0} = h [\tau(0, t) - g(t)], \quad (10.3)$$

$$\left. \frac{\partial \tau(x, t)}{\partial x} \right|_{x=1} = 0, \quad (10.4)$$

где h — коэффициент теплопередачи, который предполагается постоянным, $g(t)$ — температура газовой среды.

Температурой газовой среды $g(t)$ управляют с помощью расхода топлива $u(t)$. Эти величины связаны следующим дифференциальным уравнением:

$$\Theta \frac{dg(t)}{dt} + g(t) = u(t), \quad (10.5)$$

где Θ — постоянная времени топки.

Расход топлива $u(t)$ является управляющей переменной задачи, он управляет распределением температуры в металле посредством температуры газовой среды. Требуется получить определенное распределение температуры вдоль металла $\tau_d(x)$. Цель управления заключается в том, чтобы при заданном времени T минимизировать функционал

$$J = \int_0^1 [\tau_d(x) - \tau(x, T)]^2 dx. \quad (10.6)$$

Функционал J является мерой отклонения распределения температуры $\tau(x, T)$ при $t=T$ от заданного распределения $\tau_d(x)$.

Преобразовав по Лапласу уравнения от (10.1) до (10.5), получим следующую передаточную функцию системы [18]:

$$P(x, s) = \frac{\tau(x, s)}{U(s)} = \frac{\text{ch}(1-x) \sqrt{s}}{(\Theta s + 1) [(\sqrt{s}/h) \text{sh} \sqrt{s} + \text{ch} \sqrt{s}]}, \quad (10.7)$$

где $\tau(x, s) = L[\tau(x, t)]$, $U(s) = L[u(t)]$.

Согласно теореме свертки

$$\tau(x, t) = \int_0^t p(x, \sigma) u(t - \sigma) d\sigma = \int_0^t p(x, t - \sigma) u(\sigma) d\sigma, \quad (10.8)$$

где $p(x, t) = L^{-1}[P(x, s)]$.

Для момента окончания процесса $t=T$ имеем

$$\tau(x, T) = \int_0^T p(x, T - \sigma) u(\sigma) d\sigma. \quad (10.9)$$

Переменная управления $u(t)$ ограничена следующим образом:

$$0 \leq u(t) \leq 1, \quad 0 \leq t \leq T. \quad (10.10)$$

Прежде чем применить какой-либо метод математического программирования, дискретизируются интегралы (10.6) и (10.9). По формуле Симпсона [21] интеграл (10.6) можно представить в виде

$$J \simeq \sum_{i=0}^N c_i [\tau_d(x_i) - \tau(x_i, T)]^2, \quad (10.11)$$

где

$$\left. \begin{aligned} x_i &= \frac{i}{N}, \quad i = 0, 1, \dots, N, \quad N - \text{четное число,} \\ c_0 &= c_N = \frac{1}{3N}, \\ c_1 &= c_3 = \dots = c_{N-1} = \frac{4}{3N}, \\ c_2 &= c_4 = \dots = c_{N-2} = \frac{2}{3N}. \end{aligned} \right\} (10.12)$$

Применяя ту же формулу к интегралу (10.9), получим

$$\tau(x_i, T) \simeq T \sum_{j=0}^N c_j p(x_i, T - \sigma_j) u(\sigma_j), \quad (10.13)$$

где $\sigma_j = \frac{jT}{N}$, $j = 0, 1, \dots, N$.

Введем следующие обозначения:

$$\left. \begin{aligned} a_{ij} &= T c_j p(x_i, T - \sigma_j), \\ u_j &= u(\sigma_j), \\ \tau_{di} &= \tau_d(x_i). \end{aligned} \right\} (10.14)$$

Подставляя теперь (10.14) в (10.13) и, далее, в (10.11), получим

$$J \simeq \sum_{i=0}^N c_i \left(\tau_{di} - \sum_{j=0}^N a_{ij} u_j \right)^2. \quad (10.15)$$

Ограничения (10.10) запишутся в виде

$$0 \leq u_j \leq 1, \quad j=0, 1, \dots, N. \quad (10.16)$$

Теперь мы имеем классическую задачу математического программирования с $N+1$ переменной u_j и $N+1$ ограничениями в форме неравенств (10.16). Так как J квадратично, а ограничения линейны, то мы имеем задачу квадратичного программирования. Та же задача может быть сформулирована также в виде задачи линейного программирования, если воспользоваться следующим показателем качества:

$$J \simeq \sum_{i=0}^N c_i \left| \tau_{di} - \sum_{j=0}^N a_{ij} u_j \right|. \quad (10.17)$$

Абсолютная величина в (10.17) далее трактуется методами, описанными в гл. 5 и 7.

10.3. Импульсные многомерные системы с распределенными параметрами

Лорчирачункул и Пьер распространили работу Сакава на импульсные системы с распределенными параметрами с несколькими входами [19, 20]. Структура рассматриваемой системы схематически показана на рис. 10.1.

Вход системы представлен r -мерным вектором управления $\mathbf{m}(t)$. Каждая компонента вектора управления проходит через импульсный элемент, период которого равен T , и через фиксатор нулевого порядка, образуя таким образом r -мерный входной вектор $\mathbf{u}(t)$. Система с распределенными параметрами является одномерной и характеризуется выходной функцией $y(x, t)$. Выход зависит от пространственной координаты x и от времени t . Задан требуемый выход $y_d(x, t)$. Ошибка, определяе-

мая разностью между требуемым и фактическим выходом, равна

$$e(x, t) = y_d(x, t) - y(x, t). \quad (10.18)$$

Назначение управления заключается в минимизации ошибки; это и диктует выбор показателя качества. Пространственная координата x делится на N_x интервалов

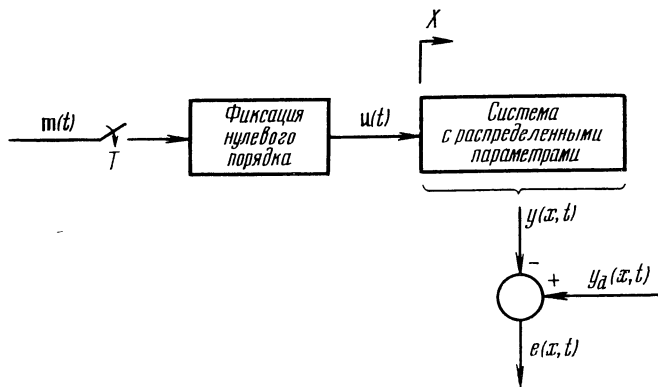


Рис. 10.1. Импульсное представление систем с распределенными параметрами.

дискретизации. Рассматриваемый общий временной интервал равен NT .

Выход определяется следующим выражением [20]:

$$y(x, t) = k(x, t) + \int_0^t g^T(x, t, \tau) u(\tau) d\tau, \quad (10.19)$$

где $k(x, t)$ — составляющая реакции системы, определяемая начальным распределением $y(x, 0)$, $g^T(x, t, \tau)$ — r -мерный вектор-строка, отражающий управляющее действие на выходное распределение $y(x, t)$.

Для j -го дискрета выхода можно записать

$$y(x, jT) = k(x, jT) + \sum_{i=1}^j h_j^T(x, iT) m(i-1), \quad (10.20)$$

где $m(i-1) = m[(i-1)T]$, $h_j^T(x, iT)$ — r -мерный вектор-

строка, заданный следующим образом:

$$\mathbf{h}_j^T(x, iT) = \int_0^T \mathbf{g}^T[x, jT, (i-1)T + \tau] d\tau. \quad (10.21)$$

Для момента окончания N периодов дискретизации, получим следующую систему уравнений:

$$\mathbf{Y}(x) = \mathbf{K}(x) + H(x)\mathbf{M}, \quad (10.22)$$

где

$$\mathbf{Y}_{N \times 1}(x) = \begin{bmatrix} y(x, T) \\ y(x, 2T) \\ \dots \\ y(x, NT) \end{bmatrix}, \quad \mathbf{K}_{N \times 1}(x) = \begin{bmatrix} k(x, T) \\ k(x, 2T) \\ \dots \\ k(x, NT) \end{bmatrix},$$

$$\mathbf{M}_{r \times N \times 1} = \begin{bmatrix} \mathbf{m}(0) \\ \mathbf{m}(1) \\ \dots \\ \mathbf{m}(N-1) \end{bmatrix},$$

$$H_{N \times N \times r}(x) = \begin{bmatrix} \mathbf{h}_1^T(x, T) & \mathbf{0}^T & \dots & \mathbf{0}^T \\ \mathbf{h}_2^T(x, T) & \mathbf{h}_2^T(x, 2T) & \dots & \mathbf{0}^T \\ \dots & \dots & \dots & \dots \\ \mathbf{h}_N^T(x, T) & \mathbf{h}_N^T(x, 2T) & \dots & \mathbf{h}_N^T(x, NT) \end{bmatrix}.$$

$\mathbf{0}^T = [0, 0, \dots, 0]$ — r -мерный нулевой вектор-строка.

Показатель качества принят следующего вида:

$$\text{минимизировать } J = \sum_{i=1}^N \sum_{j=1}^N a_{ij} |e(x_i, jT)|. \quad (10.23)$$

Ошибка e выражается с помощью (10.18) и (10.22) в виде функции вектора управления \mathbf{M} . Абсолютные значения в показателе качества преобразуются в линейную форму с помощью методов, описанных в гл. 5 и 7. Кроме того, на переменную управления наложены следующие ограничения:

$$0 \leq m_j(i) \leq m_{ji \max}, \quad i=0, 1, \dots, N-1; \quad j=1, 2, \dots, r. \quad (10.24)$$

При некоторых обстоятельствах, в промежутках между дискретами оптимального распределения могут появиться пульсации. Правильным выбором ограничений

производных ошибки можно эти пульсации эффективно уменьшить.

Дифференцируя (10.19) по времени, получим

$$\frac{\partial y(x, t)}{\partial t} = \frac{\partial k(x, t)}{\partial t} + \int_0^t \frac{\partial g^T(x, t, \tau)}{\partial t} u(\tau) d\tau + g^T(x, t, t) u(t). \quad (10.25)$$

j -я частная производная по t равна

$$\begin{aligned} \frac{\partial^j y(x, t)}{\partial t^j} &= \frac{\partial^j k(x, t)}{\partial t^j} + \int_0^t \frac{\partial^j g^T(x, t, \tau)}{\partial t^j} u(\tau) d\tau + \\ &+ \sum_{i=1}^j \frac{\partial^{j-i}}{\partial t^{j-i}} \left[u(t) \frac{\partial^{i-1} g^T(x, t, \tau)}{\partial t^{i-1}} \Big|_{\tau=t} \right]. \end{aligned} \quad (10.26)$$

При $t=iT$ получаем

$$\begin{aligned} \frac{\partial^j y(x, iT)}{\partial t^j} &= \frac{\partial^j k(x, iT)}{\partial t^j} + \int_0^{iT} \frac{\partial^j g^T(x, iT, \tau)}{\partial t^j} u(\tau) d\tau + \\ &+ \sum_{i=1}^j \frac{\partial^{j-i}}{\partial t^{j-i}} \left[u(t) \frac{\partial^{i-1} g^T(x, t, \tau)}{\partial t^{i-1}} \Big|_{\tau=t} \right]_{t=iT}. \end{aligned} \quad (10.27)$$

Аналогично, для j -й частной производной по x имеем

$$\frac{\partial^j y(x, t)}{\partial x^j} = \frac{\partial^j k(x, t)}{\partial x^j} + \int_0^t \frac{\partial^j g^T(x, t, \tau)}{\partial x^j} u(\tau) d\tau. \quad (10.28)$$

j -е частные производные ошибки по t и по x при $x=x_m$ и $t=iT$ равны

$$e_t^j(x_m, iT) = \frac{\partial^j y_d(x_m, iT)}{\partial t^j} - \frac{\partial^j y(x_m, iT)}{\partial t^j}, \quad (10.29)$$

$$e_x^j(x_m, iT) = \frac{\partial^j y_d(x_m, iT)}{\partial x^j} - \frac{\partial^j y(x_m, iT)}{\partial x^j}, \quad (10.30)$$

Чтобы уменьшить пульсации между дискретами, абсолютные значения временных и пространственных про-

изводных ошибки ограничиваются некоторыми постоянными величинами. Эти постоянные определяются при расчете на основе учета допустимой величины пульсаций в оптимальной системе.

Таким образом, имеем следующие ограничения в форме неравенств:

$$|e_t^j(x_m, iT)| \leq E_{t,mi}^j, \quad (10.31)$$

$$|e_x^j(x_m, iT)| \leq E_{x,mi}^j. \quad (10.32)$$

Абсолютные значения выражений, входящие в ограничения (10.31) и (10.32), преобразуются в линейные выражения согласно методам, описанным в гл. 5 и 7.

Пример. Для иллюстрации описанного метода рассмотрим систему, определяемую следующим уравнением:

$$\frac{\partial y(x, t)}{\partial t} = \frac{\partial^2 y(x, t)}{\partial x^2}, \quad t > 0, \quad 0 \leq x \leq 1, \quad (10.33)$$

с граничными условиями

$$y(0, t) = y(1, t) = u(t) \quad (10.34)$$

и

$$y(x, 0) = 0. \quad (10.35)$$

Надо найти такой граничный вход $u(t)$, чтобы при всех a_{ij} , равных единице, был минимизирован показатель качества (10.23). Требуемое распределение $y_d(x, t)$ равно 1, а на управление наложено ограничение

$$0 \leq m_j \leq 2. \quad (10.36)$$

Решение дифференциального уравнения (10.33) при граничных условиях (10.34) и (10.35) можно записать в виде

$$y(x, t) = 4 \int_0^t \sum_{n=1}^m \omega_{2n-1} \sin(\omega_{2n-1} x) \exp[-\omega_{2n-1}^2 (t - \tau)] u(\tau) d\tau, \quad (10.37)$$

где

$$\omega_{2n-1} = (2n-1)\pi. \quad (10.38)$$

Из теории рядов Фурье известно, что при $x=0$ и $x=1$ бесконечные ряды в правой части уравнения (10.37) не сходятся к $y(x, t)$. Однако из граничных условий (10.34) следует, что значения выходного распределения на границе равны входу $u(t)$.

Таким образом, исходя из уравнения (10.21), получаем, что каждый ненулевой элемент матрицы $H(x)$ равен

$$h_j(x, iT) = 4 \sum_{n=1}^m \frac{\sin(\omega_{2n-1}x)}{\omega_{2n-1}} \exp[-\omega_{2n-1}^2(j-i+1)T] \times \\ \times [\exp(\omega_{2n-1}^2T) - 1]. \quad (10.39)$$

Из уравнения (10.20) следует, что распределенный выход равен

$$y(x, jT) = \sum_{i=1}^j m_{i-1} h_j(x, iT). \quad (10.40)$$

Тогда, согласно (10.18), общий член последовательности ошибки равен

$$e(x_m, jT) = 1 - \sum_{i=1}^j m_{i-1} h_j(x_m, iT). \quad (10.41)$$

Из граничных условий (10.34) и (10.35) следует, что выходное распределение системы симметрично. Таким образом, характерные значения x должны быть определены либо для интервала $0 \leq x \leq 0,5$, либо для интервала $0,5 \leq x \leq 1$. В нашем примере N_x взято равным

Таблица 10.1

Оптимальная управляющая последовательность, обеспечивающая минимум показателя качества при $a_{ij} \equiv 1$, $m_{j\max} = 2,0$ и $T = 0,01$ сек

Время (сек)	Управляющая последовательность	Время (сек)	Управляющая последовательность
0,00	2,000	0,15	1,001
0,01	2,000	0,16	1,001
0,02	2,000	0,17	0,999
0,03	2,000	0,18	1,000
0,04	2,000	0,19	1,000
0,05	2,000	0,20	1,000
0,06	1,665	0,21	1,000
0,07	1,091	0,22	1,000
0,08	1,089	0,23	1,000
0,09	1,045	0,24	1,000
0,10	1,025	0,25	1,000
0,11	1,013	0,26	1,000
0,12	1,007	0,27	1,000
0,13	1,004	0,28	1,000
0,14	1,002	0,29	1,900

Реакция системы при оптимальном входе, показанном
в таблице 10.1

Распределительный выход $y(x, t)$

t	$x=0,1$	$x=0,2$	$x=0,3$	$x=0,4$	$x=0,5$
0,01	0,9590	0,3146	0,0678	0,0094	0,0016
0,02	1,2342	0,6347	0,2682	0,0964	0,0497
0,03	1,3666	0,8306	0,4499	0,2325	0,1649
0,04	1,4501	0,9683	0,6043	0,3824	0,3084
0,05	1,5115	1,0767	0,7393	0,5273	0,4554
0,06	1,5614	1,1682	0,8592	0,6628	0,5956
0,07	1,4436	1,1956	0,9557	0,7857	0,7253
0,08	1,1602	1,1231	1,0000	0,8819	0,8357
0,09	1,0920	1,0624	1,0000	0,9371	0,9109
0,10	1,0492	1,0339	1,0000	0,9660	0,9519
0,11	1,0268	1,0183	1,0000	0,9817	0,9741
0,12	1,0143	1,0099	1,0000	0,9901	0,9860
0,13	1,0077	1,0077	1,0000	0,9947	0,9925
0,14	1,0042	1,0029	1,0000	0,9971	0,9959
0,15	1,0022	1,0016	1,0000	0,9985	0,9978
0,16	1,0011	1,0008	1,0000	0,9992	0,9988
0,17	1,0008	1,0005	1,0000	0,9995	0,9993
0,18	0,9998	1,0001	1,0000	0,9998	0,9997
0,19	0,9999	0,9999	0,9999	0,9998	0,9998
0,20	0,9999	0,9999	0,9999	0,9998	0,9998
0,21	0,9999	0,9999	0,9999	0,9998	0,9999
0,22	0,9999	0,9999	0,9999	0,9999	0,9999
0,23	0,9999	0,9999	0,9999	0,9999	0,9999
0,24	0,9999	0,9999	0,9999	0,9999	0,9999
0,25	0,9999	0,9999	0,9999	0,9999	0,9999
0,26	0,9999	0,9999	0,9999	0,9999	0,9999
0,27	0,9999	0,9999	0,9999	0,9999	0,9999
0,28	0,9999	0,9999	0,9999	0,9999	0,9999
0,29	0,9999	0,9999	1,0000	0,9999	0,9999
0,30	0,9999	0,9999	1,0000	0,9999	0,9999

3 при значениях $x_1=0,1$, $x_2=0,3$ и $x_3=0,5$. Наконец, период дискретизации принят равным 0,1 сек и общее число интересующих нас дискретов равно 30.

На рис. 10.2 и в табл. 10.1 приведен оптимальный вход, вычисленный с помощью линейного программирования. Оптимальный выход вычислен с помощью уравнения (10.40) и показан на рис. 10.3,

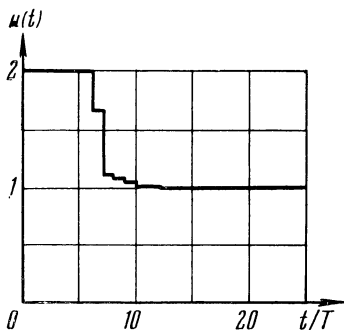


Рис. 10.2. Оптимальный вход, обеспечивающий минимум показателя качества при $a_{ij} \equiv 1$, $m_{j \max} = 2$ и $T = 0,01$ сек.

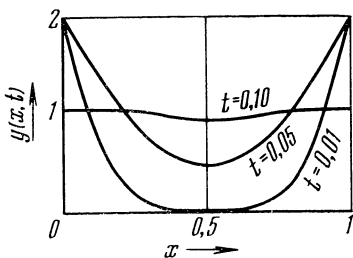


Рис. 10.3. Реакция выхода системы при оптимальном входе, показанном в табл. 10.2.

а также в табл. 10.2. Как видно из рис. 10.3, требуемое распределение во всех практических случаях может быть получено меньше чем за 0,15 сек.

Литература

1. M. Athans, Some Remarks on the Control of Distributed Parameter Systems, 1969 JACC, Boulder, Colo.
2. S. K. Mitter, Optimal Control of Distributed Parameter Systems, 1969 JACC, Boulder, Colo.
3. P. K. C. Wang, Mathematical Modeling of Systems with Distributed Parameters, 1969 JACC, Boulder, Colo.
4. J. C. Willems, A Survey of Stability of Distributed Parameter Systems, 1969 JACC, Boulder, Colo.
5. D. L. Briggs and C. N. Shen, Distributed Parameter Optimum Control of a Nuclear Rocket with Thermal Stress Constraints, Trans. ASME, J. Basic Eng., pp. 300—306, June 1967.
6. N. N. Puri and D. Tabak, Analysis and Control of Flexible Structure, Princeton Conference of Information Sciences and Systems, Princeton, N. J., March 1968, pp. 237—241.
7. D. Tabak, Analysis and Control of Space Vehicles with Flexible Appendages, Technical Report, Wolf R. and D. Corp., Riverdale, Md., Aug. 1968.
8. P. K. C. Wang, Control of Distributed Parameter Systems, Advances in Control Theory 1, edited by C. T. Leondes, pp. 75—172, Academic Press, New York (1964).

9. А. Г. Бутковский, Теория оптимального управления системами с распределенными параметрами, «Наука», 1965.
10. W. L. Brogan, Optimal Control Theory Applied to Systems Described by Partial Differential Equations, Advances in Control Theory 6, edited by C. T. Leondes, pp. 221—236, Academic Press, New York (1968).
11. M. H. Yeh and J. T. Tou, Optimum Control of a Class of Distributed Parameter Systems, IEEE Trans. Automatic Control **AC-12**, pp. 29—37 (1967).
12. H. Erzberger and M. Kim, Optimum Boundary Control of Distributed Parameter Systems, Inform. Control **9**, pp. 265—278 (1966).
13. А. И. Егоров, Об оптимальном управлении процессами в распределенных объектах, Прикладная математика и механика **27**, 688—698 (1963).
14. Y. Yavin and R. Sivan, The Optimal Control of a Distributed Parameter System, IEEE Trans. Automatic Control **AC-17**, pp. 758—761 (1967).
15. V. Комков, The Optimal Control of a Transverse Vibration of Beam, SIAM J. Control **6**, pp. 401—421 (1963).
16. A. P. Sage. Gradient and Quasilinearization Computational. Techniques for Distributed Parameter Systems, Intern. J. Control **6**, pp. 81—98 (1967).
17. E. I. Axelband, An Approximation Technique for the Optimal Control of Linear Distributed Parameter Systems with Bounded Inputs, IEEE Trans. Automatic Control **AC-11**, pp. 42—45 (1966).
18. Y. Sakawa, Solution of an Optimal Control Problem in a Distributed-Parameter System, IEEE Trans, Automatic Control **AC-9**, pp. 420—426 (1964).
19. V. Lorchirachonkul, Optimal Sampled-Data Control of Distributed-Parameter Systems, Ph. D. Thesis, Montana State University, Bozeman, March 1967.
20. V. Lorchirachonkul and D. A. Pierre, Optimal Control of Multivariable Distributed-Parameter Systems through Linear Programming, 1967 JACC pp. 702—710, Philadelphia, Pa.
21. A. D. Booth, Numerical Methods, Butterworth, London, 1955. [Русский перевод: Э. Д. Бут, Численные методы, «Наука», 1959.]
- 22*. Ю. В. Егоров, Некоторые задачи теории оптимального управления, Журнал вычислительной математики и математической физики **3**, № 5 (1963), 887—904.
- 23*. Ж.-Л. Лионс, Оптимальное управление системами, описываемыми уравнениями с частными производными, «Мир», 1972.
- 24*. Ж. Сеа, Оптимизация. Теория и алгоритмы, «Мир», 1973.
- 25*. Н. В. Баничук, В. М. Петров, Ф. Л. Черноусько, Численное решение вариационных и краевых задач методом локальных вариаций, Журнал вычислительной математики и математической физики **6**, № 6 (1966).
- 26*. Т. П. Иванова, Б. Т. Поляк, Г. В. Пухова, Численные методы решения некоторых экстремальных задач с частными производными, Вычислительные методы и программирование **9**, Изд-во МГУ (1967).

ИМЕННОЙ УКАЗАТЕЛЬ

- Абади (Abadie J.) 22, 27, 28, 29, 32, 39, 53, 55, 77, 83, 91
 Авдеева Л. И. 22, 28, 40
 Айзекс (Isaacs D.) 17, 20, 250, 263
 Аксельбанд (Axelband E. I.) 264, 275
 Арис (Aris R.) 11, 21
 Атанс (Athans M.) 11, 13, 15, 19, 79, 80, 91, 164, 264, 274
 Балински (Balinski M. L.) 28, 39
 Баничук Н. В. 264, 275
 Бар (Barг R. O., Jr.) 94, 144
 Барнс (Burns J. F.) 262, 263
 Бейтлер (Beightler C. S.) 11, 20, 39, 72, 73, 77, 85, 92, 167
 Беллман (Bellman R.) 11, 13, 15, 16, 20, 148, 166
 Бендерс (Benders J. F.) 29, 39
 Беркович Е. М. 90, 92
 Блум (Blum S.) 169, 207
 Болтянский В. Г. 11, 13, 14, 15, 19, 20, 91, 164, 208, 211, 236
 Босардж (Bosarge W. E.) 92
 Брайсон (Bryson A. E.) 11, 17, 20, 21, 83, 92, 238, 262
 Браун (Brown R. F.) 11, 21
 Брентани (Brentani P. B.) 85, 91
 Бригз (Briggs D. L.) 264, 274
 Броган (Brogan W. L.) 264, 275
 Брокштейн (Brockstein A. J.) 208
 Будах Б. М. 90, 92
 Букреев В. З. 11, 21
 Бут (Boot J. C. G.) 27, 39, 53, 55, 77
 Бут (Booth A. D.) 266, 275
 Бутковский А. Г. 11, 21, 264, 274
 Бьюси (Bucu R. S.) 238, 262
 Вайда (Vaida S.) 22, 27, 28, 39
 Вайнштейн Л. А. 243, 263
 Вален (Whalen B. H.) 18, 20, 169, 170, 175
 Валентайн (Valentine F. A.) 29, 39, 77
 Ван-Слайк (van Slyke R. M.) 84, 91, 125, 144
 Вань (Wang C. S.) 208, 236
 Вань (Wang P. K. C.) 264, 274
 Волин Ю. М. 167
 Вулф (Wolfe P.) 22, 27, 28, 35, 39, 40, 55, 57, 77, 125, 131
 Габасов Р. 11, 21
 Гамкрелидзе Р. В. 11, 13, 14, 15, 19, 91, 164, 208, 211, 236
 Гасс (Gass S. I.) 25, 39, 41, 43, 76
 Геглей (Gegley K. A.) 227, 230, 233, 237
 Гельфанд И. М. 13, 20
 Гирсамов И. В. 84, 92
 Гноенский Л. С. 83, 92
 Гольштейн Е. Г. 22, 25, 29, 37, 40, 41, 84, 92, 207
 Гомори (Gomory S. H.) 204, 207
 Грейвз (Grayes R. L.) 22, 27, 28, 39
 Гудвин (Goodwin G. C.) 11, 21
 Гуленко В. П. 90, 92, 167
 Гурвич (Hurwicz L.) 22, 28, 40, 84, 91
 Гурман В. И. 11, 21
 Давидон (Davidon W. C.) 74, 75, 78, 153, 167
 Даниель (Daniel J. W.) 72, 74, 78, 92
 Данилин Ю. М. 72, 74, 78
 Данциг (Dantzig G. B.) 25, 29, 39, 41, 43, 76, 125, 131, 144
 Даффин (Duffin R. J.) 39
 Дезоер (Desoer C. A.) 94, 95, 108, 144
 Демьянов В. Ф. 22, 28, 40
 Деннис (Dennis J. B.) 22, 28, 40
 Денхем (Denham W. F.) 17, 20
 Джалджури (Djadjuri K.) 169, 193, 194, 197, 207
 Джизмаджан (Jizmagian G. S.) 94, 96, 125, 127, 128, 130, 133, 134, 135, 144, 145
 Джозеф (Joseph P. D.) 238, 262
 Джонсон (Johnson O. G.) 92
 Джордан (Jordan B. W.) 208, 236
 Джури (Jury E. I.) 219, 237
 Дрейфус (Dreyfus S.) 11, 20
 Дубовицкий А. Я. 84, 92
 Е (Yeh M. H.) 264, 275
 Егоров А. И. 264, 275
 Егоров Ю. В. 264, 275
 Ермольев Ю. М. 90, 92, 167
 Заде (Zadeh L. A.) 18, 20, 94, 95, 108, 144, 169, 206
 Зангвилл (Zangwill W. I.) 22, 28, 40
 Зендер (Zehnder C. A.) 49, 53, 55, 77
 Зенер (Zener C. M.) 39
 Зойтендейк (Zoutendijk G.) 22, 27, 28, 39, 61, 62, 64, 65, 73, 77
 Зубаков В. Д. 243, 263
 Зуховицкий С. И. 22, 28, 40
 Иванилов Ю. П. 145
 Иванова Т. П. 264, 275
 Ивелей (Eveleigh V. W.) 11, 20
 Итон (Eaton J. H.) 169, 171, 173, 174, 207
 Каллам (Cullum C. D.) 83, 92, 237
 Каллам (Cullum J.) 85, 92
 Карлин С. 22, 28, 40
 Катлер (Cutler L.) 172, 179, 207

- Кац (Catz S.) 208, 236
 Келли (Kelley H. J.) 74, 78
 Кеннет (Kenneth P.) 17, 20
 Ким (Kim M.) 169, 171, 174, 193, 194, 197, 207, 264, 275
 Кириллова Ф. М. 11, 21
 Клохеси (Clohessy W. H.) 106, 107, 144
 Ковалик (Kowalik J.) 166
 Коллатц (Collatz L.) 166
 Комков (Komkov V.) 264, 275
 Копп (Kopp R. E.) 17, 20
 Корбут А. А. 22, 28, 40
 Красовский Н. Н. 11, 21
 Крелле (Krelle W.) 22, 27, 28, 32, 34, 39, 53, 55, 58, 77
 Крогов В. Ф. 11, 21
 Крэнделл (Crandall S. H.) 202, 207
 Кун (Kuhn H. W.) 32, 33, 34, 36, 39, 54, 56, 58, 84, 85, 210, 211, 237
 Куо (Kuo B. C.) 86, 92, 144, 168, 174, 179, 206, 207, 208, 213, 214, 220, 228, 237, 240, 263
 Курант (Courant R.) 83, 91, 152, 166
 Кэлайз (Calise A. J.) 242, 245, 246, 249, 250, 263
 Кэнон (Canon M. D.) 83, 92, 169, 171, 173, 174, 207, 211, 237
 Кэррол (Carroll C. W.) 69, 77
 Кюнци (Kunzi H. P.) 22, 27, 28, 32, 34, 39, 49, 53, 55, 58, 72, 74, 77, 78

 Ларсон (Larson R. E.) 13, 16, 17, 20
 Левелли (Llewellyn R. W.) 25, 39, 41, 43, 77
 Лейтман (Leitmann G.) 11, 20, 21
 Леондес (Leondes C. T.) 11, 21
 Летов А. М. 11, 21
 Ли (Lee B. W.) 219, 237
 Ли (Lee E. B.) 11, 20, 79, 80, 91, 164
 Ли (Lee R.) 263
 Лионс (Lions J. L.) 264, 275
 Лорчирачункул (Lorchirachoonkul V.) 264, 267, 268, 275
 Лохри (Lochrie W. D.) 250, 263
 Лэк (Lack G. N. T.) 169, 207
 Лэсдон (Lasdon L. S.) 74, 78, 163, 164
 Люенбергер (Luenberger D. G.) 11, 20, 164, 167

 Мак-Брайд (McBride L. E.) 256, 263
 Мак-Гилл (McGill R.) 17, 20
 Мак-Кормик (McCormick G. P.) 35, 37, 39, 69, 71, 72, 73, 74, 77, 163, 214, 223, 224, 232, 237
 Мангасарьян (Mangasarian O. L.) 211, 237
 Маркус (Markus L.) 11, 20, 79, 80, 91, 164
 Маровиц (Marowitz J.) 249, 263
 Мейер (Meyer R. R.) 165—167, 236, 237
 Мейерс (Meyers G. E.) 72, 74, 77, 78
 Мельц И. О. 167
 Мерриэм (Merriam C. W.) 11, 21
 Миллотин А. А. 84, 92
 Миттер (Mitter S. K.) 74, 78, 164, 167, 264, 274
 Мищенко Е. Ф. 11, 13, 15, 19, 91, 164, 208, 211, 236
 Мовшович С. М. 83, 92
 Монсеев Н. Н. 11, 20, 167

 Мойер (Moyer H. C.) 17, 20
 Молер (Mohler R. R.) 94, 139, 141, 142, 143, 144

 Нарендра (Narendra K. S.) 74, 78
 Нейштадт (Neustadt L. W.) 85, 91
 Норрис (Norris D. O.) 84, 91

 Орлов В. С. 164, 167
 Осборн (Osborne M. K.) 166
 Островский Г. М. 167

 Пауэлл (Powell M. J. D.) 72, 74, 75, 78
 Первозванский А. А. 83, 92
 Петров В. М. 264, 275
 Пирсон (Pierston B. L.) 152, 153, 154, 166
 Пирсон (Pearson J. B.) 211, 237
 Пирсон (Pearson J. D.) 61, 72, 73, 74, 77, 208, 236
 Питерсон (Peterson E. L.) 39
 Плант (Plant J. B.) 17, 20
 Полак (Polak E.) 92, 206, 207, 211, 236, 237
 Поляк Б. Т. 72, 74, 78, 164, 167, 208, 264, 275
 Понштейн (Ponstein J.) 84, 91
 Понтрягин Л. С. 11, 13, 14, 15, 19, 85, 91, 164, 208, 211, 236
 Попов В. М. (Popov V. M.) 156, 157, 167
 Порчелли (Porcelli G.) 169, 176, 178, 180, 182, 183, 206, 207, 227, 230, 233, 237
 Прайс (Price H. J.) 94, 139, 141, 142, 143, 144
 Пропой А. И. 11, 21, 83, 92, 145, 169, 174, 175, 206, 208, 236, 237
 Пухова Г. В. 264, 275
 Пшеничный В. Н. 84, 91, 92, 207
 Пьер (Pierre D. A.) 264, 267, 268, 275
 Пьюри (Puri N. N.) 264, 274

 Райс (Rice R. K.) 163, 167
 Рассел (Russel D. L.) 84, 91
 Ребрий В. А. 164, 167
 Ривз (Reeves C. M.) 75, 153, 166
 Робертс (Roberts J. J.) 146, 148, 149, 151, 166
 Розен (Rosen J. B.) 64, 68, 69, 77, 138, 145, 165, 166, 167, 208, 211, 233, 234, 235, 236, 237
 Розеноэр Л. И. 208, 236
 Рокафеллар (Rockafellar R. T.) 22, 29, 40
 Росточки (Rosztochy Z. K.) 146, 166
 Рубинов А. М. 22, 28, 40

 Сакава (Sakawa Y.) 264, 265, 275
 Сеа (Sea J.) 22, 28, 40, 264, 275
 Сейдж (Sage A. P.) 11, 20, 79, 80, 91, 164, 238, 262, 264, 275
 Сиван (Sivan R.) 264, 275
 Симонар (Simonard M.) 25, 39, 41, 43, 76
 Синнот (Sinnott J. F., Jr.) 164, 167
 Скарборо (Scarborough J. B.) 87, 90, 92
 Смит (Smith H. P.) 146, 148, 149, 151, 166
 Соловьева Е. Н. 90, 92
 Соренсон (Sorenson H. W.) 238, 262
 Спиди (Speedy C. D.) 11, 21
 Сридхар (Sridhar R.) 211, 237
 Стейглиц (Steiglitz K.) 256, 263

- Табак (Tabak D.) 72, 73, 74, 78, 86, 92, 94, 98, 144, 146, 156, 157, 158, 161, 163, 166, 167, 169, 174, 184, 188, 189, 193, 207, 211, 213, 214, 217, 218, 219, 221, 222, 227, 230, 233, 237, 239, 262, 263, 264, 274
- Таккер (Tucker A. W.) 32, 33, 34, 36, 39, 54, 56, 58, 84, 85, 210, 211, 237
- Торнг (Torng H. C.) 169, 170, 173, 174, 206
- Третьяков Н. В. 164, 167
- Трипати (Tripathi S. S.) 74, 78
- Ту (Tou J. T.) 208, 215, 236, 247, 263, 264, 275
- Уайлд (Wilde D. J.) 11, 20, 39, 72, 73, 77, 85, 92, 167
- Удзава (Uzava H.) 22, 28, 40, 91
- Уивер (Weaver L. E.) 146, 166
- Уиллемс (Willems J. C.) 264, 274
- Уилтшир (Wiltshire R. S.) 106, 107, 144
- Уорен (Waren A. D.) 74, 78, 163, 164, 167
- Уэспи (Waespy C. M.) 94, 103, 105, 111, 112, 125, 144
- Фалб (Falb P. L.) 11, 13, 15, 19, 79, 80, 91, 164
- Фан (Fan L. T.) 208, 236
- Фас (Fath A. E.) 94, 103, 104, 105, 113, 114, 115, 144
- Фегли (Fegley K. A.) 169, 176, 178, 179, 180, 182, 183, 206, 207, 242, 245, 246, 249, 250, 254, 255, 256, 263
- Федоренко П. П. 167
- Фельдбаум А. А. 11, 20, 21
- Фиакко (Fiacco A. V.) 35, 37, 39, 69, 71, 73, 74, 77, 163, 214, 223, 232, 237
- Финкельштейн Ю. Ю. 22, 28, 40
- Фишер (Fisher J. R.) 238, 262
- Флетчер (Fletcher R.) 74, 75, 78, 153, 166
- Фолц (Volz R. A.) 98, 100, 144
- Фомин С. В. 13, 20
- Фромовиц (Fromovitz S.) 211, 237
- Халкин (Halkin H.) 85, 91, 208, 236
- Харисон (Harrison R. W.) 254, 255, 256, 262, 263
- Хедли (Hadley G.) 22, 25, 27, 28, 29, 32, 34, 37, 38, 39, 41, 43, 45, 53, 51, 55, 57, 58, 76, 77
- Хензон (Hanson M. A.) 85, 91
- Хестенс (Hestenes M. R.) 74, 78
- Хиггинс (Higgins T. J.) 94, 103, 104, 144
- Хо Ю-ши (Ho Y. C.) 21, 83, 85, 91, 238, 262
- Хольцман (Holtzman J. M.) 208, 236
- Хсу (Hsu M. I.) 169, 206
- Цлаф Л. Я. 11, 21
- Цыпкин Я. З. 207, 263
- Чанг (Chang S. S. L.) 208, 236
- Чох (Tzschach H. G.) 49, 53, 55, 77
- Чернусько Ф. Л. 264, 275
- Шармак (Scharmack D. K.) 17, 20
- Шатровский Л. И. 167
- Шен (Shen C. N.) 174, 264
- Энз (Enns M.) 169, 207
- Эрроу (Arrow K. J.) 22, 28, 40, 91
- Эрцбергер (Erzberger H.) 275
- Эттли (Oettli W.) 72, 74, 78
- Эш (Ash M.) 137, 144, 146, 147, 148, 150, 166
- Юдин Д. Б. 22, 25, 29, 37, 40, 41, 207
- Явин (Yavin Y.) 264, 275

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

Алгоритм проектирования 73

Базис 43

Больша задача 13

Включение 115

Вулфа метод 55

Выпуклость 29, 30

Гамильтона функция 14

Гиперплоскость опорная 64

Двойственность 35, 37, 121

Задача Больша 13

— встречи при минимальном расходе топлива 105

— двойственная 35

Задача Лагранжа 13

— Майера 13

— минимаксная 148

— минимального времени 148, 203

— — расхода топлива 95, 103, 134, 169

— минимальной ошибки 174

— — квадратичной 195

— — энергии 94

— минимума квадратичного показате-

ля качества 171

— на быстродействие 95, 173

— оптимального управления 80, 84, 90

— — — в дискретной форме 86

— — — с непрерывным временем 82

— прямая 35

— управления, оптимального по быстродействию 113

Куна — Таккера теорема 33

Лагранжа задача 13

— множители 33

— функция 33

Майера задача 13

Максимизация без ограничений 22

Матрица направляющая 61, 72

Метод внутреннего штрафа 163

— возможных направлений 61

— Вулфа 55

— градиентный 58, 60

— наискорейшего подъема 61

— Ньютона обобщенный 73

— ограничивающих переменных 104

— переменной метрики 72

— последовательной минимизации без

ограничений 69

— проекции градиента 65

— разности переменных 105

— Флетчера — Пауэлла — Давидона 74

— Фрэнк и Вулфа 57

— штрафных функций 69

Метрика 61, 72

Минимум глобальный 31

— локальный 32

Множество выпуклое 29, 30

— достижимости 114

Множители Лагранжа 35

Направление возможное 62

Ньютона метод обобщенный 73

Ограничения 22

Переменные базисные 43

— дополнительные 41, 173

— небазисные 43

План 22

— оптимальный 22

Показатель качества 22, 81, 94, 95

Попова теорема 157

Принцип максимума Понтрягина 14

— оптимальности Беллмана 15

Программа главная ограничительная 126

Программирование геометрическое 28

— квадратичное 27, 53

— — параметрическое 128

— линейное 25, 41, 42

— — параметрическое 127

— — целочисленное 28

— нелинейное 28, 58

— обобщенное 125

— смешанное целочисленное 29

— стохастическое 29

Проекция вектора на подпростран-

ство 66

— градиента 68

Регулятор цифровой 176

Решение базисное 43

— допустимое 22, 43

— — базисное 43

— оптимальное 22

Симплекс-алгоритм 43, 45

Симплекс-таблица 46

Система линейная с дискретным вре-

менем 168

— — с заданными интервалами вре-

мени 98

— стохастическая 239

Теорема Дзуржи и Ли 219

— Кука — Таккера 33

— о седловой точке 33

— Попова 157

Точка крайняя 44

— седловая 33

Траектории допустимые 114

Управления допустимые 114

Условие регулярности первого поряд-

ка 36

— Слейтера 33

Функция вогнутая 31

— выпуклая 30

— Гамильтона 14

— Лагранжа 33

— строго вогнутая 31

— — выпуклая 30

— целевая 22

Д. Табак, Б. Куо
ОПТИМАЛЬНОЕ УПРАВЛЕНИЕ
И МАТЕМАТИЧЕСКОЕ
ПРОГРАММИРОВАНИЕ

(Серия: «Теоретические основы технической кибернетики»)

М., 1975 г., 280 стр. с илл.

Редактор *Б. Т. Поляк*
Техн. редактор *Н. Я. Мурашова*
Корректор *Е. Я. Строева*

Сдано в набор 11/XI 1974 г. Подписано к печати
13/VI 1975 г. Бумага 84X108¹/₃₂. Физ. печ. л. 8,75, Условн.
печ. л. 14,7. Уч.-изд. л. 14,33 . Тираж 18 000 экз. Цена
книги 1 р. 19 к. Заказ № 266.

Издательство «Наука»
Главная редакция физико-математической литературы
117071, Москва, В-71, Ленинский проспект, 15

630077, г. Новосибирск, 77, Станиславского, 25.
4-я типография издательства «Наука».

