

Санкт-Петербургский государственный университет
Факультет прикладной математики — процессов управления

Гирдюк Дмитрий Викторович

Магистерская диссертация

**Применение адаптивного метода
в нелинейной управляемой динамической
модели межотраслевого баланса**

Направление 01.04.02 «Прикладная математика и информатика»

Основная образовательная программа магистратуры ВМ.5505
«Математическое и информационное обеспечение
экономической деятельности»

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

Рецензент:
кандидат физ.-мат. наук,
Шахов Я. А.

Санкт-Петербург
2019

Оглавление

Введение	3
Обзор литературы	6
Глава 1. Адаптивный метод в теории оптимального управления	8
1.1. Общая постановка линейной задачи оптимального управления .	8
1.2. Функциональная форма линейной задачи оптимального управле- ния	9
1.3. Адаптивный метод решения интервальных задач линейного про- граммирования	13
1.4. Оптимальное управление в билинейных системах	19
Глава 2. Нелинейная динамическая модель межотраслевого ба- ланса	21
2.1. Статическая модель МОБ	21
2.2. Динамическая модель МОБ	24
2.3. Управляемая динамическая модель МОБ и постановка задач управ- ления	25
Глава 3. Идентификация и применение нелинейной управляемой динамической модели МОБ	28
3.1. Идентификация модели МОБ	28
3.2. Примеры использования модели МОБ	32
Заключение	39
Список литературы	41
Приложение А. Двойственный адаптивный метод	44
Приложение Б. Линеаризация и сведение нелинейной динамиче- ской модели МОБ к ИЗЛП	53

Введение

Адаптивный метод линейного программирования (ЛП) возник [1] в результате анализа классического симплекс-метода. Основная цель обобщения состояла в избавлении от некоторых известных недостатков и создание метода, который можно использовать для решения динамических задач ЛП. Эти задачи принято называть задачами оптимального управления дискретными системами [2].

В наиболее общем случае управляемый объект (тут важно отметить, что управление ищется в виде кусочно-постоянных функций) должен попасть в конечный момент времени в некоторую область, задаваемую системой линейных интервальных неравенств. Известно [3], что линейная задача оптимального управления может быть сведена к интервальной задаче линейного программирования (ИЗЛП). Специфика этих задач такова, что для применения симплекс-метода необходимо введение $4n + 2m$ дополнительных новых переменных и $2(n + m)$ новых основных ограничений для сведения задачи к канонической форме, где m и n – количество ограничений и переменных исходной ИЗЛП. Подобное увеличение размеров задачи значительно сказывается на эффективности симплекс-метода.

В качестве приложения данного метода в работе будет рассмотрена задача оптимального управления параметрами нелинейной динамической модели межотраслевого баланса (МОБ), описанной в работах [4, 5].

Теоретические основы модели «затраты-выпуск» (модель межотраслевого баланса) были заложены в работах нобелевских лауреатов по экономике В. В. Леонтьева и Л. В. Канторовича. В настоящее время МОБ является международно признанным научным инструментом для анализа региональных экономических и социальных систем, а также их макроэкономических тенденций. «Международная ассоциация межотраслевого баланса» (International Input-Output Association, ИОА) [6], объединяющая ученых, занимающихся теорией и практикой применения моделей «затраты-выпуск», активно действует уже порядка 30 лет. ИОА издает собственный научный журнал «Economic System Research» [7].

Статическая модель МОБ представляет собой инструмент, позволяющий охарактеризовать межотраслевые производственные отношения между всеми возможными отраслями региональной экономики. Динамическая модель МОБ,

в отличие от статической, характеризует экономику в долгосрочной перспективе и позволяет руководству страны корректировать цели своего развития в режиме реального времени в зависимости от изменений объемов производства и издержек производства.

Основная цель данной работы состоит в дальнейшем развитии динамической модели МОБ из работ [4, 5]. Для этого необходимо решить ряд задач:

- доработка структуры модели в соответствии с современными стандартами формирования таблиц МОБ;
- разработка плана и программная реализация полной идентификации модели на основе опубликованных мировых таблиц МОБ;
- формирование на основе динамической модели систем управления, позволяющих исследовать и прогнозировать динамику развития экономики при варьировании экзогенных параметров, включенных в модель;
- исследование и программная реализация алгоритмов поиска оптимального управления для рассматриваемых систем управления.

Работа имеет следующую структуру:

- введение,
- обзор использованной литературы,
- 3 главы,
- заключение,
- список библиографических источников,
- два приложения с программной реализацией алгоритмов.

В первой главе представлены необходимые сведения из теории оптимального управления линейными системами, описан алгоритм двойственного адаптивного метода [1, 3, 8], рассмотрен подход к построению оптимального управления в билинейных системах, представленный в работе [9]. Во второй главе представлена доработанная структура модели МОБ из работ [4, 5]: описаны этапы моделирования статической и динамических моделей, а также приведены

примеры систем управления, которые могут быть созданы путем варьирования экзогенных параметров модели. В третьей главе описан алгоритм полной идентификации рассматриваемой модели МОБ на основе таблиц, публикуемых «Организацией экономического сотрудничества и развития» (ОЕСД) [10]; приведены примеры и решены задачи оптимального управления на основе таблиц МОБ США, Индии и Китая.

Обзор литературы

Основные определения и обозначения теории оптимального управления позаимствованы из книги [2], в которой подробно описываются различные вычислительные методы и задачи ОУ.

Первая часть работы посвящена адаптивному методу, применяемому для решения задач оптимального управления в классе кусочно-постоянных управлений путем ее сведения к интервальным задачам линейного программирования, который впервые был подробно изложен в работе [1]. Дальнейшее развитие этого метода и его модификаций описывается в статьях [8, 11, 12].

В книге [3] приводится подробное описание и обоснование адаптивного метода. Помимо этого, в пособии приведено большое количество примеров конкретных прикладных задач, в особенности из экономической области, для которых построены математические модели в виде линейных задач оптимизации.

В данной работе решаются билинейные задачи оптимального программного управления, к которым адаптивный метод напрямую неприменим. Тем не менее, в публикациях [9, 13] предлагается подход к построению управления путем решения последовательности линеаризованных задач, причем их решения равномерно стремятся к решению для билинейной системы.

Вторая часть работы посвящена теории межотраслевого баланса (МОБ), создателем которой является американский экономист российского происхождения, лауреат Нобелевской премии по экономике Василий Васильевич Леонтьев [14]. Самая первая таблица МОБ была создана им для экономики США 1919 года [15]. Развитие теории межотраслевого баланса имеет достаточно долгую историю, описание которой выходит за рамки данной работы. С наиболее подробным описанием всех этапов развития модели можно ознакомиться в работе [16]. Основное внимание при изучении метода стоит обратить на книгу самого Леонтьева [14], в которой он описан в полной мере.

В работе [17] приводится комплексная характеристика существующих на данный момент стандартов построения и работы с таблицами межотраслевого баланса.

Примеры динамических моделей межотраслевого баланса представлены в публикациях [18—22]. Особый интерес вызывает статья [22], в которой аналогично данной работе рассматривается управляемая динамическая модель меж-

отраслевого баланса (в качестве управления выступают затраты на конечное производство). Наиболее подробное описание всех этапов развития динамических моделей МОБ представлено в тринадцатой главе книги [23].

В данной работе рассматривается динамическая модель МОБ, основы которой были заложены в работах [4, 5].

Основным источником информации для идентификации параметров рассматриваемой модели являются национальные таблицы межотраслевого баланса, публикуемые «Организацией экономического сотрудничества и развития» (ОЭСР, OECD) [10]. В качестве альтернативного источника информации для идентификации модели могут быть использованы глобальная база данных цепей поставок проекта «Еога» [24], состоящая из межрегиональной таблицы МОБ, или «Мировая база данных межотраслевого баланса» (World input-output database, WIOD) [25, 26] — проект по созданию мировых таблиц МОБ, профинансированный в 2009 году Евросоюзом. Для исходной рассматриваемой модели идентификация основных параметров на основе таблиц WIOD (для версии 2013 г.) уже была проделана в работе [27].

По промежуточным результатам данной работы были опубликованы три статьи [28—30].

Глава 1. Адаптивный метод в теории оптимального управления

1.1. Общая постановка линейной задачи оптимального управления

Рассмотрим *линейную задачу оптимального управления*:

$$J(u) = c^T x(t^*) \longrightarrow \max, \quad (1.1)$$

$$\dot{x} = A(t)x + B(t)u, \quad x(t_*) = x_0, \quad (1.2)$$

$$g_* \leq Hx(t^*) \leq g^*, \quad (1.3)$$

$$x(t) \in \mathbb{R}^n, \quad u(t) \in \mathbb{R}^r, \quad |u(t)| \leq \tilde{u}, \quad \tilde{u} \in \mathbb{R}^r, \quad t \in T = [t_*, t^*],$$

$$g_*, g^* \in \mathbb{R}^m, \quad \text{rank } H = m < n,$$

где $J(u)$, $u \in U$ есть функционал качества, а $A(t)$, $B(t)$, $t \in T$, представляют $[n \times n]$ - и $[n \times m]$ -матричные функции соответственно. Их элементы являются кусочно-непрерывными при $t \in T$ функциями.

Пусть $h = (t^* - t_*)/N$, $t_* < t^* < +\infty$, N – натуральное число, $T_u = \{t_*, t_* + h, \dots, t^* - h\}$. Функцию $u(t)$, $t \in T$ будем называть *допустимым управлением*, если она кусочно-непрерывная и в каждый момент времени t принимает значения из заданного ограниченного множества U : $u(t) \in U, t \in T$. Управление $u(t) \in U, t \in T$, будем называть *допустимым дискретным управлением*, если $u(t) = u(t_* + kh)$, $t \in [t_* + kh, t_* + (k+1)h)$, $k = \overline{0, N-1}$.

Задача состоит в поиске допустимого управления $u(t)$: $|u(t)| \leq u^* = \text{const}$, максимизирующего линейный функционал качества $J(u)$, $u \in U$, и переводящего систему из начального состояния в момент времени t_* в конечное состояние, задающееся системой линейных интервальных неравенств за время $t^* - t_*$.

1.2. Функциональная форма линейной задачи оптимального управления

Покажем, что рассмотренную в Пункте 1.1 задачу можно свести к интервальной задаче линейного программирования. Для этого будем использовать соображения из работ [1, 8], где это продемонстрировано для случая скалярного управления, т.е. при $r = 1$.

Воспользуемся формулой Коши для решения системы (1.2):

$$x(t, t_*, x_0) = Y(t) \left[Y^{-1}(t_*)x_0 + \int_{t_*}^t Y^{-1}(\tau)B(\tau)u(\tau) d\tau \right],$$

где $Y(t)$ – фундаментальная матрица соответствующей ей однородной системы. В конечный момент времени имеем

$$x(t_*) = x(t^*, t_*, x_0) = Y(t^*) \left[Y^{-1}(t_*)x_0 + \int_{t_*}^{t^*} Y^{-1}(\tau)B(\tau)u(\tau) d\tau \right]. \quad (1.4)$$

Представление (1.4) подставим в функционал качества (1.1)

$$c^T x(t^*) = c^T Y(t^*) \left[Y^{-1}(t_*)x_0 + \int_{t_*}^{t^*} Y^{-1}(\tau)B(\tau)u(\tau) d\tau \right].$$

Ограничимся случаем, когда фундаментальная матрица $Y(t)$ является нормированной в момент времени $t = t_*$, т.е. $Y(t)Y^{-1}(t_*)$ определяется однозначно. Учитывая тот факт, что в последнем представлении функционала первое слагаемое не зависит от управления, то максимизация исходного функционала эквивалентна

$$\int_{t_*}^{t^*} c^T Y(t^*)Y^{-1}(\tau)B(\tau)u(\tau) d\tau \longrightarrow \max. \quad (1.5)$$

Далее рассмотрим сопряженную систему

$$\dot{z} = -A^T z. \quad (1.6)$$

Пусть $Z(t)$ есть ее фундаментальная матрица. Рассмотрим общее решение системы (1.6)

$$z(t) = Z(t)\gamma,$$

где γ – вектор произвольных постоянных, и решим задачу Коши при начальном условии

$$z(t^*) = c,$$

где t^* и c из исходной постановки задачи оптимального управления. Получим

$$z_c(t) = Z(t)Z^{-1}(t^*)c.$$

Одно из основных свойств сопряженных систем состоит в следующем

$$Z^T(t) = Y^{-1}(t).$$

Используя его, нетрудно убедиться, что

$$z_c^T(t) = c^T (Z^{-1}(t^*))^T Z(t)^T = c^T Y(t^*)Y^{-1}(t),$$

и, следовательно, функционал (1.5) принимает вид

$$\int_{t_*}^{t^*} z_c^T(\tau)B(\tau)u(\tau) d\tau \longrightarrow \max. \quad (1.7)$$

Теперь подставим представление (1.4) в ограничения системы (1.3)

$$g_* \leq HY(t^*) \left[Y^{-1}(t_*)x_0 + \int_{t_*}^{t^*} Y^{-1}(\tau)B(\tau)u(\tau) d\tau \right] \leq g^*.$$

Обозначим через $x_0(t)$ вектор, являющийся решением системы (1.2) при нулевом векторе управления $u(t)$. Очевидно, что $x_0(t) = Y(t)Y^{-1}(t_*)x_0$. Тогда последнее условие можно переписать в виде

$$\tilde{g}_* \leq \int_{t_*}^{t^*} HY(t^*)Y^{-1}(\tau)B(\tau)u(\tau) d\tau \leq \tilde{g}^*, \quad (1.8)$$

где $\tilde{g}_* = g_* - Hx_0(t^*)$, $\tilde{g}^* = g^* - Hx_0(t^*)$. Далее обозначим через $h_i, i = \overline{1, m}$, строки матрицы H . Для каждой i -ой строки рассмотрим решение $z_i(t)$ сопряженной системы (1.6) с начальным условием

$$z_i(t^*) = h_i^T.$$

Ее решение имеет вид

$$z_i(t) = Z(t)Z^{-1}(t^*)h_i^T,$$

откуда получаем

$$z_i^T(t) = h_i (Z^{-1}(t^*))^T Z^T(t) = h_i Y(t^*)Y^{-1}(t).$$

Это означает, что z_i^T является i -ой строкой матрицы $HY(t^*)Y^{-1}(t)$ в (1.8). Обозначим через G матрицу со строками z_i^T . Тогда (1.8) можно переписать в виде

$$\tilde{g}_* \leq \int_{t_*}^{t^*} G(\tau)B(\tau)u(\tau) d\tau \leq \tilde{g}^*. \quad (1.9)$$

Таким образом, для построения $z_c(t)$ и $z_i(t), i = \overline{1, m}$ необходимо $m+1$ раз проинтегрировать сопряженную систему (1.6) с начальными условиями $z_c(t^*) = c$ и $z_i(t^*) = h_i^T$ соответственно при обратном отсчете времени t от t^* до t_* .

Далее, учитывая вид допустимого управления, то (1.7) и (1.9) можно записать в виде

$$\sum_{t \in T_u} \int_t^{t+h} z_c^T(\tau)B(\tau) d\tau u(t) \longrightarrow \max, \quad (1.10)$$

$$\tilde{g}_* \leq \sum_{t \in T_u} \int_t^{t+h} G(\tau)B(\tau) d\tau u(t) \leq \tilde{g}^*. \quad (1.11)$$

Понятно, что все интегралы по элементарным промежуткам (1.10), (1.11) можно вычислить заранее. Введем для них обозначения

$$p_j(t) = \int_t^{t+h} z_c^T(\tau)b^j(\tau) d\tau, \quad (1.12)$$

$$q_j(t) = \int_t^{t+h} G(\tau) b^j(\tau) d\tau, \quad (1.13)$$

где b^j есть j -ый столбец матрицы $B(t)$. Кроме того, введем в рассмотрение rN -мерный вектор, состоящий из постоянных управлений

$$V = (u_1(t_*), \dots, u_r(t_*), u_1(t_* + h), \dots, u_r(t_* + h), \dots, u_1(t^* - h), \dots, u_r(t^* - h))^T,$$

а также rN -мерный вектор, состоящий из величин (1.12), вычисленных в соответствующие моменты времени

$$P = (p_1(t_*), \dots, p_r(t_*), p_1(t_* + h), \dots, p_r(t_* + h), \dots, p_1(t^* - h), \dots, p_r(t^* - h))^T,$$

и $[m \times rN]$ -матрица, состоящая из величин (1.13), вычисленных в соответствующие моменты времени

$$Q = (q_1(t_*), \dots, q_r(t_*), q_1(t_* + h), \dots, q_r(t_* + h), \dots, q_1(t^* - h), \dots, q_r(t^* - h))^T.$$

Тогда окончательно имеем эквивалентное представление исходной линейной задачи оптимального управления

$$\begin{aligned} PV &\longrightarrow \max, \\ \tilde{g}_* &\leq QV \leq \tilde{g}^*, \\ |V_j| &\leq \tilde{u}_i, \quad i = j \bmod r. \end{aligned}$$

Подведем итоги. В классе кусочно-постоянных управлений линейную задачу оптимального управления можно свести к задаче линейного программирования. В общем случае, когда ограничения на состояние системы в конечный момент времени задаются системой интервальных неравенств, для построения оптимального программного управления необходимо решить интервальную задачу линейного программирования.

1.3. Адаптивный метод решения интервальных задач линейного программирования

В данной работе для решения интервальных задач линейного программирования применяется адаптивный метод [1, 3]. Последующее изложение есть компактное описание одной из его разновидностей – двойственного адаптивного метода, всецело основанное на работе [3].

Рассмотрим *интервальную задачу линейного программирования*:

$$\begin{aligned}\varphi(x) = c^T x &\longrightarrow \max \\ b_* &\leq Ax \leq b^*, \\ d_* &\leq x \leq d^*,\end{aligned}$$

где c, d_*, d^* есть постоянные n -мерные векторы, b_*, b^* – постоянные m -мерные векторы, A – матрица размерности $m \times n$. Любой n -мерный вектор $x \in X$ называется *планом*, а множество $X = \{x \in \mathbb{R}^n : \text{справедливы (1.3), (1.3)}\}$ – *множеством планов*. Ограничения (1.3) называются *основными*, а (1.3) – *прямыми*. План x^0 называется *оптимальным*, если на нем целевая функция достигает своего максимального значения, т.е.

$$x^0 = \arg \max_{x \in X} c^T x.$$

Кроме того, необходимо ввести следующие обозначения: $I = \{1, \dots, m\}$, $J = \{1, \dots, n\}$, вектор затрат $z = Ax$.

Выберем два множества индексов $I_{\text{оп}} \subseteq I, J_{\text{оп}} \subseteq J, |I_{\text{оп}}| = |J_{\text{оп}}|$. По их паре $K_{\text{оп}} = \{I_{\text{оп}}, J_{\text{оп}}\}$ составим квадратную матрицу $A_{\text{оп}} = A(I_{\text{оп}}, J_{\text{оп}})$. Множество $K_{\text{оп}}$ называется *опорой*, а матрица $A_{\text{оп}}$ – *опорной матрицей*. Пара $\{x, K_{\text{оп}}\}$ называется *опорным планом*.

В соответствии с определением опоры, справедливо следующее равенство

$$z(I_{\text{оп}}) = z_{\text{оп}} = A_{\text{оп}}x_{\text{оп}} + A_{\text{о,н}}x_{\text{н}}, \quad (1.14)$$

где $A_{\text{о,н}} = A(I_{\text{оп}}, J_{\text{н}})$. Ввиду невырожденности опорной матрицы, из этого ра-

венства можно выразить $x_{\text{оп}}$

$$x_{\text{оп}} = A_{\text{оп}}^{-1} z_{\text{оп}} - A_{\text{оп}}^{-1} A_{\text{о,н}} x_{\text{н}}.$$

Таким образом, по своей сути опора есть такая пара подмножеств индексов, при которой для любых компонент $z_{\text{оп}}$ и $x_{\text{н}}$ можно найти такое $x_{\text{оп}}$, при котором на плане x выполняется условие (1.14). В свою очередь, это означает, что на основе опоры всегда есть возможность построить такой план, на котором будет справедлива опорная группа основных ограничений.

По опоре $K_{\text{оп}}$ построим *вектор потенциалов* u

$$\begin{aligned} u &= u(I) = (u(I_{\text{оп}}), 0), \\ u(I_{\text{оп}})^{\text{T}} &= u_{\text{оп}}^{\text{T}} = c_{\text{оп}}^{\text{T}} A_{\text{оп}}^{-1}, \end{aligned} \quad (1.15)$$

и *вектор оценок* Δ

$$\begin{aligned} \Delta &= \Delta(J) = (0, \Delta(J_{\text{н}})), \\ \Delta(J_{\text{н}})^{\text{T}} &= \Delta_{\text{н}}^{\text{T}} = c_{\text{н}}^{\text{T}} - u_{\text{оп}}^{\text{T}} A_{\text{о,н}}, \\ J_{\text{н}} &= J \setminus J_{\text{оп}}, \quad A_{\text{о,н}} = A(I_{\text{оп}}, J_{\text{н}}). \end{aligned} \quad (1.16)$$

Построим с помощью опоры $K_{\text{оп}}$, вектора потенциалов u и вектора оценок Δ псевдоплан \varkappa и вектор псевдозатрат ξ :

$$\begin{aligned} \varkappa_j &= d_{*j}, \text{ если } \Delta_j < 0, \\ \varkappa_j &= d_j^*, \text{ если } \Delta_j > 0, \\ d_{*j} &\leq \varkappa_j \leq d_j^*, \text{ если } \Delta_j = 0, \quad j \in J_{\text{н}}, \\ \varkappa_{\text{оп}} &= A_{\text{оп}}^{-1} (\xi_{\text{оп}} - A_{\text{о,н}} \varkappa_{\text{н}}), \end{aligned} \quad (1.17)$$

$$\begin{aligned} \xi_i &= b_{*i}, \text{ если } u_i < 0, \\ \xi_i &= b_i^*, \text{ если } u_i > 0, \\ b_{*i} &\leq \xi_i \leq b_i^*, \text{ если } u_i = 0, \quad i \in I_{\text{оп}}, \\ \xi_{\text{н}} &= A(I_{\text{н}}, J) \varkappa. \end{aligned} \quad (1.18)$$

Найдем $\beta(x, K_{\text{оп}}) = \max \varphi(x)$ на множестве

$$b_{i*} \leq z_i \leq b_i^*, \quad i \in I_{\text{оп}}; \quad d_{j*} \leq x_j \leq d_j^*, \quad j \in J_{\text{н}}.$$

Можно показать [3], что

$$\beta(x, K_{\text{оп}}) = c'x - c'x^0.$$

Пусть задано число $\varepsilon \geq 0$. План $x^\varepsilon \in X$ называется *субоптимальным* (ε -оптимальным), если на нем выполняется неравенство

$$c'x^0 - c'x^\varepsilon \leq \varepsilon, \quad \text{где } x^0 = \max_{x \in X} c'x.$$

Справедлива следующая теорема [3].

Теорема 1.1 (критерий ε -оптимальности). *Для ε -оптимальности плана x при любом $\varepsilon \geq 0$ необходимо и достаточно существования такой опоры $K_{\text{оп}}$, при которой $\beta(x, K_{\text{оп}}) \leq \varepsilon$.*

Из способа построения псевдоплана x и вектора псевдозатрат ξ , а также Теоремы 1.1 следует, что если для опорных компонент псевдоплана и неопорных компонент вектора псевдозатрат выполнены следующие интервальные неравенства

$$\begin{aligned} d_{*j} &\geq x_j \geq d_j^*, & j \in J_{\text{оп}}, \\ b_{*i} &\geq \xi_i \geq b_i^*, & i \in I_{\text{н}}, \end{aligned} \tag{1.19}$$

то в таком случае x является оптимальным планом.

Рассмотренных результатов достаточно для общего понимания и программной реализации двойственного адаптивного метода. По своей сути метод состоит в построении последовательности опор, не требуя инициализации (в качестве начальной опоры можно взять пустые множества индексов $I_{\text{оп}}$ и $J_{\text{оп}}$). Найдя оптимальную опору, построенный по ней псевдоплан будет являться оптимальным планом решаемой задачи.

Алгоритм двойственного адаптивного метода [3].

На начальном этапе метода необходимо задание начальной опоры $K_{\text{оп}} = \{I_{\text{оп}}, J_{\text{оп}}\}$, которая может быть пустой.

Шаг 1. Вычислить векторы потенциалов и оценок по формулам (1.15), (1.16).

Шаг 2. Построить сопровождающий псевдоплан x и вектор псевдозатрат $\xi = Ax$ по формулам (1.17), (1.18).

Шаг 3. Если выполнен критерий оптимальности (1.19), то закончить работу (в таком случае x является оптимальным планом). Иначе, выбрать индекс $k_* = j_* \in I_{\text{н}}$, на котором критерий не выполняется.

Шаг 4. Построить направления изменения вектора потенциалов Δu и вектора оценок $\Delta \delta$ по правилу симплексной нормировки

- если $k_* = j_* \in J_{\text{оп}}$, то

$$\begin{aligned}\Delta \delta_{j_*} &= \text{sign}(\varkappa_{j_*} - d_{*j_*}), & \Delta \delta(J_{\text{оп}} \setminus j_*) &= 0, & \Delta u(I_{\text{н}}) &= 0, \\ \Delta u(I_{\text{оп}})^{\text{T}} &= -\Delta \delta(J_{\text{оп}})^{\text{T}} A_{\text{оп}}^{-1}, & \Delta \delta(J_{\text{н}}) &= -A_{\text{о,н}}^{\text{T}} \Delta u(I_{\text{оп}}),\end{aligned}$$

- если $k_* = i_* \in I_{\text{н}}$, то

$$\begin{aligned}\Delta u_{i_*} &= \text{sign}(\xi_{i_*} - b_{*i_*}), & \Delta u(I_{\text{н}} \setminus i_*) &= 0, & \Delta \delta(J_{\text{оп}}) &= 0, \\ \Delta u(I_{\text{оп}})^{\text{T}} &= -\Delta u(I_{\text{н}})^{\text{T}} A_{\text{н,о}}^{-1}, & \Delta \delta(J_{\text{н}}) &= -A(I, J_{\text{н}})^{\text{T}} \Delta u(I),\end{aligned}$$

Шаг 5. Вычислить длинный или короткий шаг вдоль построенного направления. Сначала вычисляем шаги σ_k , $k \in J_{\text{н}} \cup I_{\text{оп}}$:

- если $k \in J_{\text{н}}$

$$\sigma_k = \begin{cases} -\Delta_k / \Delta \delta_k & \text{при } \Delta \delta_k < 0 \text{ и } \varkappa_k > d_{*k} \text{ или } \Delta \delta_k > 0 \text{ и } \varkappa_k < d_k^*, \\ \infty, & \text{иначе.} \end{cases}$$

- если $k \in I_{\text{оп}}$

$$\sigma_k = \begin{cases} -u_k / \Delta u_k & \text{при } \Delta u_k < 0 \text{ и } \xi_k > b_{*k} \text{ или } \Delta u_k > 0 \text{ и } \xi_k < b_k^*, \\ \infty, & \text{иначе.} \end{cases}$$

Тогда *короткий шаг* $\sigma^0 = \sigma_{k_0} = \min\{\sigma_k, k \in J_{\text{н}} \cup I_{\text{оп}}\}$.

Альтернативой выступает *длинный шаг*, который находится путем циклического выполнения 5 операций.

Предварительная инициализация: $\sigma^0 = 0$,

$$\alpha = \begin{cases} \varkappa_{j_*} - d_{*j_*} & \text{при } k_* = j_* \in J_{\text{оп}}, \varkappa_{j_*} < d_{*j_*}, \\ d_{j_*}^* - \varkappa_{j_*} & \text{при } k_* = j_* \in J_{\text{оп}}, \varkappa_{j_*} > d_{j_*}^*, \\ \xi_{i_*} - b_{*i_*} & \text{при } k_* = i_* \in I_{\text{н}}, \xi_{i_*} < b_{*i_*}, \\ b_{i_*}^* - \xi_{i_*} & \text{при } k_* = i_* \in I_{\text{н}}, \xi_{i_*} > b_{i_*}^*, \end{cases}$$

1. Положить $\sigma^0 = \sigma_{k_0} = \min\{\sigma_k, k \in J_{\text{н}} \cup I_{\text{оп}}\}$. Если $\sigma^0 = \infty$, выход из цикла.
2. Вычислить

$$\Delta\alpha = \begin{cases} \Delta\delta_k(d_{*k} - \varkappa_k), & \text{при } k \in J_{\text{н}}, \Delta\delta_k < 0, \\ \Delta\delta_k(d_k^* - \varkappa_k), & \text{при } k \in J_{\text{н}}, \Delta\delta_k > 0, \\ \Delta u_k(b_{*k} - \xi_k), & \text{при } k \in J_{\text{н}}, \Delta u_k < 0, \\ \Delta u_k(b_k^* - \xi_k), & \text{при } k \in J_{\text{н}}, \Delta u_k > 0. \end{cases}$$

3. Положить $\alpha = \alpha + \Delta\alpha$.

4. • Если $\alpha \geq 0$, то положить

$$\varkappa_{j_*} = \begin{cases} d_{*j_*} & \text{при } k_* = j_* \in J_{\text{оп}}, \varkappa_{j_*} < d_{*j_*}, \\ d_{j_*}^* & \text{при } k_* = j_* \in J_{\text{оп}}, \varkappa_{j_*} > d_{j_*}^*, \end{cases}$$

$$\xi_{j_*} = \begin{cases} b_{*i_*} & \text{при } k_* = i_* \in I_{\text{н}}, \xi_{j_*} < b_{*i_*}, \\ b_{i_*}^* & \text{при } k_* = i_* \in I_{\text{н}}, \xi_{j_*} > b_{i_*}^*, \end{cases}$$

и выйти из цикла.

- Если $\alpha < 0$, нужно положить

$$\varkappa_k = \begin{cases} d_{*k} & \text{при } k \in J_{\text{н}}, \Delta\delta_k < 0, \\ d_k^* & \text{при } k \in J_{\text{н}}, \Delta\delta_k > 0, \end{cases}$$

$$\xi_k = \begin{cases} b_{*k} & \text{при } k \in J_{\text{н}}, \Delta u_k < 0, \\ b_k^* & \text{при } k \in J_{\text{н}}, \Delta u_k > 0, \end{cases}$$

5. Положить $\sigma_{k_0} = \infty$ и вернуться к началу цикла.

Если $\sigma^0 = \infty$, то необходимо завершить выполнение метода ввиду неограниченности снизу двойственной целевой функции (в этом случае ограничения исходной задачи несовместны).

Шаг 6. В зависимости от того, каким из множеств принадлежат индексы k_0 и k_* , производится замена опоры. Всего возможны 4 случая:

- $k_0 = j_0 \in J_{\text{н}}, k_* = j_* \in J_{\text{оп}}$. В этом случае в опорной матрице $A_{\text{оп}}$ столбец j_* заменяется на столбец j_0 . Элементы обратной матрицы $Q = A_{\text{оп}}^{-1}$ могут

быть пересчитаны по формулам

$$q_{ij} = \begin{cases} q_{ij} - \frac{q_{j_*j}w_i}{p} & \text{при } i \neq j_*, \\ \frac{q_{ij}}{p} & \text{при } i = j_*, \end{cases}$$

где $w = w(J_{\text{оп}}) = QA(I_{\text{оп}}, j_0)$, $p = w_{j_*}$.

- $k_0 = j_0 \in J_{\text{н}}$, $k_* = i_* \in I_{\text{н}}$. Тогда в опорную матрицу добавляется строка i_* и столбец j_0 . Элементы обратной матрицы Q могут быть пересчитаны по формулам

$$q_{ij} = \begin{cases} q_{ij} + \frac{v_i w_j}{p} & \text{при } i, j = \overline{1, n_{\text{оп}}}, \\ -\frac{v_j}{p} & \text{при } i = n_{\text{оп}} + 1, j = \overline{1, n_{\text{оп}}}, \\ -\frac{w_i}{p} & \text{при } i = \overline{1, n_{\text{оп}}}, j = n_{\text{оп}} + 1, \\ \frac{1}{p} & \text{при } i, j = n_{\text{оп}} + 1, \end{cases}$$

где $w = w(J_{\text{оп}}) = QA(I_{\text{оп}}, j_0)$, $v^T = v(I_{\text{оп}})^T = A(i_*, J_{\text{оп}})^T Q$, $p = A(i_*, j_0) - v^T A(I_{\text{оп}}, j_0)$, $n_{\text{оп}} = |I_{\text{оп}}| = |J_{\text{оп}}|$.

- $k_0 = i_0 \in I_{\text{оп}}$, $k_* = j_* \in J_{\text{оп}}$. Тогда из опорной матрицы удаляются строка i_0 и столбец j_* . Формулы пересчета элементов обратной матрицы имеют вид (в предположении, что удаляются последние строка и столбец)

$$q_{ij} = q_{ij} - \frac{q_{in_{\text{оп}}}q_{n_{\text{оп}}j}}{q_{n_{\text{оп}}n_{\text{оп}}}} \text{ при } i, j = \overline{1, n_{\text{оп}}}.$$

- $k_0 = i_0 \in I_{\text{оп}}$, $k_* = i_* \in I_{\text{н}}$. Тогда в опорной матрице строка i_0 заменяется на строку i_* . Формулы пересчета элементов обратной матрицы имеют вид

$$q_{ij} = \begin{cases} q_{ij} - \frac{q_{ii_0}v_j}{p} & \text{при } j \neq i_0, \\ \frac{q_{ij}}{p} & \text{при } j = i_0, \end{cases}$$

где $v^T = v(I_{\text{оп}})^T = A(i_*, J_{\text{оп}})Q$, $p = v_{i_0}$.

Конец итерации, вернуться к Шагу 1.

Интервальную задачу линейного программирования можно решать и стандартным симплекс-методом, однако в таком случае ее придется свести к канонической форме, что в значительной степени увеличит как количество целевых переменных (от n к $4n + 2m$), так и количество основных ограничений (от m к $2n + 2m$) [3].

Программная реализация метода в среде MATLAB 2018b представлена в Приложении А.

1.4. Оптимальное управление в билинейных системах

Далее в работе потребуется строить оптимальные программные управления в билинейных системах вида

$$c^T x(t) \longrightarrow \max_{|u(t)| \leq \tilde{u}}, \quad (1.20)$$

$$\dot{x} = Ax + \sum_{j=1}^n u_j B_j x, \quad (1.21)$$

в которых, очевидно, описанный выше алгоритм напрямую неприменим.

Тем не менее, в работе [9] доказана лемма, позволяющая свести решение задачи с билинейной системой к решению последовательности линейных задач (приводится в соответствии с обозначениями, введенными в Пунктах 1.1, 1.2).

Введем в рассмотрение множество $U^n(T)$ непрерывных n -мерных функций на интервале $T = [t_*, t^*]$.

Лемма 1.1. *Рассмотрим в $U^n(T)$ последовательность n -мерных вектор-функций $\{x^i(t)\}$, являющихся решениями следующих задач*

$$\begin{aligned} \dot{x}^0 &= Ax^0, \\ \dot{x}^i &= Ax^i + \sum_{j=1}^n u_j B_j x^{i-1}, \quad i = 1, 2, \dots \end{aligned} \quad (1.22)$$

с общим начальным условием как и у системы (1.21)

$$x^i(t_*) = x(t_*) = x^0, \quad i = 1, 2, \dots$$

Тогда для $\forall u(t) \in U^r(T)$ последовательность $\{x^i\}$ равномерно сходится на T

к решению системы (1.21)

$$\lim_{i \rightarrow \infty} x^i = x.$$

Авторы работы [13] развили этот результат и доказали, что для системы (1.21) с квадратичным функционалом качества [2]

$$J = \int_0^{\infty} (x^T Q x + u^T R u)$$

последовательное нахождение оптимальных управлений для задач с линейными системами из леммы приводит к их [оптимальных управлений] сходимости к оптимальному управлению для билинейной системы. Но сделать это удалось лишь потому, что оптимальное управление в LR-алгоритме строится в форме обратной связи.

Далее в данной работе потребуется строить оптимальные программные управления для билинейных экономических систем с функционалом качества вида (1.20). Пользуясь описанным в лемме способом и идеями из статьи [13], на каждой итерации при решении задачи с линейной системой из леммы можно находить для нее оптимальное управление, после чего замыкать им систему и находить ее решение. Но, строго говоря, вопрос сходимости оптимальных управлений для линейных систем к оптимальному управлению для билинейной системы остается открытым. Тем не менее, для рассматриваемых задач при учете вида функционала (важно лишь конечное состояние системы), ограничимся этим подходом.

Тут важно отметить, что система (1.21) в матричной форме эквивалентна следующей системе

$$\dot{x} = Ax + [B_1 x | \dots | B_n x] u.$$

Это позволит напрямую применять адаптивный метод при решении последовательности задач из Леммы 1.1.

Глава 2. Нелинейная динамическая модель межотраслевого баланса

2.1. Статическая модель МОБ

Классическая модель Леонтьева (модель «затраты–выпуск», модель межотраслевого баланса) представляет собой макроэкономическую модель, которая отражает потоки промежуточных и конечных товаров и услуг между производителями и потребителями в рамках экономики [14, 15]. Наиболее общий вид представлен на рис. 2.1.

	Сектор 1	...	Сектор N	Конечное потребление			Выпуск
				Потребление	Инвестиции	Экспорт	
Сектор 1							
...							
Сектор N							
Сектор 1, импорт							
...							
Сектор N, импорт							
Чистые налоги на продукцию							
Добавленная стоимость							
Выпуск							

Рис. 2.1. Общая схема таблицы МОБ

Таблица межотраслевого баланса состоит из нескольких блоков.

- *Первый блок* таблицы есть матрица производственной сферы. В ней представлена стоимость межпроизводственных потоков товаров и услуг, произведенных внутри страны и импортированных (в зависимости от вида модели, импорт учитывается в коэффициентах матрицы или сама матрица разбивается на два блока; в данной работе рассматривается последний вариант). Каждый элемент строки z_{ij}^d матрицы определяет производственные затраты, которые i -й сектор, рассматриваемый как производитель, по-

ставляет каждому j -му сектору в год. Аналогично для матрицы импорта Z^m .

- *Вторым блоком* таблицы является матрица затрат на конечное потребление, которая включает конечный спрос на товары и услуги, инвестиции и экспорт.
- *Третий блок* представляет собой строку чистых налогов, т.е. налогов минус субсидии, на продукцию.
- *Четвертый блок* – это строка добавленных стоимостей по каждому сектору экономики.
- Наконец, *пятый и шестой блоки* представляют собой общую сумму по столбцу и строке таблицы и являются затратами на производство и потребление соответственно. Таблицы межотраслевого баланса симметричны в том смысле, что общие затраты на производство товара (выпуски экономики) равны общим затратам на конечное потребление товара.

Перейдем к математической модели таблиц межотраслевого баланса. Тут важно отметить, что она отличается от классической модели Леонтьева, в которой основными переменными выступают выпуски экономики, участием ВВП в качестве $(n + 1)$ -ой переменной. Данная модель является доработанной статической моделью из работ [4, 5]. Балансовые соотношения в относительных ценах имеют вид

$$X_i = a_{i1}X_1 + \dots + a_{in}X_n + Yr_iX_{n+1}, \quad i = 1, \dots, n, \quad (2.1)$$

где $a_{ij} = z_{ij}/X_j$, $z_{ij} = z_{ij}^d + z_{ij}^m$ есть элементы матрицы относительных цен $A = \{a_{ij}\}$, X_{n+1} есть валовый внутренний продукт (ВВП), а $Yr_i = Y_i/X_{n+1}$ являются нормированными по ВВП компоненты вектора конечного потребления. Под ВВП здесь и далее мы подразумеваем сумму добавленных стоимостей VA_j , $j = 1, \dots, n$, созданных в секторах экономики, и налоги минус субсидии (далее, чистые налоги) на продукцию NT_j , $j = 1, \dots, n + 1$

$$GDP = \sum_{j=1}^n VA_j + \sum_{j=1}^{n+1} NT_j, \quad (2.2)$$

Обозначим через W_j затраты на оплату труда, через ONT_j налоги минус субсидии на производство, а через S_j – чистую прибыль. Тогда структура добавленной стоимости VA_j в j -ом секторе экономики выглядит следующим образом

$$VA_j = W_j + ONT_j + S_j, \quad j = 1, \dots, n. \quad (2.3)$$

В дальнейшем мы будем использовать соотношения основных величин с фазовыми переменными X_j

$$rp_j = \frac{X_j - VA_j - NT_j}{X_j}, \quad tp_j = \frac{NT_j}{X_j},$$

$$rw_j = \frac{W_j}{VA_j}, \quad rt_j = \frac{ONT_j}{VA_j},$$

для $j = 1, \dots, n$. Здесь rp_j определяет общую долю внутреннего потребления в годовом объеме производства X_j , tp_j соответствует доли чистых налогов на продукцию в общем выпуске, rw_j – долю добавленной стоимости, затрачиваемой на заработную плату, а rt_j – долю чистых налогов на производство в добавленной стоимости VA_j .

Таким образом, принимая во внимание (2.3), (2.1), мы получаем представление чистой прибыли

$$S_j = (1 - rw_j - rt_j)(1 - rp_j - tp_j)X_j, \quad j = 1, \dots, n. \quad (2.4)$$

Кроме того, пусть tp_{n+1} определяет долю налогов на продукцию в ВВП.

$$tp_{n+1} = NT_{n+1} / \text{GDP}.$$

В таком случае получим $(n + 1)$ -ое балансовое соотношение

$$\begin{aligned} \text{GDP} = X_{n+1} &= \sum_{j=1}^n VA_j + \sum_{j=1}^{n+1} NT_j = \\ &= \sum_{j=1}^n (1 - rp_j - tp_j)X_j + tp_{n+1}X_{n+1}. \end{aligned} \quad (2.5)$$

2.2. Динамическая модель МОБ

Исходная модель основана на двух главных предположениях [4, 5]. *Первое предположение*: количество инвестиций, необходимых для увеличения выпусков (для ускорения производства), пропорционально необходимому ускорению.

$$\varphi_j \dot{X}_j(t) = I_j(t), \quad j = 1, \dots, n, \quad (2.6)$$

где через I_j обозначены инвестиции, а через φ_j – коэффициенты фондоемкости j -го сектора экономики. Фондоемкость измеряет расход капитала на единицу прироста продукции за единицу времени. *Второе предположение*: источником инвестиций является доля чистой прибыли

$$I_j = rn_j S_j, \quad j = 1, \dots, n, \quad (2.7)$$

где через rn_j выражается доля чистой прибыли. Для последней компоненты предположения аналогичны, только в качестве инвестиций рассматривается общая их сумма по секторам

$$I_{n+1} = rn_{n+1} X_{n+1}, \quad I_{n+1} = \sum_{j=1}^n I_j. \quad (2.8)$$

В таком случае, принимая во внимание рассмотренные выше обозначения и основные предположения, получим систему дифференциальных уравнений, описывающих динамику развития всей экономики.

$$\dot{X}_i = \frac{rn_i(1 - rw_i - rt_i)(1 - rp_i - tp_i)}{\varphi_i} \times \left(\sum_{j=1}^n a_{ij} X_j + Y r_i X_{n+1} \right), \quad i = 1, \dots, n. \quad (2.9)$$

$$\dot{X}_{n+1} = \frac{rn_{n+1}}{\varphi_{n+1}} \left(\sum_{j=1}^n (1 - rp_j - tp_j) X_j + tp_{n+1} X_{n+1} \right). \quad (2.10)$$

В векторной форме система выглядит следующим образом

$$\begin{aligned}\dot{X} &= DX, \\ D &= M\tilde{A}.\end{aligned}\tag{2.11}$$

Расширенная матрица относительных цен \tilde{A} и диагональная матрица M имеют следующий вид

$$\tilde{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} & Yr_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & Yr_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n1} & \cdots & a_{nn} & Yr_n \\ 1 - rp_1 & 1 - rp_2 & \cdots & 1 - rp_n & tp_{n+1} \end{pmatrix},$$

$$M = \begin{pmatrix} \frac{\alpha_1}{\varphi_1} & 0 & \cdots & 0 & 0 \\ 0 & \frac{\alpha_2}{\varphi_2} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \frac{\alpha_n}{\varphi_n} & 0 \\ 0 & 0 & \cdots & 0 & \frac{rn_{n+1}}{\varphi_{n+1}} \end{pmatrix},$$

$$\alpha_j = rn_j(1 - rw_j - rt_j)(1 - rp_j - tp_j), \quad j = 1, \dots, n.$$

Модель (2.11) позволяет прогнозировать развитие экономики страны, а также анализировать влияние описанных ранее основных экономических показателей, таких как

$$rp_j, \quad rw_j, \quad rt_j, \quad tp_j, \quad rn_j,$$

на динамику роста экономики.

2.3. Управляемая динамическая модель МОБ и постановка задач управления

Приведенная выше структура динамической модели МОБ важна для понимания возможностей управления динамическими процессами в экономике. Рассмотрим несколько примеров управляемых систем, которые можно получить, введя в систему управление в качестве вариации того или иного парамет-

ра.

Рассмотрим доли затрат на заработную плату в добавленной стоимости rw_j . Обозначим через u_j ее вариацию (управление), удовлетворяющую естественным экономическим ограничениям: $u_{*j} \leq u_j \leq u_j^*$, где u_j^* есть положительная константа. Таким образом доля заработной платы в добавленной стоимости имеет вид $rw_j + u_j$. Тогда система (2.11) принимает вид

$$\dot{X} = DX + \sum_{j=1}^n u_j D_j X, \quad u_* \leq u \leq u^*, \quad (2.12)$$

$$D_j = M_j \tilde{A},$$

$$M_j = \text{diag}(0, \dots, 0, -\beta_j, 0, \dots, 0), \quad \beta_j = \frac{rn_j(1 - rp_j - tp_j)}{\varphi_j}, \quad j = 1, \dots, n.$$

В случае с долей чистых налогов на производство rt_j , $j = 1, \dots, n$ матрица M_j имеет аналогичный вид.

Вариант с долями чистых налогов на продукцию в выпусках tp_j несколько выбивается из общей картины, ведь размерность этого вектора равна $n + 1$. Кроме того, параметр tp_{n+1} присутствует в матрице \tilde{A} . После включения в систему вариаций v_j и выделения слагаемых с управлением, получаем следующую систему

$$\dot{X} = DX + \sum_{j=1}^{n+1} v_j D_j X, \quad v_* \leq v \leq v^*, \quad (2.13)$$

$$D_j = M_j \tilde{A}, \quad j = 1, \dots, n, \quad (2.14)$$

$$M_j = \text{diag}(0, \dots, 0, -\gamma_j, 0, \dots, 0), \quad \gamma_j = \frac{rn_j(1 - rw_j - rt_j)}{\varphi_j}, \quad j = 1, \dots, n. \quad (2.15)$$

$$D_{n+1} = M \tilde{A}_{n+1}, \quad (2.16)$$

$$\tilde{A}_{n+1} = \mathbb{O}_{[(n+1) \times (n+1)]}, \quad \tilde{A}_{n+1}(n+1, n+1) = 1. \quad (2.17)$$

Описанные выше модели позволяет рассматривать различные задачи анализа и управления:

1. *Прогноз макроэкономических трендов в выпусках экономики.* Общий алгоритм решения этой задачи состоит из двух частей. Во-первых, необходимо статистически оценить все основные макроэкономические показатели региона: коэффициенты матрицы расширенных цен \tilde{A} , векторы коэффи-

циентов макроэкономических показателей модели $rp_j, rw_j, rt_j, tp_j, rn_j$ и другие. Конечной целью этого этапа является идентификация элементов матриц D в системе (2.11). Второй – численное интегрирование построенной системы.

2. *Задача программного управления.* Необходимо разработать план для каждого сектора экономики, используя вариации параметров rw_j, rt_j, tp_j . Основная цель может состоять, например, в планировании роста производства или конечного потребления.
3. *Задача оптимального управления.* Пусть дан набор допустимых параметров управления и функционал, который позволяет оценивать различные стратегии управления. В этом случае необходимо найти оптимальную стратегию (как эффективно управлять контролируруемыми параметрами), чтобы максимизировать/минимизировать значение функционала.

Напоследок стоит отметить, что при анализе экономического роста порой бывает нецелесообразно использовать все n экономических секторов. В таких случаях применяют процедуру агрегации, т.е. объединения нескольких секторов в один расширенный агрегированный сектор. Алгоритм объединения тривиален, для рассматриваемой модели представлен в работе [5]. Макроэкономические параметры, рассматриваемые в модели, вычисляются на основе таблицы, полученной после агрегирования.

Глава 3. Идентификация и применение нелинейной управляемой динамической модели МОБ

3.1. Идентификация модели МОБ

На сегодняшний день таблицы межотраслевого баланса публикуются статистическими агентствами многих стран [17], а также мировыми экономическими организациями. Примерами могут служить «Росстат» [31], «Евростат» [32] и «Организация экономического сотрудничества и развития» (ОЭСД) [10]. Кроме того, существуют самостоятельные проекты, занимающиеся созданием баз данных с региональными таблицами МОБ. В качестве примеров могут быть приведены такие проекты, как WIOD [25, 26] и Eora [24], основная задача которых состоит в создании и поддержке мировых таблиц межотраслевого баланса. Первый содержит мировые таблицы МОБ для 28 стран европейского союза, а также 15 главных экономик мира, таких как США, Китай, Россия, Япония и другие за период 2000–2014 гг. по 56 секторам экономики (версия 2016 года). Второй проект еще масштабнее: 190 стран и данные за 1990–2015 гг.

Для идентификации модели в данной работе используются база данных «Организации экономического сотрудничества и развития». Наиболее поздняя версия таблиц («Интернациональный Стандарт Индустриальной Классификации» экономической деятельности, ISIC Rev. 4 [17]) содержит 36 секторов, данные опубликованы за период с 2005 по 2015 гг. для 62 стран. Таблицы, публикуемые этой организацией, имеют наиболее общую структуру по включению импорта в модель. Схематичное представление таблиц совпадает с той, что изображена на рис. 2.1, имеют в базе данных кодификатор DOMIMP.

Второй блок таблицы, представляющий собой блок конечного потребления товаров и услуг, состоит из девяти столбцов (в скобках приведены кодификаторы соответствующих столбцов):

1. Final consumption expenditure of households (HFCE).
2. Final consumption expenditure of non-profit institutions serving households (NPISH).

3. Final consumption expenditure of general government (GGFC).
4. Gross Fixed Capital Formation (GFCF).
5. Changes in inventories (INVNT).
6. Direct purchases abroad by residents (imports) (CONS_ABR).
7. Direct purchases by non-residents (exports) (CONS_NONRES).
8. Exports (EXPO).
9. Imports (IMPO).

Первые три столбца суммарно есть общие затраты на конечное потребление по секторам экономики. Четвертый и пятый столбцы в сумме представляют собой размер инвестиций. Столбцы с шестого по девятый содержат информацию по объемам экспорта/импорта товаров и услуг.

Также стоит отметить, что в таблице чистые налоги на промежуточную и конечную продукцию представлены в двух строках:

1. Taxes less subsidies on intermediate and final products (paid in foreign countries) (TXS_IMP_FNL).
2. Taxes less subsidies on intermediate and final products (paid in domestic territory) (TXS_INT_FNL).

Тут важно подчеркнуть, что при расчете ВВП учитываются лишь сумма чистых налогов на продукцию, уплаченных на территории исследуемой страны.

Кроме того, в базе данных присутствуют отдельные таблицы по структуре добавленных стоимостей (кодификатор таблицы – VAL), содержащих три позиции:

1. Compensation of employees (LABR) — затраты на оплату труда.
2. Other taxes less subsidies on production (OTXS) — чистые налоги на производство.
3. Gross operating surplus and mixed income (GOPS) — чистая прибыль.

В мировой практике принято оценивать добавленную стоимость именно по этим трем показателям [17]. В таблице DOMIMP строка добавленных стоимостей имеет кодификатор VALU.

Перейдем непосредственно к соотнесению параметров рассматриваемой модели и таблиц OECD. По столбцу выпуски X_j могут быть представлены в виде

$$X_j = \sum_{i=1}^{36} (Z_{ij} + Z_{i+36,j}) + TXS_IMP_FNL_j + TXS_INT_FNL_j + VALU_j, \quad (3.1)$$

$$VALU_j = LABR_j + OTXS_j + GOPS_j, \quad j = 1, \dots, 36,$$

где через Z_{ij} обозначены элементы матрицы промежуточных затрат (в том числе и на импорт) из первого блока таблицы размерности 72×36 . По строке продажи X_j имеют вид

$$X_i = \sum_{j=1}^{36} Z_{ij} + Y_i, \quad i = 1, \dots, 36,$$

где Y_j есть сумма i -ых элементов 9 столбцов второго блока таблицы (для $i = 1, \dots, 36$ столбцы CONS_ABR и IMPO нулевые ввиду того, что относятся к импорту). Чтобы идентифицировать параметры статической модели МОБ из Пункта 2.1, добавим и вычтем в последнем выражении промежуточные затраты на импорт

$$X_i = \sum_{j=1}^{36} (Z_{ij} + Z_{i+36,j}) + Y_i - \sum_{j=1}^{36} Z_{i+36,j}, \quad i = 1, \dots, 36, \quad (3.2)$$

Кроме того, проблему вызывают чистые налоги на продукцию TXS_IMP_FNL , уплаченные на территории других государств. Зачастую они включаются в затраты на промежуточное потребление, но в таблицах OECD выделяются отдельной строкой. Самый простой вариант установления баланса в модели состоит в исключении их из рассмотрения как в (3.1), так и вычитание из (3.2).

Принцип расчета ВВП (учитываются чистые налоги на продукцию как

на промежуточное, так и на конечное потребление) состоит в следующем

$$\text{GDP} = \sum_{j=1}^{36} \text{VALU}_j + \sum_{j=1}^{45} \text{TXS_INT_FNL}_j.$$

Таким образом, элементы расширенной матрицы относительных цен a_{ij} представляют собой

$$a_{ij} = \frac{Z_{ij} + Z_{i+36,j}}{\widehat{X}_j}, \quad \widehat{X}_j = X_j - \text{TXS_IMP_FNL}_j, \quad i, j = 1, \dots, 36,$$

$$Yr_i = \frac{\widehat{Y}_i}{\text{GDP}},$$

$$\widehat{Y}_i = Y_i - \sum_{k=1}^{36} Z_{i+36,k} - \text{TXS_IMP_FNL}_i, \quad i = 1, \dots, 36,$$

$$rp_j = \frac{\widehat{X}_j - \text{VALU}_j - \text{TXS_INT_FNL}_j}{\widehat{X}_j}, \quad j = 1, \dots, 36,$$

$$tp_j = \frac{\text{TXS_INT_FNL}_j}{\widehat{X}_j}, \quad j = 1, \dots, 36,$$

$$tp_{\text{GDP}} = \left[\sum_{j=37}^{45} \right] \text{TXS_INT_FNL}_j / \text{GDP}.$$

Формулы для оставшихся двух наборов параметров статической модели (rw, rt) опустим ввиду того, что структура разбиения добавленной стоимости в таблице VAL полностью совпадает со структурой статической модели МОБ из Пункта 2.1.

Последняя часть идентификации относится к параметрам динамической модели МОБ из Пункта 2.2. Как было отмечено выше, инвестиции представлены во втором блоке таблицы двумя столбцами

$$I_i = \text{GFCF}_i + \text{INVNT}_i, \quad i = 1, \dots, 36.$$

Параметры rn_j определяется следующим образом

$$rn_j = \frac{I_i}{\text{GOPS}_j}, \quad i = j, \quad j = 1, \dots, 36, \quad rn_{\text{GDP}} = \left[\sum_{k=1}^{36} I_k \right] / \text{GDP}.$$

Единственным набором параметров модели, который не может быть напрямую определен из статистических данных, являются коэффициенты фондоемкости φ_j , которые могут быть вычислены по формуле [5]

$$\varphi_i(t) = \frac{I_i(t)}{X_i(t+1) - X_i(t)}, \quad i = 1, \dots, 36, \quad (3.3)$$

$$\varphi_{n+1}(t) = \left[\sum_{i=1}^n I_i(t) \right] / [\text{GDP}(t+1) - \text{GDP}(t)]. \quad (3.4)$$

Используя данные из таблиц МОБ за предыдущие года, фондоемкости для текущего года \tilde{t} могут быть определены путем построения регрессионного уравнения и последующего прогнозирования. В работе [27] предлагается использовать в качестве независимой переменной время или разницу между выпусками за текущий и предыдущие года. На текущий момент это самый действенный (и единственный) способ их вычисления.

Все этапы идентификации модели, а также примеры управления различными параметрами были программно реализованы в среде MATLAB 2018b. Результаты применения представлены в следующем пункте данной работы.

3.2. Примеры использования модели МОБ

Рассмотрим несколько приложений рассмотренной модели на примере трех стран: США, Индии и Китая. Для удобства восприятия результатов, будем рассматривать экономику, агрегированную до 3 секторов: промышленность, инфраструктура, жизнеобеспечение.

Пример 1. Результаты прогнозирования выпусков экономики и ее ВВП.

В таблицах 3.1–3.6 представлены результаты прогнозирования на основе динамических моделей, построенных с помощью таблиц МОБ США, Индии и Китая за 2013 и 2014 гг. Через X^{2013} , X^{2014} обозначены реальные значения выпусков и ВВП из таблиц МОБ за 2013 и 2014 гг. соответственно. Через $\tilde{X}_{t_1}^{t_2}$ обозначены результаты прогноза для t_2 года

$$\tilde{X}_{t_1}^{t_2} = \exp(D^{t_1}(t_2 - t_1)) X^{t_1},$$

где D^{t_1} есть матрица из системы (2.11), построенная на основе таблицы МОБ

за t_1 год.

В целом, для достаточно стабильно развивающихся экономических секторов прогноз на год вперед достаточно точен, но, в то же время, существенно деградирует уже при попытке прогнозирования на 2 года вперед.

Таблица 3.1. Результаты прогнозирования выпусков и ВВП на основе таблицы МОБ 2013 г. для экономики США

mln. US \$	X^{2013}	X^{2014}	\tilde{X}_{2013}^{2014} (rel. err.)	X^{2015}	\tilde{X}_{2013}^{2015} (rel. err.)
Ind	7004749	7187317	7124459 (0,875%)	6638631	7247354 (-9,169%)
Is	15655754	16571911	16430935 (0,851%)	17342923	17231917 (0,640%)
Ls	6381374	6552118	6430108 (1,862%)	6817815	6480371 (4,949%)
GDP	16683184	17418612	17207696 (1,21%)	18110876	17749842 (1,993%)

Таблица 3.2. Результаты прогнозирования выпусков и ВВП на основе таблицы МОБ 2014 г. для экономики США

mln. US \$	X^{2014}	X^{2015}	\tilde{X}_{2013}^{2015} (rel. err.)	\tilde{X}_{2014}^{2015} (rel. err.)
Ind	7187317	6638631	7247354 (-9,169%)	7306528 (-10,061%)
Is	16571911	17342923	17231917 (0,640%)	17804032 (-2,659%)
Ls	6552118	6817815	6480371 (4,949%)	6652865 (2,419%)
GDP	17418612	18110876	17749842 (1,993%)	18156309 (-0,251%)

Таблица 3.3. Результаты прогнозирования выпусков и ВВП на основе таблицы МОБ 2013 г. для экономики Индии

mln. US \$	X^{2013}	X^{2014}	\tilde{X}_{2013}^{2014} (rel. err.)	X^{2015}	\tilde{X}_{2013}^{2015} (rel. err.)
Ind	1874414	1895406	1960210 (-3.419%)	1866530	2051009 (-9,883%)
Is	1624300	1733767	1737610 (-0,222%)	1801554	1858427 (-3,157%)
Ls	306296	334938	335974 (-0,309%)	368423	367704 (0,195%)
GDP	1923657	2046100	2056319 (-0,499%)	2115848	2197493 (-3,859%)

Пример 2. Управление долями фондов заработной платы в добавленной стоимости rw_j в экономике США на момент 2013–2014 гг.

В качестве функционала качества будем использовать линейную свертку выпусков экономики и ВВП. Оптимальное управление будем искать в классе кусочно-постоянных функций для $N = 4$, что соответствует поквартальным

Таблица 3.4. Результаты прогнозирования выпусков и ВВП на основе таблицы МОБ 2014 г. для экономики Индии

mln. US \$	X^{2014}	X^{2015}	\tilde{X}_{2013}^{2015} (rel. err.)	\tilde{X}_{2014}^{2015} (rel. err.)
Ind	1895406	1866530	2051009 (-9,883%)	1951572 (-4,556%)
Is	1733767	1801554	1858427 (-3,157%)	1847012 (-2,523%)
Ls	334938	368423	367704 (0,195%)	364182 (1,151%)
GDP	2046100	2115848	2197493 (-3,859%)	2178571 (-2,964%)

Таблица 3.5. Результаты прогнозирования выпусков и ВВП на основе таблицы МОБ 2013 г. для экономики Китая

mln. US \$	X^{2013}	X^{2014}	\tilde{X}_{2013}^{2014} (rel. err.)	X^{2015}	\tilde{X}_{2013}^{2015} (rel. err.)
Ind	13881691	15281290	15179293 (0,667%)	15587327	16607503 (-6,545%)
Is	7811213	8627085	8643300 (-0,188%)	9629793	9563280 (0,691%)
Ls	1457027	1614227	1653756 (-2,449%)	1954423	1871993 (4,218%)
GDP	9607938	10483050	10660471 (-1,692%)	11065288	11822288 (-6,841%)

изменениям. Математическая постановка задачи на основе системы (2.12) имеет следующий вид

$$\begin{aligned}
 cX(t) &\longrightarrow \max_{u_* \leq u \leq u^*}, \\
 \dot{X} &= D^{2013}X + \sum_{j=1}^3 u_j D_j^{2013}X, \\
 t \in T &= [2013; 2014], \quad X(2013) = X^{2013}, \\
 X_* &\leq X(2014) \leq X^*.
 \end{aligned}$$

В случае, когда наблюдается общий рост выпусков секторов экономики и ВВП, фондоемкости по определению принимают положительные значения. Это означает, что при любых весовых коэффициентах функционала (которые вряд ли могут быть положены равными отрицательным значениям) задание широких границ на конечное состояние переменных модели приведет к оптимальному управлению, которое будет определяться своими нижними границами u_* (меньше денег на зарплаты – больше на инвестиции). Поэтому проведение экспериментов с моделью в первую очередь следует начинать с определения желаемых показателей выпусков и ВВП, задаваемых интервалом из векторов X_* , X^* . Кро-

Таблица 3.6. Результаты прогнозирования выпусков и ВВП на основе таблицы МОБ 2014 г. для экономики Китая

mln. US \$	X^{2014}	X^{2015}	\tilde{X}_{2013}^{2015} (rel. err.)	\tilde{X}_{2014}^{2015} (rel. err.)
Ind	15281290	15587327	16607503 (-6,545%)	16595013 (-6,465%)
Is	8627085	9629793	9563280 (0,691%)	9453096 (1,835%)
Ls	1614227	1954423	1871993 (4,218%)	1784213 (8,709%)
GDP	10483050	11065288	11822288 (-6,841%)	11408583 (-3,102%)

ме того, следует учесть и то, что наличие естественных ограничений на управление не позволяет достигнуть сколь угодно больших значений исследуемых переменных.

Пусть для определенности границы на управление задаются 10% от значений параметров rw_j

$$u_* = -0,1 \times rw^{2013}, \quad u^* = 0,1 \times rw^{2013},$$

а все коэффициенты функционала равны единице. На основе информации о собственном движении системы (т.е. изменении переменных при отсутствии управляющего воздействия), определим следующие границы на состояния системы

$$\begin{aligned} \tilde{X}_1^{2014} &\leq X_1(2014) \leq 1,02 \times X_1(2013), \\ \tilde{X}_2^{2014} &\leq X_2(2014) \leq 1,05 \times X_2(2013), \\ \tilde{X}_3^{2014} &\leq X_3(2014) \leq 1,01 \times X_3(2013), \\ \tilde{X}_4^{2014} &\leq X_4(2014) \leq 1,04 \times X_4(2013). \end{aligned}$$

Для построения оптимального управления применяется алгоритм из Пункта 1.4. Программная реализация итерации алгоритма представлена в Приложении Б. Для решения линейных задач оптимального управления применяется адаптивный метод, описанный в Пункте 1.3. Процесс останавливается, когда евклидова норма между конечным состоянием билинейной и текущей линейной системы (обе замкнутые оптимальным управлением для линейной системы) становится меньше 1×10^{-10} .

Оптимальное управление представлено в таблице 3.7, для его получения потребовалось 4 итерации. Результаты управления параметрами представлены

в таблице 3.8. В первых двух столбцах представлены конечные значения состояний системы без и с управляющим воздействием. В третьем столбце представлены конечные значения состояний билинейной системы, замкнутой оптимальным управлением, найденным для линейной системы. В четвертом столбце приведены относительные ошибки между их значениями. Столь малый порядок ошибки является подтверждением справедливости Леммы 1.1.

Таблица 3.7. Оптимальное управление вариациями rw_j в экономике США на момент 2013–2014 гг.

%	rw	1 кв.	2 кв.	3 кв.	4 кв.
u_1	40,16	-4,02	-4,02	-4,02	-4,02
u_2	48,70	-4,87	3,01	3,01	4,87
u_3	81,49	-8,15	-8,15	-8,15	4,18

Таблица 3.8. Результаты управления вариациями rw_j в экономике США на момент 2013–2014 гг.

\tilde{X}^{2014} (rel. to X^{2013})	X_{opt}^{2014} (rel. to X^{2013})	$X_{\text{opt,bl}}^{2014}$	rel. err.
7124459(+1,709%)	7133050(+1,832%)	7133050	$7 \times 10^{-14}\%$
16430935(+4,951%)	16438541(+5,000%)	16438541	$4 \times 10^{-13}\%$
6430108(+0,764%)	6445187(+1,000%)	6445187	$7 \times 10^{-13}\%$
17207696(+3,144%)	17208394(+3,148%)	17208394	$6 \times 10^{-13}\%$

Пример 3. Управление долей налогов на продукцию экономики Индии на момент 2013–2014 гг.

Математическая постановка задачи на основе системы (2.13) выглядит следующим образом

$$\begin{aligned}
 cX(t) &\longrightarrow \max_{v_* \leq v \leq v^*}, \\
 \dot{X} &= D^{2013}X + \sum_{j=1}^4 v_j D_j^{2013}X, \\
 t \in T &= [2013; 2014], \quad X(2013) = X^{2013}, \\
 X_* &\leq X(2014) \leq X^*.
 \end{aligned}$$

Пусть границы на управление задаются самими значениями параметров tp_j . Коэффициенты функционала также равны 1. Границы на конечное состояние

системы для данного примера имеют вид

$$\begin{aligned}\tilde{X}_1^{2014} &\leq X_1(2014) \leq 1,047 \times X_1(2013), \\ \tilde{X}_2^{2014} &\leq X_2(2014) \leq 1,070 \times X_2(2013), \\ \tilde{X}_3^{2014} &\leq X_3(2014) \leq 1,100 \times X_3(2013), \\ \tilde{X}_4^{2014} &\leq X_4(2014) \leq 1,070 \times X_4(2013).\end{aligned}$$

Оптимальное управление представлено в таблице 3.9. Для его нахождения потребовалось 5 итераций. Результаты управления параметрами представлены в таблице 3.10.

Таблица 3.9. Оптимальное управление вариациями tp_j в экономике Индии на момент 2013–2014 гг.

%	tp	1 кв.	2 кв.	3 кв.	4 кв.
v_1	1,50	-1,50	-1,50	-1,50	0,73
v_2	1,61	-1,61	1,61	0,98	1,61
v_3	0,73	-0,73	-0,73	-0,73	-0,73
v_4	4,97	4,97	4,97	1,02	-4,97

Таблица 3.10. Результаты управления вариациями tp_j в экономике Индии на момент 2013–2014 гг.

\tilde{X}^{2014} (rel. to X^{2013})	X_{opt}^{2014} (rel. to X^{2013})	$X_{\text{opt,bl}}^{2014}$	rel. err.
1960210(+4,577%)	1962512(+4,700%)	1962512	$2 \times 10^{-13}\%$
1737610(+6,976%)	1738002(+7,000%)	1738002	$4 \times 10^{-12}\%$
335974(+9,689%)	336305(+9,797%)	336305	$8 \times 10^{-12}\%$
2056319(+6,896%)	2058313(+7,000%)	2058313	$4 \times 10^{-12}\%$

Подведем итоги моделирования. Для относительно стабильно разважающихся экономик динамическая модель позволяет достаточно точно предсказывать значения выпусков и ВВП на год вперед. Качество прогноза может быть улучшено за счет разработки подходов к более точному определению коэффициентов пропорциональности в динамической модели – фондоемкостям.

Программно реализованные алгоритмы позволяют находить оптимальное управление в задачах с линейным функционалом качества и билинейной системой управления. Полученные результаты с практической точки зрения под-

тверждают конструктивность подхода, состоящего в нахождении оптимального управления для билинейной системы путем построения оптимального управления для последовательности линеаризованных систем.

Заключение

В заключении выделим основные результаты проделанной работы:

1. Доработана нелинейная модель межотраслевого баланса из работ [4, 5]. В частности, была переработана структура статической модели в соответствии с текущими стандартами представления национальных таблиц МОБ [17].
2. Рассмотрены варианты формирования управляемой системы на основе данной модели.
3. Описан план и разработан программный комплекс в среде MATLAB 2018b полной идентификации параметров модели на основе таблиц межотраслевого баланса «Организации экономического сотрудничества и развития» [10].
4. Приведены примеры и решены задачи оптимального управления экзогенными параметрами модели при помощи адаптивного метода [1, 8] и методологии линеаризации билинейных систем управления [9, 13]. Программные реализации в среде MATLAB 2018b представлены в Приложении А и Приложении Б соответственно.
5. Результаты работы были представлены на двух конференциях:
 - 49-ая международная научная конференция аспирантов и студентов «Процессы управления и устойчивость» (CPS'18).
Гирдюк Д. В. Оптимальное управление средними ставками оплаты труда на основе нелинейной динамической модели межотраслевого баланса // Процессы управления и устойчивость. 2018. Т. 5, № 1. С. 450–457 [28].
 - 3rd International Conference on Applications in Information Technology (ICAIT'18).
Girdyuk D. V., Smirnov N. V., Smirnova T. E. Optimal Control of the Profit Tax Rate Based on the Nonlinear Dynamic Input-Output Model // Proceedings of the 3rd International Conference on Applications in

Information Technology. Association for Computing Machinery, 2018. С. 80–84. (ACM International Conference Proceeding Series) [29].

А также опубликована статья:

Цифровые технологии в экономике – средство использования теории в практике управления ее развитием / Гирдюк Д. В., Пересада В. П., Смирнов Н. В., Смирнова Т. Е. // Финансы и Бизнес. 2018. Т. 15, № 4. С. 24–36. [30]

В рамках данной работы удалось решить все поставленные задачи. Дальнейшие исследования могут быть направлены на модернизацию подходов к определению и прогнозированию фондоемкостей; совершенствование структуры модели, создание «более детализированного» варианта (например, выделении отдельными позициями налогов и субсидий на продукцию и производство товаров и услуг); поиск новых источников информации для идентификации модели.

Список литературы

1. *Габасов Р., Кириллова Ф. М.* Конструктивные методы оптимизации. Ч. 2.: Задачи управления. Минск : Университетское, 1984.
2. *Болтянский В. Г.* Математические методы теории оптимального управления. Москва : Наука, 1968.
3. *Альсевич В. В., Габасов Р., Глушенко В. С.* Оптимизация линейных экономических моделей: статические задачи. Минск : БГУ, 2000.
4. *Peresada V. P., Smirnov N. V., Smirnova T. E.* Development control of a multicommodity economy based on the dynamical input-output model // Vestnik of St. Petersburg University. Applied Mathematics. Informatics. Control Processes. 2014. № 4. С. 119—132.
5. *Пересада В. П., Смирнов Н. В., Смирнова Т. Е.* Статические и динамические модели многопродуктовой экономики: Учебное пособие. СПб : Издательский Дом Федоровой Г.В., 2017.
6. International Input-Output Association (ИИОА). URL: <http://www.iioa.org/>. (дата обращения: 30.03.2019).
7. Economic Systems Research. The international scholarly journal of the International Input-Output Association (ИИОА). URL: <https://www.tandfonline.com/toc/cesr20/current>. (дата обращения: 30.03.2019).
8. *Балашевич Н. В., Габасов Р., Кириллова Ф. М.* Численные методы программной и позиционной оптимизации линейных систем управления // Журнал вычислительной математики и физики. 2000. Т. 40, № 6. С. 838—859.
9. *Bruni C., Pillo G. D., Koch G.* On the mathematical models of bilinear systems // Ricerche Di Automatica. 1971. Янв. Т. 2(i).
10. Organisation for Economic Co-operation and Development (OECD). URL: <https://data.worldbank.org>. (дата обращения: 30.03.2019).
11. *Балашевич Н. В., Габасов Р., Кириллова Ф. М.* Численные методы программной и позиционной оптимизации кусочно-линейных систем // Журнал вычислительной математики и физики. 2001. Т. 41, № 11. С. 1658—1674.

12. *Балашевич Н. В., Габасов Р., Кириллова Ф. М.* Оптимальное управление нелинейными системами // Журнал вычислительной математики и физики. 2002. Т. 42, № 7. С. 969—995.
13. *Cebuhar W. A., Costanza V.* Approximation procedures for the optimal control of bilinear and nonlinear systems. 1984.
14. *Леонтьев В. В.* Экономическое эссе. Теории, исследования, факты и политика: Пер. с англ. Москва : Политиздат, 1990.
15. *Leontief W. W.* Quantitative Input and Output Relations in the Economic Systems of the United States // The Review of Economics and Statistics. 1936. Т. 18, № 3. С. 105—125.
16. *Meade D. S.* Handbook of Input-Output Analysis // / под ред. Т. Т. Раа. Edward Elgar Publishing Ltd., 2017. Гл. Early days of the input-output table. С. 7—40.
17. Eurostat manual of supply, use and input-output tables. Luxembourg, Office for Official Publications of the European Communities, 2008.
18. *Zhang J. S.* A multi-sector nonlinear dynamic input-output model with human capital // Economic Systems Research. 2008. Т. 20, № 2. С. 223—237.
19. *Hoekstra R., Janssen M. A.* Environmental responsibility and policy in a two-country dynamic input-output model // Economic Systems Research. 2006. Т. 18, № 1. С. 61—84.
20. The uncertainty recovery analysis for interdependent infrastructure systems using the dynamic inoperability input-output model / W. Xu [и др.] // International Journal of Systems Science. 2015. Т. 46, № 7. С. 1299—1306.
21. *Ryaboshlyk V.* A dynamic input-output model with explicit new and old technologies: An application to the UK // Economic Systems Research. 2006. Т. 18, № 2. С. 183—203.
22. *Livesey D. A.* Control theory and input-output analysis // International Journal of Systems Science. 1971. Т. 2, № 3. С. 307—318.
23. *Okuyama Y.* Handbook of Input-Output Analysis // / под ред. Т. Т. Раа. Edward Elgar Publishing Ltd., 2017. Гл. Dynamic input-output analysis. С. 464—484.

24. Eora: multi-region input-output table model. URL: <https://worldmrio.com/>. (дата обращения: 30.03.2019).
25. World input-output database (WIOD). URL: <http://www.wiod.org/>. (дата обращения: 30.03.2019).
26. An Illustrated User Guide to the World Input–Output Database: the Case of Global Automotive Production. / М. Timmer [и др.] // Review of International Economics. 2015. Т. 23.
27. Белоусова М. В., Попков А. С. Построение динамической модели МОБ на основе WIOD // Процессы управления и устойчивость. 2016. Т. 3, № 1. С. 601–606.
28. Гирдюк Д. В. Оптимальное управление средними ставками оплаты труда на основе нелинейной динамической модели межотраслевого баланса // Процессы управления и устойчивость. 2018. Т. 5, № 1. С. 450–457.
29. Girdyuk D. V., Smirnov N. V., Smirnova T. E. Optimal Control of the Profit Tax Rate Based on the Nonlinear Dynamic Input-Output Model // Proceedings of the 3rd International Conference on Applications in Information Technology. Association for Computing Machinery, 2018. С. 80–84. (ACM International Conference Proceeding Series).
30. Цифровые технологии в экономике – средство использования теории в практике управления ее развитием / Д. В. Гирдюк [и др.] // Финансы и Бизнес. 2018. Т. 15, № 4. С. 24–36.
31. Федеральная служба государственной статистики «Росстат». URL: <http://www.gks.ru/>. (дата обращения: 30.03.2019).
32. European Union open data portal (Eurostat). URL: <https://ec.europa.eu/eurostat/data/database>. (дата обращения: 30.03.2019).

Приложение А

Двойственный адаптивный метод

Двойственный адаптивный метод

Программная реализация алгоритма адаптивного двойственного метода

matlab/DualMethodfIP.m

```
1 function [x_opt, cx_opt, K_size_opt, I_opt, J_opt, flag, time, iter] =  
2     ...  
3     DualMethodfIP(c0, A0, bl0, bu0, dl0, du0)  
4  
5 % Dual adaptive method for solving ILPP  
6  
7  
8 tic  
9 global c A m n bl bu dl du eps ztol dtol step  
10  
11 % Coefficients of the objective function  
12 c = c0;  
13 % The main constraints matrix  
14 A = A0;  
15 % Lower bounds of the main constraints  
16 bl = bl0;  
17 % Upper bounds of the main constraints  
18 bu = bu0;  
19 % Lower bounds of the objective variables  
20 dl = dl0;  
21 % Upper bounds of the objective variables  
22 du = du0;  
23  
24  
25  
26 ztol = 1e-12;  
27 dtol = 1e-12;  
28 step = 2;  
29  
30 iter = 0;  
31 max_iter = 10000;  
32  
33 [m,n] = size(A);  
34 K_size = 0;
```

```

31 I = linspace(1,m,m);
32 J = linspace(1,n,n);
33 Q = [];
34
35 % Initialization
36 kappa = du;
37 ksi = A * kappa;
38
39 while iter < max_iter
40     iter = iter + 1;
41     [K_size, I, J, Q, kappa, ksi, flag] = ...
42         DualMethodIter(K_size, I, J, Q, kappa, ksi);
43     switch flag
44         case 1
45             % Optimal solution was found
46             x_opt = kappa;
47             cx_opt = c * kappa;
48             K_size_opt = K_size;
49             I_opt = I;
50             J_opt = J;
51             flag = 1;
52             time = toc;
53             return;
54         case -1
55             % System constraints are inconsistent
56             x_opt = [];
57             cx_opt = [];
58             K_size_opt = [];
59             I_opt = [];
60             J_opt = [];
61             flag = -1;
62             time = toc;
63             return;
64     end
65 end
66
67 % Iter > max_iter. Solution wasn't found
68 x_opt = kappa;
69 cx_opt = c * kappa;
70 K_size_opt = [];
71 I_opt = [];

```

```

72 J_opt = [];
73 flag = -2;
74 time = toc;
75
76 end

```

Программная реализация основной итерации метода

matlab/DualMethodIter.m

```

1 function [K_size_new, I_new, J_new, Q_new, kappa_new, ksi_new, flag] = ...
2     DualMethodIter(K_size, I, J, Q, kappa, ksi)
3
4 global c A m n bl bu dl du dtol step
5
6 % Calculation of u and delta
7 u = zeros(m,1);
8 delta = zeros(n,1);
9 switch K_size
10     case 0
11         delta(J(1:n)) = c';
12     otherwise
13         u(I(1:K_size)) = (c(J(1:K_size)) * Q)';
14         delta(J(K_size + 1:n)) = (c(J(K_size + 1:n)) - u(I(1:K_size)))'
15             * ...
16             A(I(1:K_size), J(K_size + 1:n))';
17
18 end
19
20 % Check of the plan for nondegeneracy
21 isdualnondegenerate = 1;
22 for k = 1 : n
23     if k <= K_size
24         if abs(u(I(k))) <= dtol
25             isdualnondegenerate = 0;
26         end
27     else
28         if abs(delta(J(k))) <= dtol
29             isdualnondegenerate = 0;
30         end
31     end
32 end
33 if ~isdualnondegenerate

```

```

32     isdualnondegenerate
33 end
34
35 % Calculation of kappa and ksi
36 for k = 1 : K_size
37     i = I(k);
38     if u(i) < -dtol || (abs(u(i)) <= dtol && ksi(i) < bl(i))
39         ksi(i) = bl(i);
40     end
41     if u(i) > dtol || (abs(u(i)) <= dtol && ksi(i) > bu(i))
42         ksi(i) = bu(i);
43     end
44 end
45 for k = K_size + 1 : n
46     j = J(k);
47     if delta(j) > dtol || (abs(delta(j)) <= dtol && kappa(j) > du(j))
48         kappa(j) = du(j);
49     end
50     if delta(j) < -dtol || (abs(delta(j)) <= dtol && kappa(j) < dl(j))
51         kappa(j) = dl(j);
52     end
53 end
54
55 kappa(J(1:K_size)) = Q * ...
56     (ksi(I(1:K_size)) - A(I(1:K_size), J(K_size + 1:n))
57     ) * ...
58     kappa(J(K_size + 1:n));
59 ksi(I(K_size + 1:m)) = A(I(K_size + 1:m), J(1:n)) * kappa(J(1:n));
60
61 % Verification of the optimal condition
62 [isopt, errmax, k0] = IsDualOptimal(K_size, I, J, kappa, ksi);
63
64 if isopt
65     kappa_new = kappa;
66     ksi_new = ksi;
67     Q_new = Q;
68     K_size_new = K_size;
69     I_new = I;
70     J_new = J;
71     flag = 1;
72     return;

```

```

72     end
73
74     % Calculation of delta_u and delta_delta
75     delta_u = zeros(m,1);
76     delta_delta = zeros(n,1);
77     if k0 <= K_size
78         j = J(k0);
79         delta_delta(j) = sign(kappa(j) - dl(j));
80     else
81         i = I(k0);
82         delta_u(i) = sign(ksi(i) - bl(i));
83     end
84     for i = 1 : K_size
85         tmp(i) = -delta_delta(J(i));
86         for k = K_size + 1 : m
87             tmp(i) = tmp(i) - A(I(k),J(i)) * delta_u(I(k));
88         end
89     end
90     for k = 1 : K_size
91         i = I(k);
92         delta_u(i) = 0.0;
93         for j = 1 : K_size
94             delta_u(i) = delta_u(i) + Q(j,k) * tmp(j);
95         end
96     end
97     for k = K_size + 1 : n
98         j = J(k);
99         delta_delta(j) = 0.0;
100        for i = 1 : m
101            delta_delta(j) = delta_delta(j) - A(i,j) * delta_u(i);
102        end
103    end
104
105
106    % Calculation of the dual step sigma_min (if step = 1 — short step
107    % else long step)
108    for k = 1 : n
109        sigma(k) = inf;
110        if k <= K_size
111            i = I(k);
112            if delta_u(i) > dtol && ksi(i) < bu(i)

```



```

113         sigma(k) = -u(i) / delta_u(i);
114     end
115     if delta_u(i) < -dtol && ksi(i) > bl(i)
116         sigma(k) = -u(i) / delta_u(i);
117     end
118     else
119         j = J(k);
120         if delta_delta(j) < -dtol && kappa(j) > dl(j)
121             sigma(k) = -delta(j) / delta_delta(j);
122         end
123         if delta_delta(j) > dtol && kappa(j) < du(i)
124             sigma(k) = -delta(j) / delta_delta(j);
125         end
126     end
127     if sigma(k) < dtol
128         sigma(k) = 0.0;
129     end
130 end
131 alpha = -errmax;
132
133 while alpha < 0
134     sigma_min = inf;
135     [sigma_min, ks] = min(sigma);
136     switch step
137         case 1
138             alpha = 0;
139         case 2
140             sigma(ks) = inf;
141             if ks <= K_size
142                 i = I(ks);
143                 if delta_u(i) > dtol
144                     dalpha = (bu(i) - ksi(i)) * delta_u(i);
145                     ksi(i) = bu(i);
146                 end
147                 if delta_u(i) < -dtol
148                     dalpha = (bl(i) - ksi(i)) * delta_u(i);
149                     ksi(i) = bl(i);
150                 end
151             else
152                 j = J(ks);
153                 if delta_delta(j) > dtol

```

```

154         dalpha = (du(j) - kappa(j)) * delta_delta(j);
155         kappa(j) = du(j);
156     end
157     if delta_delta(j) < -dtol
158         dalpha = (dl(j) - kappa(j)) * delta_delta(j);
159         kappa(j) = dl(j);
160     end
161 end
162     alpha = alpha + dalpha;
163 end
164 end
165 if sigma_min == inf
166     kappa_new = [];
167     ksi_new = [];
168     Q_new = [];
169     K_size_new = [];
170     I_new = [];
171     J_new = [];
172     flag = -1;
173 end
174
175 % Replacement of the Support set
176 if k0 > K_size
177     i0 = I(k0);
178     if ks > K_size
179         js = J(ks);
180         % Adding row i0 and column js
181         K_size = K_size + 1;
182         I([K_size, k0]) = I([k0, K_size]);
183         J([K_size, ks]) = J([ks, K_size]);
184         Q = inv(A(I(1:K_size), J(1:K_size)));
185     else
186         is = I(ks);
187         % Replacing the string is on i0
188         I([ks, k0]) = I([k0, ks]);
189         Q = inv(A(I(1:K_size), J(1:K_size)));
190     end
191 else
192     j0 = J(k0);
193     if ks > K_size
194         js = J(ks);

```

```

195     % Replacing column j0 with js
196     J([k0, ks]) = J([ks, k0]);
197     Q = inv(A(I(1:K_size), J(1:K_size)));
198     else
199         is = I(ks);
200         % Deleting row is and column j0
201         I([ks, K_size]) = I([K_size, ks]);
202         J([k0, K_size]) = J([K_size, k0]);
203         K_size = K_size - 1;
204         Q = inv(A(I(1:K_size), J(1:K_size)));
205     end
206 end
207
208 kappa_new = kappa;
209 ksi_new = ksi;
210 Q_new = Q;
211 K_size_new = K_size;
212 I_new = I;
213 J_new = J;
214 flag = 0;

```

Программная реализация проверки условия оптимальности

matlab/IsDualOptimal.m

```

1 function [isopt, errmax, k0] = IsDualOptimal(K_size, I, J, kappa, ksi)
2
3 global m bl bu dl du dtol
4
5 errmax = dtol;
6 k0 = [];
7 for k = 1 : m
8     if k <= K_size
9         j = J(k);
10        if dl(j) - kappa(j) > errmax
11            errmax = dl(j) - kappa(j);
12            k0 = k;
13        end
14        if kappa(j) - du(j) > errmax
15            errmax = kappa(j) - du(j);
16            k0 = k;
17        end

```

```
18     else
19         i = I(k);
20         if bl(i) - ksi(i) > errmax
21             errmax = bl(i) - ksi(i);
22             k0 = k;
23         end
24         if ksi(i) - bu(i) > errmax
25             errmax = ksi(i) - bu(i);
26             k0 = k;
27         end
28     end
29 end
30
31 if isempty(k0)
32     isopt = 1;
33 else
34     isopt = 0;
35 end
```

Приложение Б

Линеаризация и сведение нелинейной динамической модели МОБ к ИЗЛП

Итерация алгоритма нахождения оптимального управления для билинейной системы, описанного в Пункте 1.4

matlab/ReductionToILPP.m

```
1 function [c,A,bl,bu,dl,du] = ReductionToILPP(c0,H,gl,gu,ul,uu,...
2                                     A0,t0,x0,tn,N,...
3                                     rn,rp,tp,rw,rt,Fe,...
4                                     v_opt,Aa,mode)
5 % Iteration of the algorithm which finds the optimal control for
6 % bilinear control system by solving the sequence of linear problems,
7 % using the solution (optimal control for linear problem) from
8 % previous step.
9
10 % c0 — coefficients of the functional
11 % H — matrix of the main constraints of the system variables
12 % gl and gu — lower and upper bounds for main constraints
13 % ul and uu — lower and upper bounds for the control variables
14 % A0 — matrix of the system
15 % t0 and tn — bounds of the time interval
16 % x0 — initial state of the system
17 % N — amount of discretization steps
18 % rn, rp, tp, rw, rt, Fe — parameters of the system
19 % v_opt — optimal control from the previous iteration
20 % Aa — matrix A of the model
21 % mode — which parameter is used for control
22 % 1 — rw or rt, 2 — tp(1:end - 1)
23
24 % Size of the control vector
25 r = size(ul,2);
26 % Number of sectors
27 n = size(x0,1);
28 % Descretization step
29 step = (tn - t0) / N;
30
31 % Fundamental matrix at the final moment of time
```

```

32 Yn = expm(A0 * (tn - t0));
33 % Fundamental matrix of the adjoint system at the initial time
34 Z0 = eye(size(A0));
35
36 P = [];
37 Q = [];
38 Vl = [];
39 Vu = [];
40
41 % Calculation of the coefficients of the objective function and
42 % main constraints matrix of ILPP. The Simpson formula is used
43 % for numerical integration.
44 X_opt_b = x0;
45 AaInit = Aa;
46
47 for j = 0 : N - 1
48     YnZaT = Yn * expm(-A0' * (j * step))';
49     YnZcT = Yn * expm(-A0' * ((j + 0.5) * step))';
50     YnZbT = Yn * expm(-A0' * ((j + 1) * step))';
51
52     M = zeros(size(x0,1));
53     % The use of optimal control v_opt for linearized problem
54     % obtained from the previous step
55     switch mode
56         case 1
57             for i = 1 : n - 1
58                 M(i,i) = rn(i) * (1 - rp(i) - tp(i)) *...
59                     (1 - rw(i) - rt(i) - v_opt(j * r + i)) /...
60                     Fe(i);
61             end
62         case 2
63             for i = 1 : n - 1
64                 M(i,i) = rn(i) *...
65                     (1 - rp(i) - tp(i) - v_opt(j * r + i)) *...
66                     (1 - rw(i) - rt(i)) /...
67                     Fe(i);
68             end
69
70     %         Aa(r,1:end - 1) = AaInit(r,1:end - 1) -...
71     %             v_opt(j * r + 1:j * r + n - 1)';
72     Aa(end,end) = AaInit(end,end) +...

```

```

73         v_opt((j + 1) * r);
74     end
75
76     M(end, end) = rn(end) / Fe(end);
77
78     X_opt_a = X_opt_b;
79     X_opt_c = expm(M * Aa * (0.5 * step)) * X_opt_b;
80     X_opt_b = expm(M * Aa * step) * X_opt_b;
81
82     switch mode
83     case 1
84         beta = -rn(1:end - 1) .* (1 - rp - tp(1:end - 1)) ./ ...
85             Fe(1:end - 1);
86         B0a = [];
87         B0c = [];
88         B0b = [];
89         for k = 1 : n - 1
90             B0a = [B0a, ...
91                 diag([zeros(1, k - 1), ...
92                     beta(k), ...
93                     zeros(1, n - k)]) * Aa * X_opt_a
94                 ];
95             B0c = [B0c, ...
96                 diag([zeros(1, k - 1), ...
97                     beta(k), ...
98                     zeros(1, n - k)]) * Aa * X_opt_c
99                 ];
100            B0b = [B0b, ...
101                diag([zeros(1, k - 1), ...
102                    beta(k), ...
103                    zeros(1, n - k)]) * Aa * X_opt_b
104                ];
105        end
106        B0a = [B0a, zeros(n, 1)];
107        B0c = [B0c, zeros(n, 1)];
108        B0b = [B0b, zeros(n, 1)];
109    case 2
110        gamma = -rn(1:end - 1) .* ...
111            (1 - rw - rt) ./ Fe(1:end - 1);
112        B0a = [];
113        B0c = [];

```

```

114         B0b = [];
115         for k = 1 : n - 1
116             Aa1 = zeros(n);
117             %           Aa1(n,k) = -rn(end) / Fe(end);
118             B0a = [B0a, ...
119                 (diag([zeros(1,k - 1), ...
120                     gamma(k), ...
121                     zeros(1,n - k)]) * Aa + Aa1) * X_opt_a
122                 ];
123             B0c = [B0c, ...
124                 (diag([zeros(1,k - 1), ...
125                     gamma(k), ...
126                     zeros(1,n - k)]) * Aa + Aa1) * X_opt_c
127                 ];
128             B0b = [B0b, ...
129                 (diag([zeros(1,k - 1), ...
130                     gamma(k), ...
131                     zeros(1,n - k)]) * Aa + Aa1) * X_opt_b
132                 ];
133         end
134         Aa1 = zeros(n);
135         Aa1(n,n) = rn(end) / Fe(end);
136         B0a = [B0a, Aa1 * X_opt_a];
137         B0c = [B0c, Aa1 * X_opt_c];
138         B0b = [B0b, Aa1 * X_opt_b];
139     end
140
141     for k = 1 : r
142         Pa = c0 * YnZaT * B0a(:,k);
143         Pc = c0 * YnZcT * B0c(:,k);
144         Pb = c0 * YnZbT * B0b(:,k);
145         P(j * r + k) = step / 6 * (Pa + 4 * Pc + Pb);
146         Qa = H * YnZaT * B0a(:,k);
147         Qc = H * YnZcT * B0c(:,k);
148         Qb = H * YnZbT * B0b(:,k);
149         Q(:,j * r + k) = step / 6 * (Qa + 4 * Qc + Qb);
150         Vl(j * r + k) = ul(k);
151         Vu(j * r + k) = uu(k);
152     end
153 end
154

```



```
155 % Calculation of boundaries for basic constraints
156 g0l = gl - H * Yn * Z0' * x0;
157 g0u = gu - H * Yn * Z0' * x0;
158
159 c = double(P);
160 A = double(Q);
161 bl = double(g0l);
162 bu = double(g0u);
163 dl = Vl';
164 du = Vu';
165
166 end
```