

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И
ИНФОРМАТИКИ**

Кафедра методов оптимального управления

**ДЕЦЕНТРАЛИЗОВАННОЕ ОПТИМАЛЬНОЕ
УПРАВЛЕНИЕ**

Курсовая работа

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

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

Минск, 2020

ОГЛАВЛЕНИЕ

	С.
ВВЕДЕНИЕ	3
ГЛАВА 1 ОСНОВНЫЕ ПОНЯТИЯ И ОБЗОР ЛИТЕРАТУРЫ	4
1.1 Задачи оптимального управления	4
1.2 Программные и позиционные решения	6
1.3 Управление в реальном времени.	8
1.4 Численные методы решения задач оптимального управления	9
ГЛАВА 2 ЦЕНТРАЛИЗОВАННОЕ ОПТИМАЛЬНОЕ УПРАВЛЕНИЕ	12
2.1 Задача оптимального управления группой динамических систем.	12
2.2 Построение централизованной обратной связи в реальном времени	15
2.3 Решение задачи оптимального управления $P(\tau)$	16
ГЛАВА 3 РЕЗУЛЬТАТЫ ЧИСЛЕННЫХ ЭКСПЕРИМЕНТОВ	19
3.1 Сведение задачи к задаче линейного программирования	20
3.2 Построение обратной связи и соответствующей траектории	21
3.3 Результаты. Сравнение с программным решением	25
ЗАКЛЮЧЕНИЕ	27
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	28

ВВЕДЕНИЕ

В работе рассматривается задача оптимального управления группой динамических взаимосвязанных систем, а именно случай централизованного оптимального управления в реальном времени с переводом объектов управления в терминальное состояние, удовлетворяющее общему терминальному ограничению, минимизируя при этом полный импульс управляющего воздействия.

Актуальность настоящей работы связана с многочисленными приложениями задач группового управления, в которых точность программного решения не удовлетворяет требованиям в силу существующих возмущений, действующих на систему в процессе управления.

Структура дипломной работы представлена следующим образом: в главе 1 формулируются основные понятия, используемые в курсовой работе, описывается алгоритм работы оптимального регулятора, рассматриваются последовательный и параллельный подход; в главе 2 в общем виде рассматривается задача оптимального управления группой линейных взаимосвязанных систем, описаны два принципа управления – централизованный и децентрализованный, по пунктам расписывается алгоритм работы централизованного оптимального регулятора, а также демонстрируются подходы для сведения централизованной задачи к задаче линейного программирования; в главе 3 решается конкретная задача оптимального управления, а именно строится реализация обратной связи, и производится сравнение решения с программным – так демонстрируется эффективность предложенных методов.

ГЛАВА 1

ОСНОВНЫЕ ПОНЯТИЯ И ОБЗОР ЛИТЕРАТУРЫ

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

1.1 Задачи оптимального управления

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

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

2) Математическая модель. Динамический процесс обычно моделируется дифференциальными

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

или разностными уравнениями

$$x(k+1) = f(x(k), u(k), k), \quad k = 0, 1, \dots,$$

где n -вектор x называется состоянием системы, r -вектор u называется управ-

лением, функция $f : \mathbb{R}^n \times \mathbb{R}^r \times \mathbb{R} \rightarrow \mathbb{R}^n$ задана.

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

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

4) Ограничения на фазовую траекторию. Ограничения на переменные состояния могут накладываться в начальный момент времени t_0 :

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

в конечный момент времени t_f , такие ограничения называются терминальными:

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

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

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

на всем промежутке управления — фазовые ограничения

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

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

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

5) Критерий качества. Множество допустимых управлений, как правило, содержит более одного элемента, поэтому возникает необходимость сравнивать их между собой. Для этого вводится функционал $J(u)$, называемый критерием качества, и выбирается операция минимизации или

максимизации этого функционала, результат которой определяет наилучшее (оптимальное) управление. В теории оптимального управления различают четыре типа критериев качества: Майера, Больца, Лагранжа, задачи быстродействия. Все 4 критерия качества эквивалентны между собой.

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

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

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

1.2 Программные и позиционные решения

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

$$J(u) = c'x(t_f) \rightarrow \min, \quad (1.1)$$

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

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

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

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

Задача (1.1) будет исследоваться в классе дискретных управляющих воздействий

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

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

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

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

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

$$J(u) = c'x(t_f) \rightarrow \min, \quad (1.2)$$

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

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

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

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

Определение 1.5 Функция

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

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

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

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

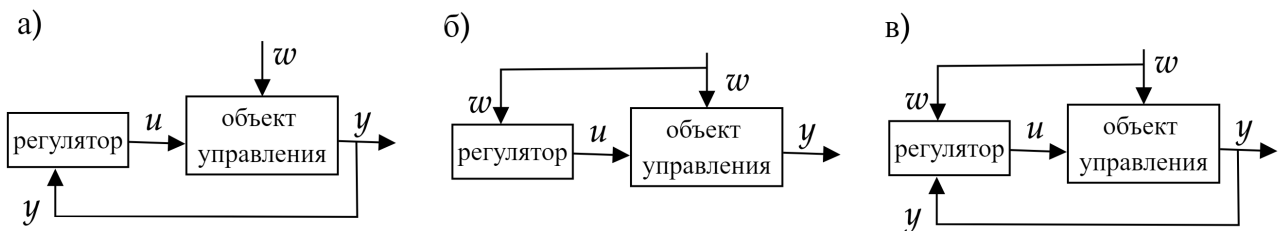


Рис. 1.1: а) обратная связь; б) прямая связь; в) комбинированная связь

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

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

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

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

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

1.3 Управление в реальном времени

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

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

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

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

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

T_h , в конкретном процессе управления.

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

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

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

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

1.4 Численные методы решения задач оптимального управления

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

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

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

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

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

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

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

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

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

Тогда исходная задача примет вид:

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

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

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

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

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

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

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

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

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

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

$$s_0 = x_0,$$

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

В итоге получим оптимизационную задачу вида:

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

$$s_0 - x_0 = 0,$$

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

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

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

ГЛАВА 2

ЦЕНТРАЛИЗОВАННОЕ ОПТИМАЛЬНОЕ УПРАВЛЕНИЕ

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

2.1 Задача оптимального управления группой динамических систем

Определим T и T_h как в главе 1:

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

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

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

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

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

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

образом $u \in \mathbb{R}^r$, $x \in \mathbb{R}^n$.

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

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

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

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

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

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

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

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

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

Целью управления является минимизация линейного терминального критерия качества:

$$\sum_{i \in I} c'_i x_i(t_f) \rightarrow \min_u, \quad (2.4)$$

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

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

$$c'x \rightarrow \min_u. \quad (2.5)$$

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

$$\sum_{i \in I} c'_i x_i(t_f) \rightarrow \min_u, \quad (2.6)$$

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

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

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

Группой динамических объектов (2.1) можно управлять централизованно и децентрализованно [5].

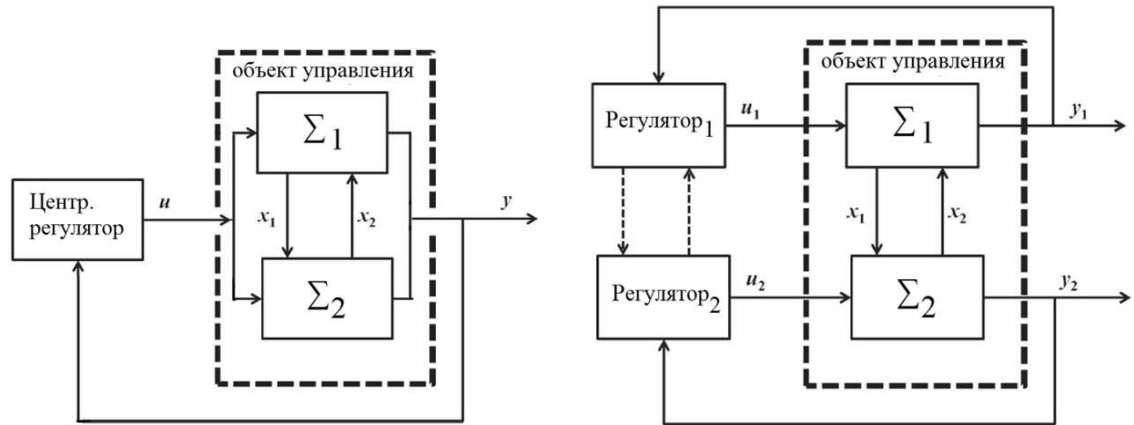


Рис. 2.1: Схема централизованного и децентрализованного управления

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

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

В настоящей главе рассматривается случай централизованного управления.

2.2 Построение централизованной обратной связи в реальном времени

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

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

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

Пусть имеется общий центр управления, который для каждого момента времени $\tau \in T_h$ по точной информации о состоянии группы

$$x^*(\tau) = \begin{pmatrix} x_1^*(\tau) \\ \vdots \\ x_q^*(\tau) \end{pmatrix}$$

вырабатывает управляющие воздействие (текущее значение оптимальной обратной связи)

$$u^*(t) = \begin{pmatrix} u_1^*(t) \\ \vdots \\ u_q^*(t) \end{pmatrix}, \quad t \in [\tau, \tau + h[, \quad (2.8)$$

которое подается на вход объекта управления до тех пор, пока не будет измерено и обработано следующее состояние $x^*(\tau + h)$.

Напомним, что такое построение оптимальной обратной связи называется ее реализацией в реальном времени. Это построение осуществляет оптимальный регулятор (см. главу 1), в данном случае — централизованный оптимальный регулятор.

Согласно ([5], с. 1713), для построения реализации (2.8) оптимальный регулятор решает задачу $P(\tau)$:

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

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

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

$$Hx(t_f) = g,$$

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

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

Оптимальную программу задачи $P(\tau)$ будем обозначать $u^0(t|\tau, x^*(\tau)), t \in [\tau, t_f]$.

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

1. Положить $\tau = t_0, x^*(\tau) = x_0$.
2. Найти оптимальную программу $u^0(t|\tau, x^*(\tau)), t \in [\tau, t_f]$, задачи $P(\tau)$.
3. Задать значение оптимальной обратной связи для позиции $(\tau, x^*(\tau))$:

$$u^0(\tau, x^*(\tau)) = u^0(\tau|\tau, x^*(\tau)),$$

и подать на объект управления управляющее воздействие

$$u^*(t) \equiv u^*(\tau) := u^0(\tau, x^*(\tau)), \quad t \in [\tau, \tau + h[.$$

4. В момент $\tau + h$ измерить $x^*(\tau + h)$, положить $\tau := \tau + h$, при $\tau < t_f$ вернуться к шагу 2.

Основная трудоемкость реализации данного алгоритма приходится на шаг 2, на котором необходимо быстро (за время, не превосходящее h) решать задачу оптимального управления $P(\tau)$.

2.3 Решение задачи оптимального управления $P(\tau)$

Для решения заметим, что в классе дискретных управлений задача $P(\tau)$ сводится к задаче линейного программирования. При сведении можно пользоваться последовательным или параллельным подходами (см. главу 1).

Сначала рассмотрим задачу $P(t_0, x_0)$ — это задача построения оптимального программного решения исходной задачи оптимального управления группой. Эта задача может решаться до начала процесса управления в реальном времени, т.е. до запуска алгоритма работы оптимального регулятора.

Рассмотрим параллельный подход. Для этого запишем формулу Коши

для системы в непрерывном времени:

$$x(t) = F(t, t_0)x_0 + \int_{t_0}^t F(t, \theta)B(\theta)u(\theta) d\theta. \quad (2.9)$$

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

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

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

В итоге задача (2.6) примет вид:

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

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

$$x(t_0) = x_0,$$

$$Hx(t_f) = g,$$

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

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

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

Рассмотрим последовательный подход. Подставим в формулу Коши (2.9) $t = t_f$, а получившуюся величину $x(t_f)$ подставим в исходную задачу (2.6). Тогда (2.6) имеет вид:

$$\sum_{s \in T_h} c'_h(s)u(s) \rightarrow \min, \quad (2.10)$$

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

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

где

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

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

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

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

Таким образом, при применении параллельного подхода получается задача линейного программирования большей размерности. Однако при этом матрица основных ограничений имеет специальную структуру, является разреженной (содержит большое количество нулей). Для ее решения применяют методы линейного программирования, которые способны учитывать разреженность матрицы для ускорения работы метода.

При последовательном подходе матрица основных ограничений плотно-заполненная. Но в ней тоже можно выделить специальную динамическую структуру — соседние столбцы при малых h почти коллинеарны, а в базис (опору) задачи оптимального управления войдут те столбцы, в которых находятся нули коуправления. В работе [3] выделена такая структура и предложен специальный метод решения задачи оптимального управления на основе задачи линейного программирования (2.10). Последняя называется функциональной формой задачи оптимального управления P .

В настоящей работе при реализации алгоритма работы оптимального регулятора на шаге 2 используется решение на основе функциональной формы. Для задачи $P(\tau)$ она имеет вид

$$\sum_{s \in T_h(\tau)} c'_h(s) u(s) \rightarrow \min, \quad (2.11)$$

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

$$u_* \leq u(s) \leq u^*, \quad s \in T_h(\tau),$$

где $T_h(\tau) = [\tau, t_f] \cap T_h$; $\tilde{g} = g - HF(t_f, \tau)x^*(\tau)$.

Таким образом, в настоящей главе сформулирована задача оптимального управления группой линейных взаимосвязанных систем и описаны два принципа управления группой — централизованный и децентрализованный. Исследован централизованный принцип: предложен алгоритм работы централизованного оптимального регулятора, приведены два подхода для сведения централизованной задачи оптимального управления к задаче линейного программирования.

ГЛАВА 3

РЕЗУЛЬТАТЫ ЧИСЛЕННЫХ ЭКСПЕРИМЕНТОВ

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

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

Управляющее воздействия ограничены по модулю $|u_i(t)| \leq L, t \in [t_0, t_f], i \in I = \{1, 2, 3\}$, и в момент времени t_f требуется перевести объект (3.1) на терминальное множество, задаваемое ограничениями

$$|z_i(t_f)| \leq d_1, \quad |\dot{z}_i(t_f)| \leq d_2, \quad i \in I,$$

а критерий качества имеет вид:

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

Система (3.1) описывает систему трех масс (выбраны равными 1), соединенных одинаковыми пружинами жесткости k . Управления приложены к каждой из масс.

Параметры задачи выберем следующим образом:

$$k = 10, \quad L = 1, \quad t_0 = 0, \quad t_f = 4.5;$$

$$d_i = 0.1, \quad \dot{z}_i(0) = 1, \quad z_i(0) = 1, \quad i \in I.$$

Пусть также $N = 100 \Rightarrow h = \frac{1}{N} = \frac{1}{100}$,

$$w(t) = \begin{pmatrix} 0.7 \sin 2t \\ 0.1 \cos t \\ -0.05 \cos 3t \end{pmatrix}.$$

3.1 Сведение задачи к задаче линейного программирования

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

Произведем замену:

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

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

$$\int_{t_0}^{t_f} \sum_{i \in I} |u_i(t)| dt \rightarrow \min, \quad (3.2)$$

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + Mw(t), \\ x(0) &= x_0, \quad t \in [t_0, t_f], \quad Hx(t_f) \leq g, \end{aligned}$$

где

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

$$H = \begin{pmatrix} E_6 \\ -E_6 \end{pmatrix},$$

$$g = 0.1 \cdot \mathbb{1}_{12},$$

где $E_6 \in \mathbb{R}^{6 \times 6}$ – единичная матрица

Отметим, что, хоть в записанной задаче (3.2) и присутствует возмущение w , оптимальная обратная связь будет определяться по детерминированной задаче $P(\tau)$. Такой подход соответствует классическому. Иными словами, для отыскания оптимальной обратной связи будем использовать следующую математическую модель:

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

Используя последовательный подход, сведем задачу (3.2) к задаче линейного программирования, аналогично тому, как сводили задачу (2.6) к задаче (2.10). Если после этого применить замену $u(s) = z(s) - v(s)$, $s \in T_h$, то в итоге задачу (3.2) можно записать следующим образом:

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

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

где $c = \mathbb{1}_{2r}$; $D_h(s)$ и \tilde{g} задаются аналогично тому, как они задавались во второй главе:

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

$$\tilde{g} = g - HF(t_f, t_0)x_0, \quad \tilde{g} \in \mathbb{R}^m.$$

3.2 Построение обратной связи и соответствующей траектории

Построение программного решения — первый шаг для построения обратной связи.

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

```

1 function Ale = Form_LP()
2
3     for_d = @(t)H*F(t_f-t)*b;
4     get_d_h = ...
5         @(s)integral(for_d,s,s+h,'ArrayValued', true);
6     d_h_values = zeros(m, r, N_initial);
7
8     for i = 1:N_initial
9         d_h_values(:, :, i) = get_d_h(t_0 + i*h-h);
10    end
11
12    Ale = [];
13    for i = 1:N_initial
14        Ale = [Ale d_h_values(:, :, i) ...
15              -d_h_values(:, :, i)];
16    end
17 end

```

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

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

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

В функции

```

1 function u = P(tau, z)
2
3     N = round((t_f - tau)/h); %new N for new tau
4
5     g_wave = g - H * F(t_f - tau) * z;
6
7     c = ones(1, 2*r*N);
8
9     ub = L*ones(2*r*N, 1);
10
11     startBlockNumber = N_initial - N + 1; % ...
12         1...N_initial

```

```

13     Opt=optimset('TolFun',1e-9,'TolX',1e-9);
14
15     z_and_v = ...
        linprog(c,Ale(:,2*r*startBlockNumber - 5 : ...
        2*r*N_initial),g_wave,[],[],zeros(2*r*N, ...
        1),ub,Opt);
16
17     zv = reshape(z_and_v, 2*r, N);
18     u   = zv(1:r,:) - zv(r+1:2*r,:); % r /times N
19 end

```

определяется оптимальная программа для позиции $(\mathbf{tau}, \mathbf{z})$. Входным параметром является позиция $(\mathbf{tau}, \mathbf{z})$.

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

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

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

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

```

1 U_0 = P(t_0, x_0);
2
3 X_0 = trajectory(x_0, t_0, t_f, U_0);

```

Функция

```

1 function x = trajectory(x0, t_begin, t_end, u)
2     N = round((t_end - t_begin)/h);
3     x = zeros(n, N);
4     x(:,1) = x0;

```

```

5     for j = 1:N
6         curr = t_begin + (j-1)*h;
7         next = t_begin + j*h;
8         x(:, j+1) = F(h) * x(:, j) + ...
            integral(@(t) F(next - t)*b, curr, next, ...
                'ArrayValued', ...
                true, 'RelTol', 0, 'AbsTol', 1e-12)*u(:, j) ...
            + integral(@(t) F(curr - t)* M * w(t), ...
                curr, next, 'ArrayValued', ...
                true, 'RelTol', 0, 'AbsTol', 1e-12);
9     end
10 end

```

принимает параметры для построения траектории: x_0 — начальное состояние, t_{begin} , t_{end} , u — начало, конец управление и само управление. Она возвращает траекторию для указанного отрезка времени.

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

```

1  X = zeros(n, N_initial);
2
3  X(:, 1) = x_0;
4
5  U = zeros(r, N_initial);
6
7  for k = 1:N_initial
8      tau = t_0 + (k - 1)*h;
9
10     u_res = P(tau, X(:, k));
11
12     U(:, k) = u_res(:, 1);
13
14     step_trajectory = trajectory(X(:, k), tau, ...
        tau + h, U(:, k));
15
16     X(:, k + 1) = step_trajectory(:, 2);
17 end

```

Стоит заметить, что в функции `function u = P(tau, z)` для отыскания позиционного решения для позиции (τ, z) в процедуру `linprog` вносится часть

уже построенной (функцией `function Ale = Form_LP()`) матрицы `Ale`. Для позиции (t_0, x_0) в `linprog` подается матрица вида (3.4), однако для следующей позиции $(t_0 + h, x(:, 2))$ в `linprog` вносится матрица без первых $2r$ столбцов:

$$(d_h(t_1) \quad -d_h(t_1) \quad \dots \quad d_h(t_f) \quad -d_h(t_f)).$$

3.3 Результаты. Сравнение с программным решением

Значение критерия качества для программного решения: 285.0503.

Значение критерия качества для оптимальной обратной связи: 217.5017 – это на 23.6971% лучше программного.

Среднее время выполнения функции $P(\tau, z)$: 0.029 секунды.

Для визуальной оценки результатов перенесем полученные траектории на фазовые графики. Для q объектов управления получится q фазовых графиков (программное решение отмечено курсивом):

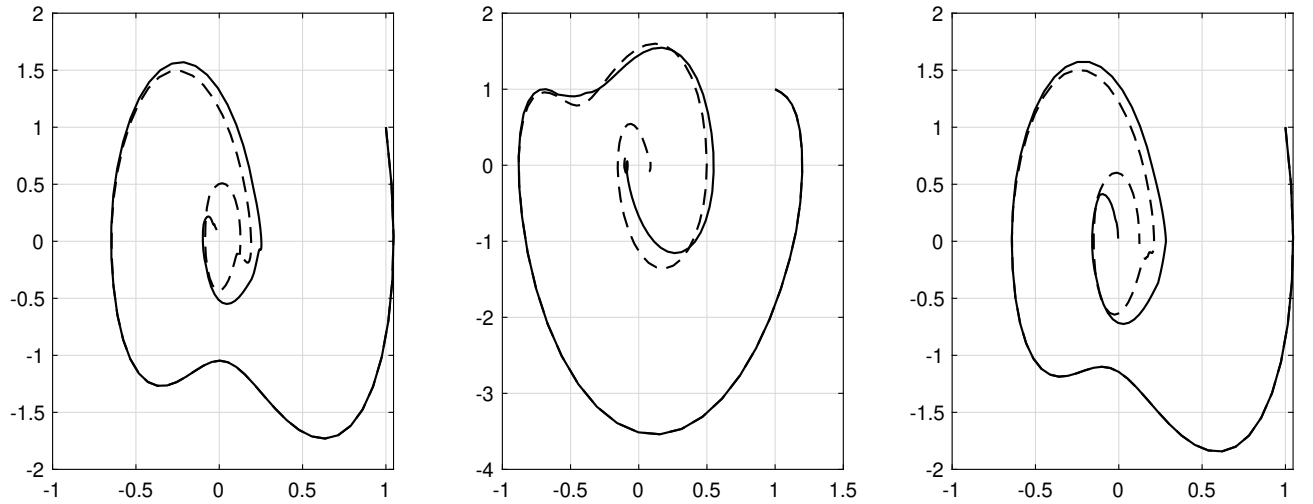


Рис. 3.1: Фазовые графики

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

Таким образом, решена задача оптимального управления для объекта, состоящего из трех взаимосвязанных систем, она сведена к задаче линейного

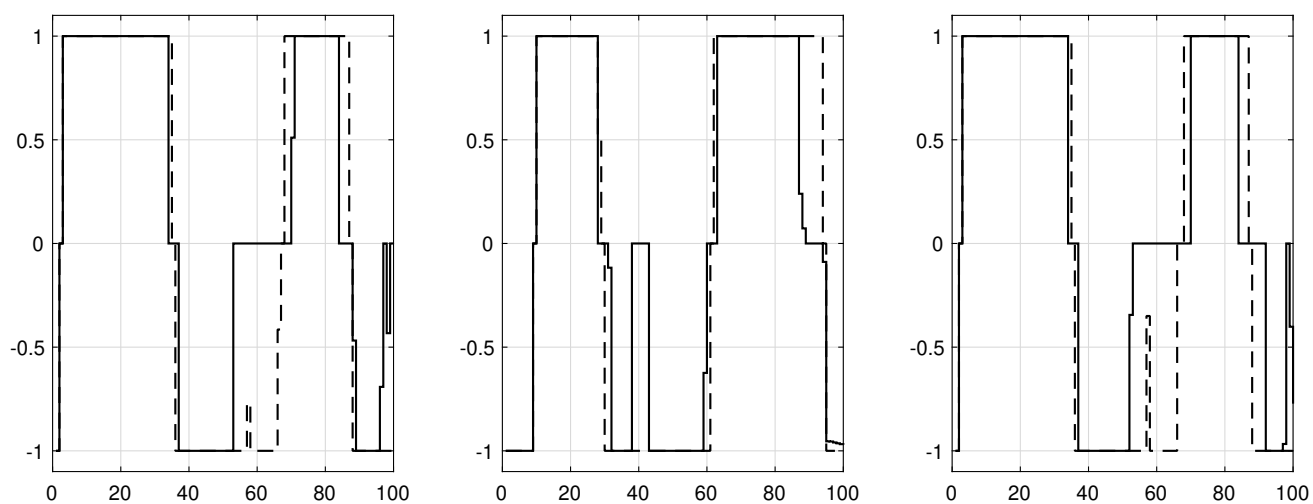


Рис. 3.2: Соответствующие фазовым графикам управления

программирования, построено программное решение и реализация обратной связи и произведен анализ полученных результатов.

ЗАКЛЮЧЕНИЕ

В работе исследована задача централизованного управления группой взаимосвязанных систем в реальном времени. Для исследуемой задачи сформулированы подходы ее сведения к задаче линейного программирования, предложен алгоритм работы централизованного оптимального регулятора. В дальнейшем с помощью последовательного подхода построена функциональная форма задачи, а алгоритм оптимального регулятора реализован в MATLAB с применением на втором шаге стандартной процедуры `linprog`. Проведен анализ программного решения и реализации обратной связи.

Дальнейшее развитие работы связано с управлением группой в децентрализованном случае, то есть в случае, где каждый объект управления имеет собственный регулятор.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

- 1 Понтрягин Л.С., Болтянский В.Г., Гамкрелидзе Р.В., Мищенко Е.Ф. Математическая теория оптимальных процессов. М.: Физматгиз, 1961. 392 с.
- 2 Беллман, Р. Динамическое программирование / Р. Беллман. – М.:Иностранная литература, 1960. – 400 с.
- 3 Балашевич Н. В., Габасов Р., Кириллова Ф. М. Численные методы программной и позиционной оптимизации линейных систем управления // Журнал вычислительной математики и математической физики. – 2000. – Т. 40, № 6. – С. 838-859.
- 4 Габасов Р., Дмитрук Н.М., Кириллова Ф.М. Оптимальное децентрализованное управление группой динамических объектов// Журнал вычислительной математики и математической физики. – 2008. –Т. 48, № 4. – С. 593-609.
- 5 Дмитрук Н.М., Калинин А.И. Асимптотически субоптимальное управление динамическими системами со слабыми взаимосвязями// Журнал вычислительной математики и математической физики. – 2016. –Т. 56, № 10. – С. 1711–1724.