

ВЕСТНИК
ТОМСКОГО
ГОСУДАРСТВЕННОГО
УНИВЕРСИТЕТА

УПРАВЛЕНИЕ,
ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА
И ИНФОРМАТИКА

TOMSK STATE UNIVERSITY
JOURNAL OF CONTROL AND COMPUTER SCIENCE

Научный журнал

2017

№ 40

Зарегистрирован в Федеральной службе по надзору в сфере
массовых коммуникаций, связи и охраны культурного наследия
(свидетельство о регистрации ПИ № ФС 77-29497 от 27 сентября 2007 г.)

Подписной индекс в объединённом каталоге «Пресса России» 44031

Журнал включен в «Перечень рецензируемых научных изданий,
в которых должны быть опубликованы основные научные результаты диссертаций
на соискание ученой степени кандидата наук, на соискание ученой степени доктора наук»,
Высшей аттестационной комиссии

Founder – Tomsk State University

EDITORIAL BOARD

Alexander Gortsev – Editor-in-Chief, Doctor of Sc., Prof., Head of the Operation Research Department, Dean of the Faculty of Applied Mathematics and Cybernetics Tomsk State University. Tel: +72822529599

Valery Smagin – Deputy Editor-in-Chief, Doctor of Sc., Prof. of the Applied Mathematics Department Tomsk State University. Tel: +72822529599

Lyudmila Nezhelskaya – Executive Editor, Cand. of Sc., Docent of the Operation Research Department.
E-mail: vestnik_uvti@mail.tsu.ru

Sergey Vorobeychikov – Doctor of Sc., Prof. of the Higher Mathematics and Mathematical Modeling Department Tomsk State University

Vladimir Vishnevsky – Doctor of Sc., Prof. Head of the laboratory Institute of Control Sciences of Russian Academy of Sciences (Moscow, Russia).

Gennady Koshkin – Doctor of Sc., Prof. of the Theoretical Cybernetics Department Tomsk State University

Yury Kostyuk – Doctor of Sc., Prof. of the Theoretical Informatics Department Tomsk State University

Anjela Matrosova – Doctor of Sc., Prof., Head of the Programming Department Tomsk State University

Anatoly Nazarov – Doctor of Sc., Prof., Head of the Probability Theory and Mathematical Statistics Department Tomsk State University

Konstantin Samouylov – Doctor of Sc., Prof., Head of the Applied Probability and Informatics Department RUDN University (Moscow, Russia)

Eugene Semenko – Doctor of Sc., Prof. System Analysis and Operations Research Department Reshetnev Siberian State Aerospace University (Krasnoyarsk, Russia)

Sergey Sushchenko – Doctor of Sc., Prof., Head of the Applied of Information Department, Dean of the Faculty of Informatics Tomsk State University

Mais Farkhadov – Doctor of Sc., Head of the laboratory Institute of Control Sciences of Russian Academy of Sciences (Moscow, Russia).

Gurami Tsitsiashvili – Doctor of Sc., Prof., Chief researcher Institute for Applied Mathematics Far Eastern Branch of RAS, Prof. Far Eastern Federal University (Vladivostok, Russia)

Editorial address:

Faculty of applied mathematics and cybernetics

National Research Tomsk State University

36 Lenina Avenue, Tomsk, 634050

Telephone / fax: +72822529599

E-mail: vestnik_uvti@mail.tsu.ru

Учредитель – Томский государственный университет

РЕДАКЦИОННАЯ КОЛЛЕГИЯ

Горцев Александр Михайлович – гл. редактор, проф., д-р техн. наук, зав. кафедрой исследования операций, декан факультета прикладной математики и кибернетики ТГУ. Тел. +72822529599

Смагин Валерий Иванович – зам. гл. редактора, проф., д-р техн. наук, проф. кафедры прикладной математики ТГУ. Тел. +72822529599

Нежельская Людмила Алексеевна – ответственный секретарь, доц., канд. техн. наук, доц. кафедры исследования операций ТГУ
E-mail: vestnik_uvti@mail.tsu.ru

Воробейчиков Сергей Эрикович – проф., д-р физ.-мат. наук, проф. кафедры высшей математики и математического моделирования ТГУ

Вишневский Владимир Миронович – проф., д-р техн. наук, зав. лабораторией Института проблем управления РАН (г. Москва)

Кошкин Геннадий Михайлович – проф., д-р физ.-мат. наук, проф. кафедры теоретической кибернетики ТГУ

Костюк Юрий Леонидович – проф., д-р техн. наук, проф. кафедры теоретической информатики ТГУ

Матросова Анжела Юрьевна – проф., д-р техн. наук, зав. кафедрой программирования ТГУ

Назаров Анатолий Андреевич – проф., д-р техн. наук, зав. кафедрой теории вероятностей и математической статистики ТГУ

Самуйлов Константин Евгеньевич – проф., д-р техн. наук, зав. кафедрой прикладной информатики и теории вероятностей РУДН (г. Москва)

Семенкин Евгений Станиславович – проф., д.т.н., проф. каф. системного анализа и исследования операций, СГАУ им. акад. М.Ф. Решетнева (г. Красноярск)

Сусенко Сергей Петрович – проф., д-р техн. наук, зав. кафедрой прикладной информатики, декан факультета информатики ТГУ

Фархадов Манс Паша Оглы – д-р техн. наук, зав. лабораторией Института проблем управления РАН (г. Москва)

Цициашвили Гурами Шалвович – проф., д-р физ.-мат. наук, гл. науч. сотр. Института прикладной математики ДВО РАН, проф. ДВФУ (г. Владивосток)

Адрес редакции и издателя: 634050, Томск, пр. Ленина, 36

Национальный исследовательский Томский государственный университет, факультет прикладной математики и кибернетики

Телефон / факс: +72822529599

E-mail: vestnik_uvti@mail.tsu.ru

EDITORIAL COUNCIL

Ana Rosa Cavalli

PhD, Prof.
University VII
Paris, France

Vladimir Dombrovskii

Doctor of Sc., Prof.
Tomsk State University
Russia

Alexander Dudin

Doctor of Sc., Prof.
Belarusian State University
Minsk, Republic Belorussia

Enco Orsingher

PhD, Prof.
University of Rome
Italy

Paolo Prinetto

Prof.
Politechnic Institute
Torino, Italy

Gilbert Saporta

PhD, Prof.
Pierre and Marie Curie
University, Paris, France

Raimund Ubar

Doctor of Sc., Prof.
University of Technology
Tallinn, Estonia

Reindert Nobel

Doctor of Sc., Associate Prof.
Vrije University of Amsterdam
Netherlands

Nina Yevtushenko

Doctor of Sc., Prof.
Tomsk State University
Russia

Yervant Zorian

PhD, Fellow & Chief Architect, Synopsys, Mountain View, CA, USA

РЕДАКЦИОННЫЙ СОВЕТ

Ана Роза Кавалли

д-р философии, проф.
Университет VII,
Париж, Франция
Владимир Домбровский
д-р техн. наук, проф.
ТГУ, Томск, Россия

Александр Дудин

д-р физ.-мат. наук, проф.
БГУ, Минск,
Республика Беларусь

Енцо Орзингер

д-р философии, проф.
Римский университет,
Италия

Паоло Принетто

проф.
Политехнический институт,
Турин, Италия

Жильберт Сапорта

д-р философии, проф.
Университет им. Пьера и Марии,
Кюри, Париж, Франция
Раймонд Убар
проф.
Технологический университет,
Таллинн, Эстония

Рейндерт Нобель

д-р, доцент
Свободный университет,
Амстердам, Нидерланды

Нина Евтушенко

д-р техн. наук, проф.
ТГУ, Томск, Россия

Ервант Зориан

д-р философии, гл. науч. сотр.
фирмы «Синopsis», США

JORNAL INFO

Tomsk State University Journal of Control and Computer Science is an independent peer-reviewed research journal that welcomes submissions from across the world.

Tomsk State University Journal of Control and Computer Science is issued four times per year, and can be subscribed to in the Russian Press Joint Catalogue (Subscription Index 44031)

The publication in the journal is free of charge and may be in Russian or in English.

The topics of the journal are the following:

- *control of dynamical systems,*
- *mathematical modeling,*
- *data processing,*
- *informatics and programming,*
- *discrete function and automation,*
- *designing and diagnostics of computer systems.*

Rules of registration articles are given in a site:

<http://journals.tsu.ru/informatics/>

ISSN 2311-2085 (Online), ISSN 1998-8605 (Print).

О ЖУРНАЛЕ

Журнал «Вестник Томского государственного университета. Управление, вычислительная техника и информатика» выходит ежеквартально и распространяется по подписке

Статьи публикуются на русском и английском языках.

Тематика публикаций журнала:

- *управление динамическими системами,*
- *математическое моделирование,*
- *обработка информации,*
- *информатика и программирование,*
- *дискретные функции и автоматы,*
- *проектирование и диагностика вычислительных систем.*

Журнал входит в систему Российского Индекса Научного Цитирования (РИНЦ).

Правила оформления статей приведены на сайте:

<http://journals.tsu.ru/informatics/>

ISSN 2311-2085 (Online), ISSN 1998-8605 (Print).

СОДЕРЖАНИЕ

УПРАВЛЕНИЕ ДИНАМИЧЕСКИМИ СИСТЕМАМИ

Домбровский В.В., Обедко Т.Ю. Управление с прогнозирующей моделью стохастическими системами с марковскими скачками и сериально коррелированными параметрами при ограничениях	4
Мансимов К.Б., Расулова Ш.М. Об одной задаче управления типа А.И. Москаленко	12
Смагин В.И. Адаптивное прогнозирующее управление в дискретных системах с неизвестным входом	23

ОБРАБОТКА ИНФОРМАЦИИ

Горцев А.М., Завгородняя М.Е. Оценивание параметра непродлевающегося мертвого времени случайной длительности в пуассоновском потоке событий	32
Медведев Г.А. Модель временной структуры доходности Лонгстаффа–Шварца и ее расширение	41
Тырсин А.Н., Костин К.К. Оценивание логистической регрессии как экстремальная задача	52

ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ

Лукин Н.А., Филимонов А.Ю. Технологии программирования алгоритмов на архитектурах однородных вычислительных сред	61
Сонькин М.А., Моисеев А.Н., Сонькин Д.М., Буртова Д.А. Объектная модель приложения для имитационного моделирования циклических систем массового обслуживания	71
Павликов М.К. Алгоритм распределения нагрузки в программной системе, построенной на основе протокола HDP	81
Сведения об авторах	85

CONTENTS

CONTROL OF DYNAMICAL SYSTEMS

Dombrovskii V.V., Obedko T.Y. Model predictive control for stochastic systems with Markovian jumps and serially correlated parameters under constraints	4
Mansimov K.B., Rasulova S.M. About one A.I. Moskalenko control problem	12
Smagin V.I. Adaptive model predictive control discrete systems with unknown input	23

DATA PROCESSING

Gortsev A.M., Zavgorodnyaya M.E. Estimation of the parameter of unextendable dead time random duration in the Poisson flow of events	32
Medvedev G.A. The Longstaff-Schwartz model of yield term structure and its expansion	41
Tyrsin A.N., Kostin K.K. Consideration of estimation of logistic regression as an optimization problem	52

INFORMATICS AND PROGRAMMING

Lookin N.A., Filimonov A.Yu. Software technologies for homogenous computing environment	61
Sonkin M.A., Moiseev A.N., Sonkin D.M., Burtovaya D.A. Object model of application for simulation of cyclic queueing systems	71
Pavlikov M.K. Load balancing algorithm in a distributed programming system, built on the basis of the HDP protocol	81
Brief information about the authors	85

УПРАВЛЕНИЕ ДИНАМИЧЕСКИМИ СИСТЕМАМИ

УДК 519.865.5

DOI: 10.17223/19988605/40/1

В.В. Домбровский, Т.Ю. Обьедко

УПРАВЛЕНИЕ С ПРОГНОЗИРУЮЩЕЙ МОДЕЛЬЮ СТОХАСТИЧЕСКИМИ СИСТЕМАМИ С МАРКОВСКИМИ СКАЧКАМИ И СЕРИАЛЬНО КОРРЕЛИРОВАННЫМИ ПАРАМЕТРАМИ ПРИ ОГРАНИЧЕНИЯХ

Рассматривается задача управления с прогнозированием по квадратичному критерию для линейных дискретных систем с марковскими скачками и сериально коррелированными параметрами. Синтезированы стратегии управления при наличии явных ограничений на управляющие воздействия. Алгоритм синтеза прогнозирующей стратегии сводится к решению последовательности задач квадратичного программирования.

Ключевые слова: стохастические системы; марковские скачки; зависимые параметры; прогнозирующее управление; ограничения.

Моделями с марковскими скачкообразными параметрами описывается широкий класс реальных систем [1]. Примерами могут служить сложные производственно-технологические, энергетические и технические системы. Гибридные системы с марковским режимом переключений также широко используются в финансовой инженерии для описания поведения инвестиционного портфеля на финансовом рынке с переключающимися режимами [2].

В таких моделях предполагается, что смена структуры системы осуществляется в соответствии с эволюцией марковской цепи с конечным пространством состояний. Решению различных задач управления и оценивания для таких систем посвящено значительное количество работ [3–9].

Эффективным подходом к синтезу систем управления с ограничениями, получившим широкое признание и применение в практике управления сложными технологическими процессами, является метод управления с прогнозирующей моделью (управление со скользящим горизонтом) [10, 11]. В работе [12] рассматривается задача синтеза прогнозирующего управления системами с марковскими скачками и мультипликативными шумами при ограничениях на управляющие воздействия, при этом предполагается, что матрица динамики системы не зависит от состояния цепи Маркова. В [13] синтезированы стратегии управления такими системами по критерию «mean-variance». В работе [14] рассматривается задача управления системами с марковским переключением режимов при условии, что матрицы динамики и управления зависят от скачков. Задача прогнозирующего управления системами с сериально коррелированными параметрами при ограничениях рассмотрена в работе [15].

В настоящей работе рассматривается задача синтеза стратегий управления с прогнозированием для дискретных линейных систем с марковскими скачками и сериально коррелированными параметрами, для которых известны только первые и вторые моменты распределений. Динамика системы определяется состоянием однородной марковской цепи с известной матрицей переходных вероятностей. Получены уравнения синтеза оптимальных стратегий управления с учетом «жестких» ограничений на управляющие переменные.

1. Постановка задачи

Пусть объект управления описывается уравнением

$$x(k+1) = A[\alpha(k+1), k+1]x(k) + B[\alpha(k+1), \eta(k+1), k+1]u(k), \quad (1)$$

где $x(k) \in \mathbb{R}^{n_x}$ – n_x -мерный вектор состояния, $u(k) \in \mathbb{R}^{n_u}$ – n_u -мерный вектор управления; $A[\alpha(k), k] \in \mathbb{R}^{n_x \times n_x}$, $B[\alpha(k), \eta(k), k] \in \mathbb{R}^{n_x \times n_u}$ – матрицы соответствующих размерностей; $\eta(k) \in \mathbb{R}^q$ – последовательность сериально коррелированных случайных величин; $B[\alpha(k), \eta(k), k]$ зависит от $\eta(k)$ линейно; $\alpha(k)$ ($k = 0, 1, 2, \dots, v$) – однородная дискретная марковская цепь с конечным множеством состояний $\{1, 2, \dots, v\}$, известной матрицей переходных вероятностей:

$$P = [P_{ij}], (i, j = \overline{1, v}), P_{ij} = P\{\alpha(k+1) = \alpha_j | \alpha(k) = \alpha_i\}, \sum_{j=1}^v P_{ij} = 1,$$

и известным начальным распределением

$$p_i = P\{\alpha(0) = i\}, (i = \overline{1, v}), \sum_{i=1}^v p_i = 1.$$

Матрицы $A[\alpha(k), k]$ и $B[\alpha(k), \eta(k), k]$ определяются состоянием α_i марковской цепи $\alpha(k)$ из множеств

$$A[\alpha(k), k] \in \{A^{(i)}(k) \in \mathbb{R}^{n_x \times n_x} : i = \overline{1, v}\},$$

$$B[\alpha(k), \eta(k), k] \in \{B^{(i)}[\eta(k), k] \in \mathbb{R}^{n_x \times n_u} : i = \overline{1, v}\}.$$

Предполагается, что состояние марковской цепи в момент времени k доступно наблюдению. Последовательности $\alpha(k)$ и $\eta(k)$ независимы.

Пусть $\mathbb{F} = (\mathcal{F}_k)_{k \geq 1}$ – поток σ -алгебр, где каждая σ -алгебра порождается последовательностью $\{\eta(s) : s = 0, 1, \dots, k\}$ и интерпретируется как доступная информация до момента времени k включительно. Для процесса $\eta(k)$ предполагаются известными условные моменты распределений

$$E\{\eta(k+i) / \mathcal{F}_k\} = \bar{\eta}(k+i), \quad (2)$$

$$E\{\eta(k+i)\eta^T(k+j) / \mathcal{F}_k\} = \Theta_{ij}(k), \quad (3)$$

$$(k = 0, 1, 2, \dots), (i, j = 1, 2, \dots, l).$$

На управляющие воздействия накладываются ограничения вида

$$u_{\min}(k) \leq S(k)u(k) \leq u_{\max}(k), \quad (4)$$

где $S(k) \in \mathbb{R}^{p \times n_u}$; $u_{\min}(k), u_{\max}(k) \in \mathbb{R}^p$.

Необходимо определить закон управления системой (1) при ограничениях (4) из условия минимума критерия со скользящим горизонтом управления

$$J(k+m/k) = \sum_{i=1}^m E\{x^T(k+i)R_1(k+i)x(k+i) / x(k), \alpha(k), \mathcal{F}_k\} - \sum_{i=1}^m R_2(k+i)E\{x(k+i) / x(k), \alpha(k), \mathcal{F}_k\} + \sum_{i=0}^{m-1} E\{u^T(k+i/k)R(k+i)u(k+i/k) / x(k), \alpha(k), \mathcal{F}_k\}, \quad (5)$$

где $E\{\dots\}$ – оператор условного математического ожидания; m – горизонт прогноза; k – текущий момент времени; $R_1(k+i) \geq 0$, $R_2(k+i) \geq 0$ и $R(k+i) > 0$ – весовые матрицы соответствующих размерностей.

Для решения сформулированной задачи используем методологию управления с прогнозирующей моделью. Данный подход позволяет получить стратегии управления с обратной связью с учетом явных ограничений на управляющие воздействия.

Стратегии управления с прогнозированием определяются по следующему правилу. На каждом шаге k минимизируем функционал (5) по последовательности прогнозирующих управлений $u(k/k), \dots, u(k+m-1/k)$, зависящих от состояния системы в момент времени k . В качестве управления в момент времени k берем $u(k) = u(k/k)$. Тем самым получаем управление $u(k)$ как функцию состояний $x(k)$ и $\alpha(k) = \alpha_j$, т.е. управление с обратной связью. Чтобы получить управление $u(k+1)$ на следующем шаге, процедура повторяется для следующего момента $k+1$ и т.д.

2. Синтез стратегий прогнозирующего управления

Цепь Маркова с дискретным временем допускает следующее представление в пространстве состояний [16]:

$$\theta(k+1) = P\theta(k) + v(k+1), \quad (6)$$

где $\theta(k) = [\delta(\alpha(k), 1), \dots, \delta(\alpha(k), v)]^T$, $\delta(\alpha(k), j)$ – функция Кронекера ($j = \overline{1, v}$); $v(k+1)$ – мартингал-разность с характеристиками

$$E\{v(k+1) / \theta(k)\} = 0,$$

$$C(k+1) = E\{v(k+1)v^T(k+1) / \theta(k)\} = \text{diag}\{P\theta(k)\} - P\text{diag}\{\theta(k)\}P^T.$$

С учетом (6) систему (1) можно представить в следующем виде:

$$x(k+1) = A[\theta(k+1), k+1]x(k) + B[\theta(k+1), \eta(k+1), k+1]u(k), \quad (7)$$

где матрицы $A[\theta(k), k]$ и $B[\theta(k), \eta(k), k]$ имеют вид

$$A[\theta(k), k] = \sum_{i=1}^v \theta_i(k) A^{(i)}(k), \quad (8)$$

$$B[\theta(k), \eta(k), k] = \sum_{i=1}^v \theta_i(k) B^{(i)}[\eta(k), k], \quad (9)$$

где $\theta_i(k)$ ($i = \overline{1, v}$) – компоненты вектора $\theta(k)$.

Критерий (5) будет иметь вид

$$J(k+m/k) = \sum_{i=1}^m E\{x^T(k+i)R_1(k+i)x(k+i) / x(k), \theta(k), \mathfrak{F}_k\} - \\ - \sum_{i=1}^m R_2(k+i)E\{x(k+i) / x(k), \theta(k), \mathfrak{F}_k\} + \sum_{i=0}^{m-1} E\{u^T(k+i/k)R(k+i)u(k+i/k) / x(k), \theta(k), \mathfrak{F}_k\}. \quad (10)$$

Теорема. Пусть динамика системы описывается выражением (1) с учетом ограничений (4). Тогда стратегия прогнозирующего управления с горизонтом прогноза m минимизирующая критерий (5) на каждом шаге k равна

$$u(k) = \begin{bmatrix} I_{n_u} & 0_{n_u} & \dots & 0_{n_u} \end{bmatrix} U(k), \quad (11)$$

где I_{n_u} – единичная матрица размерности n_u ; 0_{n_u} – квадратная нулевая матрица размерности n_u ; $U(k) = [u^T(k/k), \dots, u^T(k+m-1/k)]^T$ – вектор прогнозирующих управлений, который определяется из решения задачи квадратичного программирования с критерием вида

$$Y(k+m/k) = [2x^T(k)G(k) - F(k)]U(k) + U^T(k)H(k)U(k) \quad (12)$$

при ограничениях

$$U_{\min}(k) \leq \bar{S}(k)U(k) \leq U_{\max}(k), \quad (13)$$

где

$$\bar{S}(k) = \text{diag}(S(k), \dots, S(k+m-1)),$$

$$U_{\min}(k) = \begin{bmatrix} u_{\min}^T(k), \dots, u_{\min}^T(k+m-1) \end{bmatrix}^T, U_{\max}(k) = \begin{bmatrix} u_{\max}^T(k), \dots, u_{\max}^T(k+m-1) \end{bmatrix}^T,$$

$H(k), G(k), F(k)$ – блочные матрицы вида $H(k) = \{H_{t,s}(k)\}$, $G(k) = \{G_t(k)\}$, $F(k) = \{F_t(k)\}$, $s, t = \overline{1, m}$, блоки которых определяются выражениями

$$H_{t,t}(k) = \sum_{i_t=1}^v E\{(B^{(i_t)}[\eta(k+t), k+t])^T Q^{(i_t)}(k) B_j^{(i_t)}[\eta(k+t), k+t] / \mathfrak{F}_k\} + R(k+t-1), \quad (14)$$

$$H_{t,s}(k) = \sum_{i_t=1}^v \sum_{i_{t+1}=1}^v \dots \sum_{i_s=1}^v E\{(B^{(i_t)}[\eta(k+t), k+t])^T (A^{(i_{t+1})}(k+t+1))^T \dots (A^{(i_s)}(k+s))^T \times \\ \times Q^{(i_t, \dots, i_s)}(k) B^{(i_s)}[\eta(k+s), k+s] / \mathfrak{F}_k\}, s > t, \quad (15)$$

$$H_{s,t}(k) = H_{t,s}^T(k), s < t, \quad (16)$$

$$G_t(k) = \sum_{i_t=1}^v \dots \sum_{i_1=1}^v (A^{(i_1)}(k+1))^T \dots (A^{(i_t)}(k+t))^T Q^{(i_1, i_2, \dots, i_t)}(k) E\{B^{(i_t)}[\eta(k+t), k+t] / \mathfrak{F}_k\}, \quad (17)$$

$$F_t(k) = \sum_{i_t=1}^v Q_2^{(i_t)}(k) E\{B^{(i_t)}[\eta(k+t), k+t] / \mathfrak{F}_k\}. \quad (18)$$

Последовательность матриц $Q^{(i_t, \dots, i_s)}, Q_2^{(i_t, \dots, i_s)}$, $s, t = \overline{1, m}$, представляет собой обратную рекурсию:

$$Q^{(i_t, \dots, i_s)}(k) = \Theta^{(i_t, \dots, i_s)}(k) R_1(k+s) + \sum_{i_{s+1}=1}^v \left(A^{i_{s+1}}(k+s+1) \right)^T Q^{(i_t, \dots, i_{s+1})}(k) A^{i_{s+1}}(k+s+1), \quad (19)$$

$$t = \overline{1, m-2}, t < s < m,$$

$$Q^{(i_t)}(k) = e_{i_t} P^t \theta(k) R_1(k+t) + \sum_{i_{t+1}=1}^v \left(A^{(i_{t+1})}(k+t+1) \right)^T Q^{(i_t, i_{t+1})}(k) A^{(i_{t+1})}(k+t+1), \quad t = \overline{1, m-1}, \quad (20)$$

$$Q_2^{(i_t, \dots, i_s)}(k) = R_2(k+s) \Theta^{(i_t, \dots, i_s)}(k) + \sum_{i_{s+1}=1}^v Q_2^{(i_t, \dots, i_{s+1})}(k) A^{(i_{s+1})}(k+s+1), \quad t = \overline{1, m-2}, t < s < m, \quad (21)$$

$$Q_2^{(i_t)}(k) = R_2(k+t) e_{i_t} P^t \theta(k) + \sum_{i_{t+1}=1}^v Q_2^{(i_t, i_{t+1})}(k) A^{(i_{t+1})}(k+t+1), \quad t = \overline{1, m-1}, \quad (22)$$

с начальными условиями

$$Q^{(i_m)}(k) = e_{i_m} P^m \theta(k) R_1(k+m),$$

$$Q^{(i_t, \dots, i_m)}(k) = \Theta^{(i_t, \dots, i_m)}(k) R_1(k+m), \quad t = \overline{1, m-1},$$

$$Q_2^{(i_t, \dots, i_m)}(k) = R_2(k+m) \Theta^{(i_t, \dots, i_m)}(k), \quad t = \overline{1, m-1},$$

$$Q_2^{(i_m)}(k) = R_2(k+m) e_{i_m} P^m \theta(k),$$

где

$$\Theta^{(i_t, \dots, i_s)} = P_{i_t, i_{s-1}} P_{i_{s-1}, i_{s+1}} \dots P_{i_{t+1}, i_t} \theta_{i_t}(k+t/k), \quad t = \overline{1, m-1}, s > t, \quad (23)$$

$\theta_{i_t}(k+t/k)$ – компоненты вектора прогноза:

$$\theta(k+t/k) = E\{\theta(k+t) / \theta(k)\} = P^t \theta(k),$$

$$e_{i_t} = [0, \dots, 0, 1, 0, \dots, 0]_{1 \times v}, i_t = \overline{1, v}, t = \overline{1, m}.$$

Замечание. Принимая во внимание предположение о линейной зависимости матриц $B[\eta(k+i), k+i]$ от $\eta(k+i)$, выражения (14), (15) вычислимы без принципиальных затруднений при условии известных первых и вторых моментов (2), (3) для процесса $\eta(k+i)$.

Доказательство. Введем обозначение:

$$J_{k+s} = E\{x^T(k+1) R_1(k+1) x(k+1) - R_2(k+1) x(k+1) + u^T(k) R(k) u(k) / x(k), \theta(k), \mathfrak{F}_k\} +$$

$$+ E\{x^T(k+2) R_1(k+2) x(k+2) - R_2(k+2) x(k+2) + u^T(k+1) R(k+1) u(k+1) / x(k), \theta(k), \mathfrak{F}_k\} + \dots$$

$$+ E\{x^T(k+s) R_1(k+s) x(k+s) - R_2(k+s) x(k+s) +$$

$$+ u^T(k+s-1) R(k+s-1) u(k+s-1) / x(k), \theta(k), \mathfrak{F}_k\}, s = \overline{1, m}.$$

Очевидно, что справедливо следующее выражение:

$$J_{k+s+1} = J_{k+s} + E\{x^T(k+s+1) R_1(k+s+1) x(k+s+1) - R_2(k+s+1) x(k+s+1) +$$

$$+ u^T(k+s) R(k+s) u(k+s) / x(k), \theta(k), \mathfrak{F}_k\},$$

$$J(k+m/k) = J_{k+m}.$$

Рассмотрим J_{k+1} :

$$J_{k+1} = E\{x^T(k+1) R_1(k+1) x(k+1) - R_2(k+1) x(k+1) + u^T(k) R(k) u(k) / x(k), \theta(k), \mathfrak{F}_k\}. \quad (24)$$

Подставляя $x(k+1)$ из (1) в (24) и используя представление цепи Маркова в виде (6), получим

$$J_{k+1} = E\{x^T(k) \sum_{i_1=1}^v \sum_{j_1=1}^v (A^{(i_1)}(k+1))^T R_1(k+1) A^{(j_1)}(k+1) \theta_{i_1}(k+1) \theta_{j_1}(k+1) x(k) +$$

$$\begin{aligned}
& + u^T(k) \sum_{i_1=1}^v \sum_{j_1=1}^v (B^{(i_1)}[\eta(k+1), k+1])^T R_1(k+1) B^{(j_1)}[\eta(k+1), k+1] \theta_{i_1}(k+1) \theta_{j_1}(k+1) u(k) + \\
& + 2x^T(k) \sum_{i_1=1}^v \sum_{j_1=1}^v (A^{(i_1)}(k+1))^T R_1(k+1) B^{(j_1)}[\eta(k+1), k+1] \theta_{i_1}(k+1) \theta_{j_1}(k+1) u(k) - \\
& - R_2(k+1) \sum_{i_1=1}^v \theta_{i_1}(k+1) [A^{(i_1)}(k+1)x(k) + B^{(i_1)}[\eta(k+1), k+1]u(k)] + \\
& + u^T(k) R(k) u(k) / x(k), \theta(k), \mathfrak{F}_k \}.
\end{aligned}$$

Заметим, что $\theta_{i_1}(k+1)\theta_{j_1}(k+1) \neq 0$ только при $i_1 = j_1$. Кроме того, так как $\theta_{i_1}(k+1)$ – функция Кронекера, то $\theta_{i_1}^2(k+1) = \theta_{i_1}(k+1)$. Взяв математическое ожидание, выражение для J_{k+1} можно переписать в виде

$$\begin{aligned}
J_{k+1} &= E\{x^T(k) \sum_{i_1=1}^v (A^{(i_1)}(k+1))^T R_1(k+1) A^{(i_1)}(k+1) E_{i_1} P \theta(k) x(k) + \\
& + u^T(k/k) \sum_{i_1=1}^v E\{(B^{(i_1)}[\eta(k+1), k+1])^T R_1(k+1) B^{(i_1)}[\eta(k+1), k+1] / \mathfrak{F}_k\} e_{i_1} P \theta(k) u(k/k) + \\
& + 2x^T(k) \sum_{i_1=1}^v (A^{(i_1)}(k+1))^T R_1(k+1) E\{B^{(i_1)}[\eta(k+1), k+1] / \mathfrak{F}_k\} e_{i_1} P \theta(k) u(k/k) - \\
& - R_2(k+1) \sum_{i_1=1}^v e_{i_1} P \theta(k) [A^{(i_1)}(k+1)x(k) + E\{B^{(i_1)}[\eta(k+1), k+1] / \mathfrak{F}_k\} u(k/k)] + \\
& + u^T(k/k) R(k) u(k/k).
\end{aligned}$$

Аналогично на втором шаге имеем

$$J_{k+2} = J_{k+1} + E\{x^T(k+2)R_1(k+2)x(k+2) - R_2(k+2)x(k+2) + u^T(k+1)R(k+1)u(k+1) / x(k), \theta(k), \mathfrak{F}_k\}.$$

Используя уравнение динамики системы (1) и представление цепи Маркова в виде (6), получим после взятия математических ожиданий

$$\begin{aligned}
J_{k+2} &= x^T(k) \sum_{i_1=1}^v (A^{(i_1)}(k+1))^T Q^{(i_1)}(k) A^{(i_1)}(k+1) x(k) + \\
& + 2x^T(k) \sum_{i_1=1}^v (A^{(i_1)}(k+1))^T Q^{(i_1)}(k) E\{B^{(i_1)}[\eta(k+1), k+1] / \mathfrak{F}_k\} u(k/k) + \\
& + u^T(k/k) \sum_{i_1=1}^v E\{(B^{(i_1)}[\eta(k+1), k+1])^T Q^{(i_1)}(k) B^{(i_1)}[\eta(k+1), k+1] / \mathfrak{F}_k\} u(k/k) + \\
& + u^T(k+1/k) \sum_{i_2=1}^v E\{(B^{(i_2)}[\eta(k+2), k+2])^T Q^{(i_2)}(k) B^{(i_2)}[\eta(k+2), k+2] / \mathfrak{F}_k\} u(k+1/k) + \\
& + 2u^T(k/k) \sum_{i_1=1}^v E\{(B^{(i_1)}[\eta(k+1), k+1])^T \sum_{i_2=1}^v (A^{(i_2)}(k+2))^T Q^{(i_1, i_2)}(k) B^{(i_2)}[\eta(k+2), k+2] / \mathfrak{F}_k\} u(k+1/k) - \\
& - \sum_{i_1=1}^v Q_2^{(i_1)} A^{(i_1)}(k+1) x(k) - \sum_{i_1=1}^v Q_2^{(i_1)} E\{B^{(i_1)}[\eta(k+1), k+1] / \mathfrak{F}_k\} u(k/k) - \\
& - \sum_{i_2=1}^v Q_2^{(i_2)} E\{B^{(i_2)}[\eta(k+2), k+2] / \mathfrak{F}_k\} u(k+1/k) + \\
& + u^T(k/k) R(k) u(k/k) + u^T(k+1/k) R(k+1) u(k+1/k),
\end{aligned}$$

где матрицы $Q^{(i_1)}(k), Q^{(i_1, \dots, i_s)}(k), Q_2^{(i_1)}(k)$ определяются уравнениями (19)–(22).

Повторяя процедуру для $J_{k+3}, J_{k+4}, \dots$, получим выражение для $J(k+m/k)$:

$$\begin{aligned}
J(k+m/k) = & x^T(k) \sum_{i_1=1}^V (A^{(i_1)}(k+1))^T Q^{(i_1)}(k) A^{(i_1)}(k+1) x(k) + \\
& + 2x^T(k) \sum_{t=1}^m \sum_{i_1=1}^V \dots \sum_{i_t=1}^V (A^{(i_1)}(k+1))^T \dots (A^{(i_t)}(k+t))^T Q^{(i_1, i_2, \dots, i_t)}(k) E\{B^{(i_t)}[\eta(k+t), k+t] / \mathfrak{F}_k\} u(k+t-1/k) + \\
& + \sum_{t=1}^m u^T(k+t-1/k) \left[\sum_{i_t=1}^V E\{B^{(i_t)}[\eta(k+t), k+t]\}^T Q^{(i_t)}(k) B^{(i_t)}[\eta(k+t), k+t] / \mathfrak{F}_k\} + R(k+t-1) \right] u(k+t-1/k) + \\
& + 2 \sum_{t=1}^{m-1} \sum_{s=t+1}^m u^T(k+t-1/k) \sum_{i_t=1}^V E\{B^{(i_t)}[\eta(k+t), k+t]\}^T \sum_{i_{t+1}=1}^V \dots \sum_{i_s=1}^V (A^{(i_{t+1})}(k+t+1))^T \dots (A^{(i_s)}(k+s))^T Q^{(i_t, \dots, i_s)}(k) \times \\
& \times B^{(i_s)}[\eta(k+s), k+s] / \mathfrak{F}_k\} u(k+s-1/k) - \\
& - \sum_{i_1=1}^V Q_2^{(i_1)} A^{(i_1)}(k+1) x(k) - \sum_{t=1}^m \sum_{i_t=1}^V Q_2^{(i_t)} E\{B^{(i_t)}[\eta(k+t), k+t] / \mathfrak{F}_k\} u(k+t-1/k),
\end{aligned}$$

где матрицы $Q^{(i_1)}(k), Q^{(i_1, \dots, i_s)}(k), Q_2^{(i_t)}(k)$ определяются уравнениями (19)–(22).

Критерий $J(k+m/k)$ может быть записан в матричном виде:

$$\begin{aligned}
J(k+m/k) = & x^T(k) \sum_{i_1=1}^V (A^{(i_1)}(k+1))^T Q^{(i_1)}(k) A^{(i_1)}(k+1) x(k) - \sum_{i_1=1}^V Q_2^{(i_1)} A^{(i_1)}(k+1) x(k) + \\
& + [2x^T(k)G(k) - F(k)]U(k) + U^T(k)H(k)U(k),
\end{aligned} \tag{25}$$

где матрицы $H(k), G(k), F(k)$ имеют вид (14)–(18).

Минимизация данного критерия эквивалентна минимизации критерия вида

$$Y(k+m/k) = [2x^T(k)G(k) - F(k)]U(k) + U^T(k)H(k)U(k).$$

Таким образом, имеем задачу минимизации критерия (12) при ограничениях (13), которая эквивалентна задаче квадратичного программирования с критерием (5) при ограничениях (4).

Заключение

В данной работе предложен метод синтеза стратегий прогнозирующего управления по квадратичному критерию для линейных дискретных систем с марковскими скачками и сериально коррелированными параметрами. Данный подход позволяет в явном виде учесть ограничения на управления. Алгоритм синтеза прогнозирующей стратегии включает решение последовательности задач квадратичного программирования.

ЛИТЕРАТУРА

1. Пакшин П.В. Дискретные системы со случайными параметрами и структурой. М.: Физматлит, 1994.
2. Dombrovskii V., Obedko T. Portfolio Optimization in the Financial Market with Regime Switching under Constraints and Transaction Costs Using Model Predictive Control // European Control Conference (ECC). July 2015. P. 3371–3376.
3. Пакшин П.В., Ретинский Д.М. Робастная стабилизация систем случайной структуры с переключаемой статической обратной связью по выходу // Автоматика и телемеханика. 2005. № 7. С. 135–147.
4. Смагин В.И., Поползухина Е.В. Синтез следящих систем управления для объектов со случайными скачкообразными параметрами и мультипликативными возмущениями // Вестник Томского государственного университета. 2000. № 271. С. 171–175.
5. Blackmore L., Bektassov A., Ono M., Williams B.C. Robust optimal predictive control of jump Markov linear systems using particles // Lecture Notes in Computer Science. 2007. V. 4416. P. 104–117.
6. Costa O.L.V., Okimura R.T. Discrete-time mean-variance optimal control of linear systems with Markovian jumps and multiplicative noise // International Journal of Control. 2009. V. 82, No. 2. P. 256–267.
7. Costa O.L.V., Oliveira A. Optimal mean-variance control for discrete-time linear systems with Markovian jumps and multiplicative noises // Automatica. 2012. V. 48, No. 2. P. 304–315.
8. Dragan V., Morozan T. The Linear Quadratic Optimization Problems for a Class of Linear Stochastic Systems with Multiplicative White Noise and Markovian Jumping // IEEE Transactions on Automatic Control. 2004. V. 49, No. 5. P. 665–675.
9. Li X., Zhou X.Y. Indefinite stochastic LQ control with Markovian jumps in a finite time horizon // Communications in Information and Systems. 2002. No. 2. P. 265–282.

10. Mayne D.Q. Model predictive control: Recent developments and future promise // *Automatica*. 2014. V. 50. P. 2967–2986.
11. Rawlings J. Tutorial: Model Predictive Control Technology // *Proc. Amer. Control Conf. San Diego, California*. June 1999. P. 662–676.
12. Домбровский В.В., Обедко Т.Ю. Управление с прогнозированием системами с марковскими скачками при ограничениях и применение к оптимизации инвестиционного портфеля // *Автоматика и телемеханика*. 2011. № 5. С. 96–112.
13. Домбровский В.В., Обедко Т.Ю. Управление с прогнозирующей моделью линейными системами с марковскими скачками по критерию «mean-variance» при ограничениях // *Вестник Томского государственного университета. Управление, вычислительная техника и информатика*. 2012. № 4 (21). С. 5–13.
14. Домбровский В.В., Самородова М.В. Управление с прогнозированием по квадратичному критерию линейными дискретными системами с марковскими скачками при ограничениях // *Вестник Томского государственного университета. Управление, вычислительная техника и информатика*. 2016. № 1 (34). С. 4–10.
15. Dombrovskii V., Obedko T. Model predictive control for constrained systems with serially correlated stochastic parameters and portfolio optimization // *Automatica*. 2015. V. 54. P. 325–331.
16. Elliott R.J., Aggoun L., Moore J.B. *Hidden Markov Models: Estimation and Control*. Berlin : Springer-Verlag, 1995.

Домбровский Владимир Валентинович, д-р техн. наук, профессор. E-mail: dombrovs@ef.tsu.ru

Обедко Татьяна Юрьевна, канд. физ.-мат. наук. E-mail: tatyana.obedko@mail.ru

Национальный исследовательский Томский государственный университет

Поступила в редакцию 25 февраля 2017 г.

Dombrovskii Vladimir V., Obedko Tatiana Y. (National Research Tomsk State University, Russian Federation).

Model predictive control for stochastic systems with Markovian jumps and serially correlated parameters under constraints.

Keywords: stochastic systems; Markovian jumps; serially correlated parameters; model predictive control; constraints.

DOI: 10.17223/19988605/40/1

Assume that the plant to be controlled can be described by the following model:

$$x(k+1) = A[\alpha(k+1), k+1]x(k) + B[\alpha(k+1), \eta(k+1), k+1]u(k), \quad (1)$$

where $x(k) \in \mathbb{R}^{n_x}$ is the vector of state, $u(k) \in \mathbb{R}^{n_u}$ is the vector of control inputs; $A[\alpha(k), k] \in \mathbb{R}^{n_x \times n_x}$, $B[\alpha(k), \eta(k), k] \in \mathbb{R}^{n_x \times n_u}$ are system and input matrices, respectively; $\eta(k) \in \mathbb{R}^q$ is assumed to be a stochastic time series and all of the elements of $B[\eta(k), k]$ are assumed to be linear functions of $\eta(k)$; $\{\alpha(k); k = 0, 1, 2, \dots\}$ is a finite-state discrete-time Markov chain taking values in $\{1, 2, \dots, v\}$ with known transition probability matrix and initial distribution. We assume that $\alpha(k)$ and $\eta(k)$ are mutually independent and at the instant of decision making, the current state of the market is known, i.e., the Markov state $\{\alpha(k)\}$ is observable. Let $\mathbb{F} = (\mathfrak{F}_k)_{k \geq 1}$ be the complete filtration with σ -field $\mathfrak{F}_k$ generated by the $\{\eta(s); s = 0, 1, 2, \dots, k\}$ that models the flow of information to time k . We allow the time series $\eta(k)$ to be serially correlated. Let us assume that we know the first- and second-order conditional moments for the stochastic vector $\eta(k)$ about $\mathfrak{F}_k$:

$$E\{\eta(k+i) / \mathfrak{F}_k\} = \bar{\eta}(k+i), \quad E\{\eta(k+i)\eta^T(k+j) / \mathfrak{F}_k\} = \Theta_{ij}(k), \quad (k = 0, 1, 2, \dots), (i, j = 1, 2, \dots, l).$$

We impose the following constraints on the decision variables:

$$u_{\min}(k) \leq S(k)u(k) \leq u_{\max}(k), \quad S(k) \in \mathbb{R}^{p \times n_u}, \quad u_{\min}(k), u_{\max}(k) \in \mathbb{R}^p. \quad (2)$$

To control system (1) subject to constraints (2), at each step k we minimize the quadratic criterion with a receding horizon

$$J(k+m/k) = E\left\{\sum_{i=1}^m x^T(k+i)R_1(k+i)x(k+i) - R_2(k+i)x(k+i) + u^T(k+i-1/k)R(k+i-1)u(k+i-1/k) / x(k), \alpha(k), \mathfrak{F}_k\right\},$$

where m is the prediction horizon; $u(k/k), \dots, u(k+m-1/k)$ is the sequence of predictive controls under; $R_1(k+i) \geq 0$, $R_2(k+i) \geq 0$, and $R(k+i) > 0$ are the weight matrices of corresponding dimensions.

The model predictive control methodology was used to solve the problem. The optimal control strategies were synthesized under hard constraints imposed on the control variables.

REFERENCES

1. Pakshin, P.V. (1994) *Diskretnye sistemy so sluchaynymi parametrami i strukturoy* [Discrete-systems with stochastic parameters and structure]. Moscow: Fizmatlit.
2. Dombrovskii, V. & Obedko, T. (2015) Portfolio Optimization in the Financial Market with Regime Switching under Constraints and Transaction Costs Using Model Predictive Control. *European Control Conference (ECC)*. July. pp. 3371–3376. DOI: 10.1109/ECC.2015.7331055
3. Pakshin, P.V. & Retinskii, D.M. (2005) Robust Stabilization of Random-Structure Systems via Switchable Static Output Feedback. *Automation and Remote Control*. 66(7). pp. 1153–1161. DOI: 10.1007/s10513-005-0155-5
4. Smagin, V.I. & Popolzhina, E.V. (2000) The synthesis of tracking control systems for objects with random switching parameters and multiplicative noises. *Vestnik Tomskogo gosudarstvennogo universiteta – Tomsk State University Journal*. 271. pp. 171–175. (In Russian).

5. Blackmore, L., Bektassov, A., Ono, M. & Williams, B.C. (2007) Robust optimal predictive control of jump Markov linear systems using particles. *Lecture Notes in Computer Science*. 4416. pp. 104–117.
6. Costa, O.L.V. & Okimura, R.T. (2009) Discrete-time mean-variance optimal control of linear systems with Markovian jumps and multiplicative noise. *International Journal of Control*. 82(2). pp. 256–267. DOI: 10.1080/00207170802050825
7. Costa, O.L.V. & Oliveira, A. (2012) Optimal mean-variance control for discrete-time linear systems with Markovian jumps and multiplicative noises. *Automatica*. 48(2). pp. 304–315. DOI: 10.1080/00207170802050825
8. Dragan, V. & Morozan, T. (2004) The Linear Quadratic Optimization Problems for a Class of Linear Stochastic Systems with Multiplicative White Noise and Markovian Jumping. *IEEE Transactions on Automatic Control*. 49(5). pp. 665–675. DOI: 10.1109/TAC.2004.826718
9. Li, X. & Zhou, X.Y. (2002) Indefinite stochastic LQ control with Markovian jumps in a finite time horizon. *Communications in Information and Systems*. 2. pp. 265–282. DOI: 10.4310/CIS.2002.v2.n3.a4
10. Mayne, D.Q. (2014) Model predictive control: Recent developments and future promise. *Automatica*. 50. pp. 2967–2986. DOI: 10.1016/j.automatica.2014.10.128
11. Rawlings, J. (1999) Tutorial: Model Predictive Control Technology. *Proc. Amer. Control Conf. San Diego, California*. pp. 662–676. DOI: 10.1109/ACC.1999.782911
12. Dombrovskii, V.V. & Obedko, T.Yu. (2011) Predictive control of systems with Markovian jumps under constraints and its application to the investment portfolio optimization. *Automation and Remote Control*. 72(5), pp. 989–1003. DOI: <https://doi.org/10.1134/S0005117911050079>
13. Dombrovskii, V.V. & Obedko, T.Yu. (2012) Mean-variance MPC for linear systems with Markovian jumps under constraints. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science*. 4(21). pp. 5–13. (In Russian).
14. Dombrovskii, V.V. & Samorodova, M.V. (2016) Model predictive control with quadratic criterion for jump Markov discrete linear systems under constraints. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science*. 1(34), pp. 4–10. (In Russian).
15. Dombrovskii, V. & Obedko, T. (2015) Model predictive control for constrained systems with serially correlated stochastic parameters and portfolio optimization. *Automatica*. vol. 54. pp. 325–331. DOI: 10.1016/j.automatica.2015.02.021
16. Elliott, R.J., Aggoun, L. & Moore, J.B. (1995) *Hidden Markov Models: Estimation and Control*. Berlin: Springer-Verlag.

К.Б. Мансимов, Ш.М. Расулова

ОБ ОДНОЙ ЗАДАЧЕ УПРАВЛЕНИЯ ТИПА А.И. МОСКАЛЕНКО

Рассматривается задача оптимального управления для модели объекта, описываемого совокупностью двух дифференциальных уравнений, связанных друг с другом по независимым переменным и начальным условиям. Доказан соответствующий аналог принципа максимума Понтрягина, рассмотрен случай его вырождения.

Ключевые слова: условие максимума Понтрягина; особый случай; необходимое условие оптимальности; задача оптимального управления с распределенными параметрами.

В работах [1, 2] А.И. Москаленко исследованы задачи оптимального управления, занимающих некоторое промежуточное положение между задачами оптимального управления сосредоточенными и распределенными параметрами.

В настоящей статье изучается задача типа А.И. Москаленко [1], причем в отличие от этой работы критерий качества является многоточечным. Получены необходимые условия оптимальности первого порядка в форме принципа максимума Понтрягина и изучен случай вырождения условий (особое управление).

1. Постановка задачи

Требуется минимизировать многоточечный функционал

$$S(u) = \varphi(y(X_1), y(X_2), \dots, y(X_k)) + \int_{x_0}^{x_1} F(x, z(T_1, x), z(T_2, x), \dots, z(T_k, x)) dx \quad (1)$$

при ограничениях

$$u(t) \in U \subset R^r, \quad t \in T = [t_0, t_1],$$

$$v(x) \in V \subset R^q, \quad x \in X = [x_0, x_1], \quad (2)$$

$$\frac{\partial z(t, x)}{\partial t} = f(t, x, z(t, x), u(t)), \quad (t, x) \in D = T \times X, \quad (3)$$

$$z(t_0, x) = y(x), \quad x \in X, \quad (4)$$

$$\frac{dy}{dx} = g(x, y(x), v(x)), \quad x \in X, \quad (5)$$

$$y(x_0) = y_0. \quad (6)$$

Здесь $f(t, x, z, u)$ ($g(x, y, v)$) – заданная n -мерная вектор-функция, определенная и непрерывная в $D \times R^n \times R^r$ ($X \times R^n \times R^q$) вместе с частными производными по z (y) до второго порядка включительно; t_0, t_1, x_0, x_1, y_0 заданы; $T_i \in (t_0, t_1]$, $i = \overline{1, k}$ ($t_0 < T_1 < T_2 < \dots < T_k \leq t_1$); $X_i \in (x_0, x_1]$, $i = \overline{1, k}$, ($x_0 < X_1 < X_2 < \dots < X_k \leq x_1$) – заданные точки; $\varphi(a_1, a_2, \dots, a_k)$ ($F(x, b_1, b_2, \dots, b_k)$) – заданная и непрерывная в $R^{k \cdot n}$ ($X \times R^{k \cdot n}$) вместе с частными производными по $(a_1, a_2, \dots, a_k)$ ($(b_1, b_2, \dots, b_k)$) до второго порядка включительно скалярная функция; U и V – заданные непустые и ограниченные множества; $u(t)$ ($v(x)$) – кусочно-непрерывная (с конечным числом точек разрыва первого рода) вектор-функция управляющих воздействий.

Пару функций $(u(t), v(x))$ с вышеприведенными свойствами назовем допустимым управлением.

Предполагается, что каждому допустимому управлению $(u^o(t), v^o(x))$ (в смысле [1, 2]) соответствует единственное решение $(z^o(t, x), y^o(x))$ для (2)–(6).

Допустимое управление $(u^o(t), v^o(x))$, доставляющее минимум функционалу (1) при ограничениях (2)–(6), назовем оптимальным управлением, а соответствующий процесс $(u^o(t), v^o(x), z^o(t, x), y^o(x))$ – оптимальным процессом.

2. Формула для приращения критерия качества

Пусть $(u^o(t), v^o(x))$ – фиксированное допустимое управление, а $(\bar{u}(t) = u^o(t) + \Delta u(t), \bar{v}(t) = v^o(x) + \Delta v(x))$ – произвольное допустимое управление. Через $(z^o(t, x), y^o(x))$, $(\bar{z}(t) = z^o(t) + \Delta z(t), \bar{y}(t) = y^o(x) + \Delta y(x))$ обозначим соответствующие им решения уравнений (3)–(6).

Функции $\Delta z(t, x), \Delta y(x)$ являются решением следующих уравнений

$$\frac{d\Delta y(x)}{dx} = g(x, \bar{y}(x), \bar{v}(x)) - g(x, y(x), v(x)), \Delta y(x_0) = 0, \quad (7)$$

$$\frac{\partial \Delta z(t, x)}{\partial t} = f(t, x, \bar{z}(t, x), \bar{u}(t)) - f(t, x, z(t, x), u(t)), \Delta z(t_0, x) = \Delta y(x). \quad (8)$$

Приращения функционала качества (1) с использованием формулы Тейлора можно записать в виде

$$\begin{aligned} \Delta S(u^o, v^o) &= S(\bar{u}, \bar{v}) - S(u^o, v^o) = \sum_{i=1}^k \frac{\partial \Phi'(y(X_1), \dots, y(X_k))}{\partial a_i} \Delta y(X_i) + \\ &+ \frac{1}{2} \sum_{i,j=1}^k \Delta y'(X_i) \frac{\partial^2 \Phi(y(X_1), \dots, y(X_k))}{\partial a_i \partial a_j} \Delta y(X_j) + o_1 \left(\left[\sum_{i=1}^k \|\Delta y(X_i)\| \right]^2 \right) + \\ &+ \sum_{i=1}^k \int_{x_0}^{x_1} \frac{\partial F'(x, z(T_i, x), \dots, z(T_k, x))}{\partial b_i} \Delta z(T_i, x) dx + \\ &+ \frac{1}{2} \sum_{i,j=1}^k \int_{x_0}^{x_1} \Delta z'(T_i, x) \frac{\partial F^2(x, z(T_i, x), \dots, z(T_k, x))}{\partial b_i \partial b_j} \Delta z(T_j, x) dx + o_2 \left(\left[\sum_{i=1}^k \|\Delta z(T_i, x)\| \right]^2 \right). \end{aligned} \quad (9)$$

Введем аналоги функций Гамильтона–Понтрягина:

$$H(t, x, z, u, p^o) = p^{o'} f(t, x, z, u), \quad M(x, y, v, q^o) = q^{o'} g(y, v),$$

где p^o, q^o – пока неизвестные вектор-функции. Тогда, используя соотношения (7), (8), приращение (9) функционала качества при помощи формулы Тейлора представляется в виде

$$\begin{aligned} \Delta S(u^o, v^o) &= \sum_{i=1}^k \frac{\partial \Phi'(y(X_1), \dots, y(X_k))}{\partial a_i} \Delta y(X_i) + \frac{1}{2} \sum_{i,j=1}^k \Delta y'(X_i) \frac{\partial^2 \Phi(y(X_1), \dots, y(X_k))}{\partial a_i \partial a_j} \Delta y(X_j) + \\ &+ o_1 \left(\left[\sum_{i=1}^k \|\Delta y(X_i)\| \right]^2 \right) + \sum_{i=1}^k \int_{x_0}^{x_1} \frac{\partial F'(x, z(T_i, x), \dots, z(T_k, x))}{\partial b_i} \Delta z(T_i, x) dx + \\ &+ \frac{1}{2} \sum_{i,j=1}^k \int_{x_0}^{x_1} \Delta z'(T_i, x) \frac{\partial F^2(x, z(T_i, x), \dots, z(T_k, x))}{\partial b_i \partial b_j} \Delta z(T_j, x) dx + o_2 \left(\left[\sum_{i=1}^k \|\Delta z(T_i, x)\| \right]^2 \right) + \\ &+ \int_{t_0}^{t_1} \int_{x_0}^{x_1} p^o(t, x) \frac{\partial \Delta z(t, x)}{\partial t} dx dt - \int_{t_0}^{t_1} \int_{x_0}^{x_1} [H(t, x, \bar{z}(t, x), \bar{u}(t), p^o(t, x)) - \end{aligned}$$

$$\begin{aligned}
& -H(t, x, z^o(t, x), u^o(t), p^o(t, x)) \Big] dx dt + \int_{x_0}^{x_1} q^{0'}(x) \frac{\partial \Delta y(x)}{\partial x} dx - \\
& - \int_{x_0}^{x_1} \left[M(x, \bar{y}(x), \bar{v}(x), q^o(x)) - M(x, y^o(x), v^o(x), q^o(x)) \right] dx dt.
\end{aligned} \tag{10}$$

В дальнейшем будем использовать следующие формулы

$$\Delta z(t, x) = \Delta y(x) + \int_{t_0}^t \frac{\partial \Delta z(\tau, x)}{\partial \tau} d\tau, \tag{11}$$

$$\Delta y(x) = \int_{x_0}^x \Delta \dot{y}(s) ds. \tag{12}$$

Введем обозначения :

$$\begin{aligned}
\Delta_{\bar{u}(t)} H(t, x) &= H(t, x, z^o(t, x), \bar{u}(t), p^o(t, x)) - H(t, x, z^o(t, x), u^o(t), p^o(t, x)), \\
H_z(t, x) &= H_z(t, x, z^o(t, x), u^o(t), p^o(t, x)), \quad H_{zz}(t, x) = H_{zz}(t, x, z^o(t, x), u^o(t), p^o(t, x)), \\
\Delta_{\bar{v}(x)} M(x) &= M(x, y^o(x), \bar{v}(x), q^o(x)) - M(x, y^o(x), v^o(x), q^o(x)), \\
M_y(x) &\equiv M_y(x, y^o(x), v^o(x), q^o(x)), \quad M_{yy}(x) \equiv M_{yy}(x, y^o(x), v^o(x), q^o(x)).
\end{aligned}$$

Учитывая эти обозначения и формулы (11), (12), из (10) получим

$$\begin{aligned}
\Delta S(u^o, v^o) &= \sum_{i=1}^k \int_{x_0}^{x_1} \alpha_i(x) \frac{\partial \Phi'(y(X_1), \dots, y(X_k))}{\partial a_i} \Delta \dot{y}(x) dx + \\
&+ \frac{1}{2} \sum_{i,j=1}^k \Delta y'(X_i) \frac{\partial^2 \Phi(y(X_1), y(X_2), \dots, y(X_k))}{\partial a_i \partial a_j} \Delta y(X_j) + \\
&+ \sum_{i=1}^k \int_{t_0}^{t_1} \int_{x_0}^{x_1} \beta_i(t) \frac{\partial F'(x, z(T_1, x), \dots, z(T_k, x))}{\partial b_i} \Delta z_t(t, x) dx dt + \sum_{i=1}^k \int_{t_0}^{t_1} \int_{x_0}^{x_1} \frac{\partial F'(x, z(T_1, x), \dots, z(T_k, x))}{\partial b_i} \Delta y(x) dx + \\
&+ \frac{1}{2} \sum_{i,j=1}^k \int_{t_0}^{t_1} \int_{x_0}^{x_1} \Delta z'(T_i, x) \frac{\partial^2 F(x, z(T_1, x), \dots, z(T_k, x))}{\partial b_i \partial b_j} \Delta z(T_j, x) dx + \int_{t_0}^{t_1} \int_{x_0}^{x_1} p^{0'}(t, x) \Delta z_t(t, x) dx dt - \\
&- \int_{t_0}^{t_1} \int_{x_0}^{x_1} \Delta_{\bar{u}(t)} H(t, x) dx dt - \int_{t_0}^{t_1} \int_{x_0}^{x_1} H'_z(t, x) \Delta z(t, x) dx dt - \int_{t_0}^{t_1} \int_{x_0}^{x_1} \Delta_{\bar{u}(t)} H'_z(t, x) \Delta z(t, x) dx dt + \\
&+ \int_{x_0}^{x_1} q^{0'}(x) \Delta \dot{y}(x) dx - \int_{x_0}^{x_1} \Delta_{\bar{v}(x)} M(x) dx - \int_{x_0}^{x_1} M'_y(x) \Delta y(x) dx - \frac{1}{2} \int_{t_0}^{t_1} \int_{x_0}^{x_1} \Delta z'(t, x) H_{zz}(t, x) \Delta z(t, x) dx dt - \\
&- \int_{x_0}^{x_1} \Delta_{\bar{v}(x)} M'_y(x) \Delta y(x) dx - \frac{1}{2} \int_{x_0}^{x_1} \Delta y'(x) M_{yy}(x) \Delta y(x) dx + \eta_1(\Delta u; \Delta v),
\end{aligned} \tag{13}$$

где по определению $\alpha_i(x)$, $\beta_i(t)$, $i = \overline{1, k}$, – характеристические функции отрезков $[x_0, X_i]$, $[t_0, T_i]$, $i = \overline{1, k}$, соответственно,

$$\begin{aligned}
\eta_1(\Delta u; \Delta v) &= o_1 \left(\left[\sum_{i=1}^k \|\Delta y(X_i)\| \right]^2 \right) + \int_{x_0}^{x_1} o_2 \left(\left[\sum_{i=1}^k \|\Delta z(T_i, x)\| \right]^2 \right) dx - \int_{t_0}^{t_1} \int_{x_0}^{x_1} o_3 \left(\|\Delta z(t, x)\|^2 \right) dx dt - \\
&- \frac{1}{2} \int_{x_0}^{x_1} \Delta y'(x) \Delta_{\bar{v}(x)} M_{yy}(x) \Delta y(x) dx - \frac{1}{2} \int_{t_0}^{t_1} \int_{x_0}^{x_1} \Delta z'(t, x) \Delta_{\bar{u}(t)} H_{zz}(t, x) \Delta z(t, x) dx dt - \int_{x_0}^{x_1} o_4 \left(\|\Delta y(x)\|^2 \right) dx.
\end{aligned} \tag{14}$$

Далее из (13) имеем

$$\Delta S(u^o, v^o) = \int_{x_0}^{x_1} \sum_{i=1}^k \alpha_i(x) \frac{\partial \Phi'(y(X_1), \dots, y(X_k))}{\partial a_i} \Delta \dot{y}(x) dx + \sum_{i=1}^k \int_{x_0}^{x_1} \left[\int_x^{x_1} \frac{\partial F'(x, z(T_1, s), \dots, z(T_k, s))}{\partial b_i} ds \right] \times$$

$$\begin{aligned}
& \times \Delta \dot{y}(x) dx + \int_{t_0}^{t_1} \int_{x_0}^{x_1} p^0(t, x) \Delta z_t(t, x) dx dt - \int_{t_0}^{t_1} \int_{x_0}^{x_1} \Delta_{\bar{u}(t)} H(t, x) dx dt - \int_{t_0}^{t_1} \int_{x_0}^{x_1} \Delta_{\bar{u}(t)} H'_z(t, x) \Delta z(t, x) dx dt - \\
& - \int_{t_0}^{t_1} \int_{x_0}^{x_1} \left[\int_x^{x_1} H'_z(t, s) ds \right] \Delta \dot{y}(x) dx + \int_{t_0}^{t_1} \int_{x_0}^{x_1} \left[\int_t^{t_1} H'_z(\tau, x) d\tau \right] \Delta z_t(t, x) dx dt + \int_{x_0}^{x_1} q^{0'}(x) \Delta \dot{y}(x) dx - \\
& - \int_{x_0}^{x_1} \Delta_{\bar{v}(x)} M(x) dx - \int_{x_0}^{x_1} \left[\int_x^{x_1} M'_y(s) ds \right] \Delta \dot{y}(x) dx - \frac{1}{2} \int_{t_0}^{t_1} \int_{x_0}^{x_1} \Delta z'(t, x) H_{zz}(t, x) \Delta z(t, x) dx dt - \\
& - \frac{1}{2} \int_{x_0}^{x_1} \Delta y'(x) M_{yy}(x) \Delta y(x) dx - \frac{1}{2} \sum_{i,j=1}^k \Delta y'(X_i) \frac{\partial^2 \varphi(y(X_1), \dots, y(X_k))}{\partial a_i \partial a_j} \Delta y(X_j) + \\
& + \sum_{i=1}^k \int_{t_0}^{t_1} \int_{x_0}^{x_1} \beta_i(t) \frac{\partial F'(x, z(T_1, x), \dots, z(T_k, x))}{\partial b_i} \Delta z_t(t, x) dx dt + \\
& + \frac{1}{2} \sum_{i,j=1}^k \int_{x_0}^{x_1} \Delta z'(T_i, x) \frac{\partial^2 F(x, z(T_1, x), \dots, z(T_k, x))}{\partial b_i \partial b_j} \Delta z(T_j, x) dx - \int_{x_0}^{x_1} \Delta_{\bar{v}(x)} M_y(x) \Delta y(x) dx + \eta_l(\Delta u; \Delta v).
\end{aligned} \tag{15}$$

Если предполагать, что $(p^o(t, x), q^o(x))$ является решением системы интегральных уравнений (сопряженная система):

$$\begin{aligned}
p^o(t, x) &= \int_t^{t_1} H'_z(\tau, x) d\tau - \sum_{i=1}^k \beta_i(t) \frac{\partial F(x, z(T_1, x), \dots, z(T_k, x))}{\partial b_i}, \\
q^o(x) &= - \int_x^{x_1} \frac{\partial F'(s, z(T_1, s), \dots, z(T_k, s))}{\partial b_i} ds + \int_x^{x_1} H_z(t, s) ds + \int_x^{x_1} M_y(s) ds - \\
& - \sum_{i=1}^k \alpha_i(x_i) \frac{\partial \varphi(y(X_1), \dots, y(X_k))}{\partial a_i},
\end{aligned} \tag{16}$$

то формула приращения (15) примет вид

$$\begin{aligned}
\Delta S(u^o, v^o) &= \int_{t_0}^{t_1} \int_{x_0}^{x_1} \Delta_{\bar{u}(x)} H(t, x) dx dt - \int_{x_0}^{x_1} \Delta_{\bar{v}(x)} M(x) dx + \\
& + \frac{1}{2} \sum_{i,j=1}^k \Delta y'(X_i) \frac{\partial^2 \varphi(y(X_1), \dots, y(X_k))}{\partial a_i \partial a_j} \Delta y(X_j) + \\
& + \frac{1}{2} \sum_{i,j=1}^k \int_{x_0}^{x_1} \Delta z'(T_i, x) \frac{\partial^2 F(x, z(T_1, x), \dots, z(T_k, x))}{\partial b_i \partial b_j} \Delta z(T_j, x) dx - \int_{t_0}^{t_1} \int_{x_0}^{x_1} \Delta_{\bar{u}(t)} H'_z(t, x) \Delta z(t, x) dx dt - \\
& - \int_{x_0}^{x_1} \Delta_{\bar{v}(x)} M'_y(x) \Delta y(x) dx - \frac{1}{2} \int_{t_0}^{t_1} \int_{x_0}^{x_1} \Delta z'(t, x) H_{zz}(t, x) \Delta z(t, x) dx dt - \\
& - \frac{1}{2} \int_{x_0}^{x_1} \Delta y'(x) M_{yy}(x) \Delta y(x) dx + \eta_l(\Delta u; \Delta v).
\end{aligned} \tag{17}$$

3. Оценка нормы приращения состояния системы

Используя (3)–(6), перейдем к эквивалентным интегральным уравнениям, будем иметь

$$\Delta z(t, x) = \Delta y(x) + \int_{t_0}^t \left[f(\tau, x, \bar{z}(\tau, x), \bar{u}(\tau)) - f(\tau, x, z^o(\tau, x), u^o(\tau)) \right] d\tau, \tag{18}$$

$$\Delta y(x) = \int_{x_0}^x \left[g(s, \bar{y}(s), \bar{v}(s)) - g(s, y^o(s), v^o(s)) \right] ds. \tag{19}$$

Отсюда получим неравенство

$$\|z(t, x)\| \leq \| \Delta y(x) \| + \int_{t_0}^t \| \Delta_{\bar{u}(\tau)} f(\tau, x) \| d\tau + L_1 \int_{t_0}^t \| \Delta z(\tau, x) \| d\tau, \quad (20)$$

$$\| \Delta y(x) \| \leq \int_{x_0}^x \| \Delta_{v(s)} g(s) \| ds + L_2 \int_{x_0}^x \| \Delta y(s) \| ds, \quad (21)$$

где $L_i = \text{const} > 0$, $i = 1, 2$, – постоянные Липшица. Из (20), (21) применяя аналог леммы Гронуолла–Беллмана (см., например, [3, 4]), приходим к неравенствам

$$\| \Delta z(t, x) \| \leq \| \Delta y(x) \| + L_3 \int_{t_0}^t \| \Delta_{\bar{u}(\tau)} f(\tau, x) \| d\tau, \quad (22)$$

$$\| \Delta y(x) \| \leq L_4 \int_{x_0}^x \| \Delta_{\bar{v}(s)} g(s) \| ds. \quad (23)$$

где L_i , $i = 3, 4$, – некоторые положительные постоянные числа.

С учетом неравенства (23), неравенство (22) запишется в виде

$$\| \Delta z(t, x) \| \leq L_4 \int_{x_0}^x \| \Delta_{\bar{v}(s)} g(s) \| ds + L_3 \int_{t_0}^t \| \Delta_{\bar{u}(\tau)} f(\tau, x) \| d\tau. \quad (24)$$

Таким образом, получили оценку нормы приращения состояния, соответствующую приращению управления.

4. Необходимые условия оптимальности

Считая $(u^o(t), v^o(x))$ оптимальным управлением, его специальное приращение определим по формуле

$$\begin{cases} \bar{u}_\varepsilon(t) = \begin{cases} u, & t \in [\theta, \theta + \varepsilon), \\ u^o(t), & t \in T \setminus [\theta, \theta + \varepsilon), \end{cases} \\ \bar{v}_\mu(x) = v^o(x), \quad x \in X. \end{cases} \quad (25)$$

Здесь $\theta \in [t_0, t_1)$ – произвольная точка непрерывности управления $u^o(t)$; $u \in U$ – произвольный вектор, $\varepsilon > 0$ – произвольное число, такое что $\theta + \varepsilon < t_1$. Через $(\Delta z_\varepsilon(t, x), \Delta y_\varepsilon(x))$ обозначим специальное приращение состояния $(z^o(t, x), y^o(x))$, отвечающее приращению (25) управления $(u^o(t), v^o(x))$. Из оценок (23), (24) следует, что

$$\| \Delta z_\varepsilon^o(t, x) \| \leq L_5 \varepsilon, \quad \| \Delta y_\varepsilon(x) \| \leq L_6 \varepsilon, \quad (26)$$

где $L_5, L_6 = \text{const} > 0$ – некоторые положительные постоянные.

С учетом (25), (26) из формулы приращения (17) получим, что вдоль оптимального управления $(u^o(t), v^o(x))$:

$$\Delta_u S_\varepsilon(u^o, v^o) = -\varepsilon \int_{x_0}^{x_1} \Delta_u H(\theta, x) dx + o(\varepsilon) \geq 0. \quad (27)$$

Теперь специальное приращение оптимального управления определим по формуле

$$\begin{cases} \bar{u}_\varepsilon(t) = u(t), \quad t \in T, \\ \bar{v}_\mu(x) = \begin{cases} v, & x \in [\xi, \xi + \varepsilon), \\ v^o(x), & x \in X \setminus [\xi, \xi + \varepsilon), \end{cases} \end{cases} \quad (28)$$

где $\xi \in [x_0, x_1]$ – произвольная точка непрерывности управляющей функции $v^o(x)$; $v \in V$ – произвольный вектор; $\mu > 0$ – произвольное достаточно малое число, такое что $\xi + \mu < x_1$.

Через $(\Delta z_\mu(t, x), \Delta y_\mu(x))$ обозначим специальное приращение состояния $(z^o(t, x), y^o(x))$, отвечающее приращению (28) управления $(u^o(t), v^o(x))$.

Из оценок (23), (24) следует, что

$$\|\Delta z_\mu(t, x)\| \leq L_7 \mu, \quad \|\Delta y_\mu(x)\| \leq L_8 \mu, \quad (29)$$

где $L_7, L_8 = \text{const} > 0$ – некоторые положительные постоянные.

Принимая во внимание эти оценки и формулу (27), из (17) получим, что

$$\Delta_{\bar{v}} S_\mu(u^o, v^o) = S(u^o, \bar{v}) - S(u^o, v^o) = -\mu \Delta_v M(\xi) + o(\mu) \geq 0. \quad (30)$$

Из неравенств (27), (30), в силу произвольности ε и μ приходим к соотношениям

$$\int_{x_0}^{x_1} \Delta_u H(\theta, x) dx \leq 0, \quad (31)$$

$$\Delta_v M(\xi) \leq 0. \quad (32)$$

Сформулируем полученный результат.

Теорема 1. Для оптимальности допустимого управления $(u^o(t), v^o(x))$ в задаче (1)-(6) необходимо, чтобы неравенства (31), (32) выполнялись соответственно для всех $\theta \in [t_0, t_1]$, $u \in U$ и $\xi \in [x_0, x_1]$, $v \in V$.

Неравенства (31), (32) являются необходимыми условиями первого порядка и пара неравенств (31), (32) представляет собой аналог принципа максимума Понтрягина для рассматриваемой задачи.

Формула приращения (17) позволяет исследовать также случай особых управлений.

Определение 1. Если выполняются соотношения

$$\int_{x_0}^{x_1} \Delta_u H(\theta, x) dx = 0 \text{ для всех } \theta \in [t_0, t_1], u \in U, \quad (33)$$

$$\Delta_v M(\xi) = 0 \text{ для всех } \xi \in [x_0, x_1], v \in V, \quad (34)$$

то допустимое управление $(u^o(t), v^o(x))$ назовем особым, в смысле принципа максимума Понтрягина, управлением.

Из определения 1 видно, что для особых управлений условие максимума (31), (32), вырождаясь, становится неэффективным. Поэтому нужны новые необходимые условия оптимальности. Понятие особого управления было введено Л.И. Розоноэром [5]. В дальнейшем особые управления исследовались Г. Келли [6], Р. Коппом и Г. Мойером [7], Ю.И. Параевым [8], Р. Габасовой, Ф.М. Кирилловой [9] и др. Довольно полный обзор соответствующих результатов имеется в работах [9–16] и др. Заметим, что особые управления возникают во многих прикладных задачах управления (см., например, [6, 7, 9, 17, 18] и др.).

Перейдем к выводу необходимых условий оптимальности для особых управлений.

Из (7), (8) видно, что $(\Delta y(x), \Delta z(t, x))$ является решением линеаризованной задачи

$$\Delta \dot{y}(x) \equiv g_y(x) \Delta y(x) + \Delta_{\bar{v}(x)} g(x) + \eta_2(x; \Delta v), \quad (35)$$

$$\Delta y(x_0) = 0, \quad (36)$$

$$\frac{\partial \Delta z(t, x)}{\partial z} \equiv f_z(t, x) \Delta z(t, x) + \Delta_{\bar{u}(t)} f(t, x) + \eta_3(t, x; \Delta u), \quad (37)$$

$$\Delta z(t_0, x) = \Delta y(x). \quad (38)$$

Здесь по определению

$$\eta_2(x; \Delta v(x)) = \Delta_{\bar{v}(x)} g(x) \Delta y(x) + o_4(\|\Delta y(x)\|),$$

$$\eta_3(t, x; \Delta u(t)) = \Delta_{\bar{u}(t)} f_z(t, x) \Delta z(t, x) + o_5(\|\Delta z(t, x)\|).$$

Решения уравнений (35), (36) и (37), (38) на основе аналога формулы Коши о представлении решений линейных дифференциальных уравнений (см., например, [19, 20]) имеют вид

$$\Delta y(x) = \int_{x_0}^x \Phi(x, s) \Delta_{\bar{v}(s)} g(s) ds + \eta_4(x; \Delta v(x)), \quad (39)$$

$$\Delta z(t, x) = \int_{t_0}^t F(t, \tau, x) \Delta_{\bar{u}(\tau)} f(\tau, x) d\tau + F(t, t_0, x) \Delta y(x) + \eta_5(t, x; \Delta u(t)). \quad (40)$$

Здесь

$$\begin{aligned} \eta_4(x; \Delta v(x)) &= \int_{x_0}^x \Phi(x, s) \eta_2(s; \Delta v(s)) ds, \\ \eta_5(t, x; \Delta u(t)) &= \int_{t_0}^t F(t, \tau, x) \eta_3(\tau, x; \Delta u(\tau)) d\tau, \end{aligned}$$

где $\Phi(x, s)$, $F(t, \tau, x)$ – $(n \times n)$ матричные функции, являющиеся решениями матричных дифференциальных уравнений

$$\begin{aligned} \Phi_s(x, s) &= -\Phi(x, s) g_y(s), \quad \Phi(x, x) = E, \\ F_\tau(t, \tau, x) &= -F(t, \tau, x) f_z(\tau, x), \quad F(t, t, x) = E, \end{aligned}$$

где E – $(n \times n)$ единичная матрица.

Из (39) с учетом представления (40) получаем

$$\Delta z(t, x) = \int_{t_0}^t F(t, \tau, x) \Delta_{\bar{u}(\tau)} f(\tau, x) d\tau + \int_{x_0}^x F(t, t_0, x) \Phi(x, s) \Delta_{\bar{v}(s)} g(s) ds + \eta_6(t, x; \Delta u(t), \Delta v(x)). \quad (41)$$

Здесь по определению

$$\eta_6(t, x; \Delta u(t), \Delta v(x)) = \eta_5(t, x; \Delta u(t)) + F(t, t_0, x) \eta_4(x; \Delta v(x)).$$

С учетом оценок (23), (24) из представлений (39), (41) получаем

$$\Delta z_\varepsilon(t, x) = \int_{t_0}^t F(t, \tau, x) \Delta_{\bar{u}_\varepsilon(\tau)} f(\tau, x) d\tau + o(\varepsilon; t, x), \quad \Delta y_\varepsilon(x) = 0, \quad (42)$$

$$\Delta z_\mu(t, x) = \int_{x_0}^x F(t, t_0, x) \Phi(x, s) \Delta_{\bar{v}_\mu(s)} g(s) ds + o(\mu; t, x), \quad (43)$$

$$\Delta y_\mu(x) = \int_{x_0}^x \Phi(x, s) \Delta_{\bar{v}_\mu(s)} g(s) ds + o(\mu). \quad (44)$$

В особом случае из формулы приращения (13) получаем справедливость разложений

$$\begin{aligned} &S(u^o(t) + \Delta u_\varepsilon(t), v^o(x)) - \Delta S(u^o(t), v^o(x)) = \\ &= \frac{1}{2} \sum_{i,j=1}^k \int_{x_0}^{x_i} \Delta z'_\varepsilon(T_i, x) \frac{\partial^2 F(x, z(T_1, x), \dots, z(T_k, x))}{\partial b_i \partial b_j} \Delta z_\varepsilon(T_j, x) dx - \frac{1}{2} \int_{t_0}^{t_1} \int_{x_0}^{x_1} \Delta z'_\varepsilon(t, x) H_{zz}(t, x) \Delta z_\varepsilon(t, x) dx dt - \\ &\quad - \int_{t_0}^{t_1} \int_{x_0}^{x_1} \Delta_{\bar{u}_\varepsilon(t)} H'_z(t, x) \Delta z_\varepsilon(t, x) dx dt + o(\varepsilon^2), \quad (45) \\ &S(u^o(t), v^o(x) + \Delta v_\mu(x)) - \Delta S(u^o(t), v^o(x)) = \\ &= \frac{1}{2} \sum_{i,j=1}^k \Delta y'_\mu(X_i) \frac{\partial^2 \Phi(y(X_1), y(X_2), \dots, y(X_k))}{\partial a_i \partial a_j} \Delta y_\mu(X_j) + \\ &+ \frac{1}{2} \sum_{i,j=1}^k \int_{x_0}^{x_i} \Delta z'_\mu(T_i, x) \frac{\partial^2 F(x, z(T_1, x), z(T_2, x), \dots, z(T_k, x))}{\partial b_i \partial b_j} \Delta z_\mu(T_j, x) dx - \int_{x_0}^{x_1} \Delta_{\bar{v}_\mu(x)} M'_y(x) \Delta y_\mu(x) dx - \end{aligned}$$

$$-\frac{1}{2} \int_{x_0}^{x_1} \Delta y'_\mu(x) M_{yy}(x) \Delta y_\mu(x) dx - \frac{1}{2} \int_{t_0}^{t_1} \int_{x_0}^{x_1} \Delta z'_\mu(t, x) H_{zz}(t, x) \Delta z_\mu(t, x) dx dt + o(\mu^2). \quad (46)$$

Из (42) следует

$$\Delta z_\varepsilon(T_i, x) = \int_{t_0}^t \alpha_i(\tau) F(T_i, \tau, x) \Delta_{\bar{u}_\varepsilon(\tau)} f(\tau, x) d\tau + o(\varepsilon), \quad (47)$$

где $\alpha_i(t)$ – характеристическая функция отрезка $[t_0, T_i]$.

Используя (47), доказывается справедливость следующих тождеств:

$$\begin{aligned} \int_{x_0}^{x_1} \Delta z'_\varepsilon(T_i, x) \frac{\partial^2 F(x, z(T_1, x), z(T_2, x), \dots, z(T_k, x))}{\partial b_i \partial b_j} \Delta z_\varepsilon(T_j, x) dx = \int_{x_0}^{x_1} \int_{t_0}^{t_1} \int_{t_0}^{t_1} \beta_i(\tau) \beta_j(s) \times \\ \times \Delta_{\bar{u}_\varepsilon(\tau)} f'(\tau, x) F(T_i, \tau, x) \frac{\partial^2 F(x, z(T_1, x), \dots, z(T_k, x))}{\partial b_i \partial b_j} F(T_j, s, x) \Delta_{\bar{u}_\varepsilon(s)} f(s, x) ds d\tau dx + o(\varepsilon^2), \end{aligned} \quad (48)$$

$$\int_{t_0}^{t_1} \int_{x_0}^{x_1} \Delta_{\bar{u}_\varepsilon(t)} H'_z(t, x) \Delta z_\varepsilon(t, x) dx dt = \int_{t_0}^{t_1} \int_{x_0}^{x_1} \left[\int_t^{t_1} \Delta_{\bar{u}_\varepsilon(\tau)} H'_z(\tau, x) F(\tau, t, x) \Delta_{\bar{u}_\varepsilon(t)} f(t, x) d\tau \right] dx dt + o(\varepsilon^2), \quad (49)$$

$$\begin{aligned} \int_{t_0}^{t_1} \int_{x_0}^{x_1} \Delta z'_\varepsilon(t, x) H_{zz}(t, x) \Delta z_\varepsilon(t, x) dx dt = \\ = \int_{t_0}^{t_1} \int_{x_0}^{x_1} \int_{t_0}^{t_1} \Delta_{\bar{u}_\varepsilon(\tau)} f(\tau, x) \left\{ \int_{\max(\tau, s)}^{x_1} F'(t, \tau, x) H_{zz}(t, x) F(t, s, x) dt \right\} \Delta_{\bar{u}_\varepsilon(s)} f(s, x) d\tau ds dx + o(\varepsilon^2). \end{aligned} \quad (50)$$

Далее из представления (44) следует

$$\Delta y_\mu(X_i) = \int_{x_0}^{x_1} \alpha_i(s) \Phi(x_i, s) \Delta_{\bar{v}_\varepsilon(s)} g(s) ds + o(\mu). \quad (51)$$

При помощи представлений (44), (51) получаем

$$\begin{aligned} \Delta y'_\mu(X_i) \frac{\partial^2 \varphi(y(X_1), y(X_2), \dots, y(X_k))}{\partial a_i \partial a_j} \Delta y_\mu(X_j) = \int_{x_0}^{x_1} \int_{x_0}^{x_1} \alpha_i(\alpha) \alpha_j(\beta) \Phi(x_j, \alpha) \times \\ \times \Delta_{\bar{v}_\varepsilon(\beta)} g(\alpha) \frac{\partial^2 \varphi(y(X_1), y(X_2), \dots, y(X_k))}{\partial a_i \partial a_j} \Phi(x_j, \beta) \Delta_{\bar{v}_\varepsilon(\beta)} g(\beta) d\alpha d\beta + o(\mu^2), \end{aligned} \quad (52)$$

$$\begin{aligned} \int_{x_0}^{x_1} \Delta z'_\mu(T_i, x) \frac{\partial^2 F(x, z(T_1, x), z(T_2, x), \dots, z(T_k, x))}{\partial b_i \partial b_j} \Delta z_\mu(T_j, x) dx = \\ = \int_{x_0}^{x_1} \int_{x_0}^{x_1} \Delta_{\bar{v}_\mu(\tau)} g'(\tau) \left\{ \int_{\max(\tau, s)}^{x_1} F'(T_i, t_0, x) \Phi'(x, \tau) \frac{\partial^2 F(x, z(T_1, x), \dots, z(T_k, x))}{\partial b_i \partial b_j} F(T_j, t_0, x) \Phi(x, s) dx \right\} \times \\ \times \Delta_{\bar{v}_\mu(s)} g(s) ds d\tau + o(\mu^2), \end{aligned} \quad (53)$$

$$\int_{x_0}^{x_1} \Delta_{\bar{v}_\mu(x)} M'_y(x) \Delta y_\mu(x) dx = \int_{x_0}^{x_1} \left[\int_x^{x_1} \Delta_{\bar{v}_\mu(s)} M'_y(s) \Phi'(s, x) ds \right] \Delta_{\bar{v}_\mu(x)} g(x) dx + o(\mu^2), \quad (54)$$

$$\begin{aligned} \int_{x_0}^{x_1} \Delta y'_\mu(x) M_{yy}(x) \Delta y_\mu(x) dx = \int_{x_0}^{x_1} \int_{x_0}^{x_1} \Delta_{\bar{v}_\varepsilon(\tau)} g'(\tau) \left\{ \int_{\max(\tau, s)}^{x_1} \Phi'(x, \tau) M_{yy}(x) \Phi(x, s) dx \right\} \Delta_{\bar{v}_\varepsilon(s)} g(s) ds d\tau + o(\mu^2), \\ \int_{t_0}^{t_1} \int_{x_0}^{x_1} \Delta z'_\mu(t, x) H_{zz}(t, x) \Delta z_\mu(t, x) dx dt = \end{aligned} \quad (55)$$

$$= \int_{x_0}^{x_1} \int_{x_0}^{x_1} \Delta_{\bar{v}_\mu(\tau)} g'(\tau) \left\{ \int_{t_0}^{t_1} \int_{\max(\tau, s)}^{x_1} \Phi'(x, \tau) F'(t, t_0, x) H_{zz}(t, x) F(t, t_0, x) \Phi(x, s) dt dx \right\} \Delta_{\bar{v}_\mu(s)} g(s) ds d\tau + o(\mu^2).$$

Введем обозначения:

$$K(\tau, \alpha, x) = - \sum_{i,j=1}^k \beta_i(\tau) \beta_j(s) F'(T_i, \tau, x) \frac{\partial^2 G(x, z^o(T_i, x))}{\partial b_i \partial b_j} F(T_j, \alpha, x) + \\ + \int_{\max(\tau, \alpha)}^{t_1} F(t, \tau, x) \frac{\partial^2 H(t, x)}{\partial z^2} F(t, \alpha, x) dt, \quad (56)$$

$$L(\tau, s) = - \sum_{i,j=1}^k \alpha_i(\tau) \alpha_j(s) \Phi'(x_i, \tau) \frac{\partial^2 \Phi(y(X_1), y(X_2), \dots, y(X_k))}{\partial a_i \partial a_j} \Phi(x_j, s) + \\ + \int_{\max(\tau, s)}^{x_1} F'(T_i, t_0, x) \Phi'(x, \tau) \frac{\partial^2 F(x, z(T_1, x), z(T_2, x), \dots, z(T_k, x))}{\partial b_i \partial b_j} F(T_j, t_0, x) \Phi(x, s) dx + \\ + \int_{\max(\tau, s)}^{x_1} \Phi'(x, \tau) M_{yy}(x) \Phi(x, s) dx + \int_{t_0}^{t_1} \int_{\max(\tau, s)}^{x_1} \Phi'(x, \tau) F'(t, t_0, x) H_{zz}(t, x) F(t, t_0, x) \Phi(x, s) dt dx. \quad (57)$$

Учитывая введенные обозначения (56), (57) и тождества (52)–(57), формулы приращения (45), (46) после некоторых преобразований запишем следующим образом:

$$S(u^o(t) + \Delta u_\varepsilon(t), v^o(x)) - \Delta S(u^o(t), v^o(x)) = \\ = - \frac{\varepsilon^2}{2} \int_{x_0}^{x_1} [\Delta_u f'(\theta, x) K(\theta, \theta, x) \Delta_u f(\theta, x) + \Delta_u H'_z(\theta, x) \Delta_u f(\theta, x)] dx + o(\varepsilon^2), \quad (58)$$

$$S(u^o(t), v^o(x) + \Delta v_\mu(x)) - \Delta S(u^o(t), v^o(x)) = \\ = - \frac{\mu^2}{2} [\Delta_v g'(\xi) L(\xi, \xi) \Delta_v g(\xi) + \Delta_v M'_y(\xi) \Delta_v g(\xi)] + o(\mu^2). \quad (59)$$

Учитывая произвольность и независимость $\varepsilon > 0$ и $\mu > 0$, из разложений (58), (59) получаем, что вдоль особого оптимального управления $(u^o(t), v^o(x))$ выполняются соотношения

$$\int_{x_0}^{x_1} [\Delta_u f'(\theta, x) K(\theta, \theta, x) \Delta_u f(\theta, x) + \Delta_u H'_z(\theta, x) \Delta_u f(\theta, x)] dx \leq 0 \quad (60)$$

для всех $\theta \in [t_0, t_1]$, $u \in U$,

$$\Delta_v g'(\xi) L(\xi, \xi) \Delta_v g(\xi) + \Delta_v M'_y(\xi) \Delta_v g(\xi) \leq 0, \quad (61)$$

для всех $v \in V$, $\xi \in [x_0, x_1]$.

Сформулируем результат.

Теорема 2. Для оптимальности особого, в смысле принципа максимума Понтрягина, управления $(u^o(t), v^o(x))$ необходимо, чтобы неравенства (60), (61) выполнялись для всех $u \in U$, $\theta \in [t_0, t_1]$ и $v \in V$, $\xi \in [x_0, x_1]$ соответственно.

Заметим, что необходимые условия оптимальности (60), (61) являются аналогами условия оптимальности Габасова–Кирилловой (см., например, [9, 10]).

Заключение

В работе изучается задача оптимального управления типа А.И. Москаленко для многоточечного критерия качества. Применяя схему, предложенную в работах [11, 14, 15 и др.], установлено необходимое условие оптимальности особых, в смысле принципа максимума Понтрягина, управлений.

ЛИТЕРАТУРА

1. Москаленко А.И. Об одном классе задач оптимального регулирования // Журнал вычислительной математики и математической физики. 1969. № 1. С. 69–95.
2. Москаленко А.И. Некоторые вопросы теории оптимального управления : автореф. дис. ... канд. физ.-мат. наук. Томск : Том. гос. ун-т, 1971. 20 с.
3. Плотников В.И., Сумин В.И. Проблема устойчивости нелинейных систем Гурса-Дарбу // Дифференциальные уравнения. 1972. № 5. С. 845–856.
4. Плотников В.И., Сумин В.И. Оптимизация объектов с распределенными параметрами, описываемых системами Гурса-Дарбу // Журнал вычислительной математики и математической физики. 1972. № 1. С. 61–77.
5. Розоноэр Л.И. Принцип максимума в теории оптимальных систем I-III // Автоматика и телемеханика. 1969. № 10–12. С. 1320–1334, 1441–1458, 1561–1578.
6. Келли Г. Необходимое условие для особых экстремалей, основанное на второй вариации // Ракетная техника и космонавтика. 1964. № 8. С. 26–29.
7. Копп Р., Мойер Г. Необходимое условие оптимальности особых экстремалей // Ракетная техника и космонавтика. 1965. № 8. С. 84–91.
8. Параев Ю.И. Об особом управлении в оптимальных процессах, линейных относительно управляющих воздействий // Автоматика и телемеханика. 1962. № 9. С. 1202–1209.
9. Габасов Р., Кириллова Ф.М. Особые оптимальные управления. М. : Наука, 1973. 256 с.
10. Габасов Р., Кириллова Ф.М., Мансимов К.Б. Необходимые условия оптимальности высокого порядка (обзор) // Препринт ИМ АН БССР. 1982. № 30 (155). 48 с.
11. Мансимов К.Б. Особые управления в системах с запаздыванием. Баку : Изд-во ЭЛМ, 1999. 171 с.
12. Марданов М.Дж. Об условиях оптимальности особых управлений // Докл. АН СССР. 1980. Т. 213, № 4. С. 815–818.
13. Меликов Т.К. Об оптимальности особых управлений в системах с последствием нейтрального типа // Журнал вычислительной математики и математической физики. 2001. № 9. С. 1332–1343.
14. Марданов М.Дж., Мансимов К.Б., Меликов Т.К. Исследование особых управлений и необходимые условия оптимальности второго порядка в системах с запаздыванием. Баку : Изд-во ЭЛМ, 2013. 356 с.
15. Мансимов К.Б., Марданов М.Дж. Качественная теория оптимального управления системами Гурса-Дарбу. Баку : Изд-во ЭЛМ, 2010. 363 с.
16. Марданов М.Дж. Исследование оптимальных процессов с запаздываниями при наличии ограничений. Баку : Изд-во ЭЛМ, 2010. 114 с.
17. Параев Ю.И. Оптимальное управление двухсекторной экономикой // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2014. № 3 (28). С. 4–11.
18. Параев Ю.И., Грекова Т.И., Данилюк Е.Ю. Аналитическое решение задачи оптимального управления односекторной экономикой на конечном интервале времени // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2011. № 4(17). С. 5–15.
19. Габасов Р., Кириллова Ф.М. Оптимизация линейных систем. Минск : Изд-во БГУ, 1973. 246 с.
20. Алексеев В.М., Тихомиров В.М., Фомин С.В. Оптимальное управление. М. : Наука, 1979. 419 с.

Мансимов Камил Байрамали оглы, д-р физ.-мат. наук, профессор. E-mail: kmansimov@mail.ru
 Бакинский государственный университет, Институт систем управления НАН Азербайджана (г. Баку, Азербайджан)
Расулова Шахла Меджид кызы. E-mail: rasulzade_sh@yahoo.com
 Институт Систем Управления НАН Азербайджана (г. Баку, Азербайджан)

Поступила в редакцию 11 октября 2016 г.

Mansimov Kamil B. (Baku State University, Institute of Control Systems of Azerbaijan National Academy of Sciences).

Rasulova Shahla M. (Institute of Control Systems of Azerbaijan National Academy of Sciences).

About one A.I. Moskalenko control problem.

Keywords: Pontryagin maximum condition; special case; the necessary optimality condition; optimal control problem with distributed parameters.

DOI: 10.17223/19988605/40/2

Suppose you want to minimize the multi-point functional

$$S(u) = \varphi(y(X_1), y(X_2), \dots, y(X_k)) + \int_{x_0}^{x_1} F(x, z(T_1, x), z(T_2, x), \dots, z(T_k, x)) dx \quad (1)$$

with constraints

$$u(t) \in U \subset R^r, \quad t \in T = [t_0, t_1], \quad v(x) \in V \subset R^q, \quad x \in X = [x_0, x_1], \quad (2)$$

$$\frac{\partial z(t, x)}{\partial t} = f(t, x, z(t, x), u(t)), \quad (t, x) \in D = T \times X, \quad (3)$$

$$z(t_0, x) = y(x), \quad x \in X, \quad (4)$$

$$\frac{dy}{dx} = g(x, y(x), v(x)), \quad x \in X, \quad (5)$$

$$y(x_0) = y_0. \quad (6)$$

Here $f(t, x, z, u) (g(x, y, v))$ is a given n -dimensional vector-function defined and continuous in $D \times R^n \times R^r (X \times R^n \times R^q)$ together with partial derivatives with respect to $z(y)$ up to the second order inclusive, $T_i \in (t_0, t_1] \quad i = \overline{1, k}$, $(t_0 < T_1 < T_2 < \dots < T_k \leq t_1)$, $X_i \in (x_0, x_1] \quad (x_0 < X_1 < X_2 < \dots < X_k \leq x_1)$ are given points, $\varphi(a_1, a_2, \dots, a_k) (F(x, b_1, b_2, \dots, b_k))$ is a given and continuous in $R^{k-n} (X \times R^{k-n})$ together with the partial derivatives up to the second order inclusive a scalar function, U and V are non-empty and bounded sets, $u(t) (v(x))$ is a piecewise continuous (with a finite number of points of discontinuity of the first kind) vector-function of control actions.

Paper deals obtaining the necessary conditions for optimality in the problem (1)–(6).

REFERENCES

1. Moskalenko, A.I. (1969) Ob odnom klasse zadach optimal'nogo regulirovaniya [On a class of optimal control problems]. *Zhurnal vychislitel'noy matematiki i matematicheskoy fiziki – Journal. Comput. mat. and mat. physicists.* 1. pp. 69–95.
2. Moskalenko, A.I. (1971) *Nekotorye voprosy teorii optimal'nogo upravleniya* [Some questions in the optimal control theory]. Abstract of Physics and Mathematics Cand. Diss. Tomsk.
3. Plotnikov, V.I. & Sumin, V.I. (1972) Problema ustoychivosti nelineynykh sistem Gursa-Darbu [The problem of stability of nonlinear systems Goursat-Darboux]. *Differentsial'nye uravneniya – Differential Equations.* 5. pp. 845–856.
4. Plotnikov, V.I. & Sumin, V.I. (1972) Optimizatsiya ob'ektov s raspredelennymi parametrami, opisyyaemye sistemami Gursa-Darbu [Optimization of objects with distributed parameters, described by Goursat-Darboux systems]. *Zhurnal vychislitel'noy matematiki i matematicheskoy fiziki.* 1. pp. 61–77.
5. Rozonoer, L.I. (1969) The maximum principle in the theory of optimal systems I-III. *Automation and Remote Control.* 10–12. pp. 1320–1334, 1441–1458, 1561–1578.
6. Kelly, G. (1964) A necessary condition for the singular extremals, based on the second variation. *Rocketry and Astronautics.* 8. pp. 26–29.
7. Kopp, R. & Moyer, H. (1965) Necessary condition for optimality of singular extremals. *Rocketry and Astronautics.* no. 8. pp. 84–91.
8. Paraev, Yu.I. (1962) On special management of optimal processes, linear with respect to control actions. *Automation and Remote Control.* 9. pp. 1202–1209.
9. Gabasov, R. & Kirillova, F.M. (1973) *Osobyie optimal'nye upravleniya* [Special optimal control]. Moscow: Nauka.
10. Gabasov, R., Kirillova, F.M. & Mansimov, K.B. (1982) Neobkhodimye usloviya optimal'nosti vysokogo poryadka (obzor) [Necessary conditions for optimal-of-order (review)]. *Preprint BSSR.* 30(155).
11. Mansimov, K.B. (1999) *Osobyie upravleniya v sistemakh s zapazdyvaniem* [Specific management systems with delay]. Baku: ELM.
12. Mardanov, M.C. (1980) Ob usloviyakh optimal'nosti osobykh upravleniy [On optimality conditions for singular controls]. *Dokl. USSR Academy of Sciences.* 213(4). pp. 815–818.
13. Melikov, T.G. (2001) Ob optimal'nosti osobykh upravleniy v sistemakh s posledeystviem neytral'nogo tipa [On the optimality of singular controls in the system and aftereffect neutral type]. *Zhurnal vychislitel'noy matematiki i matematicheskoy fiziki – Journal. Comput. mat. and mat. physicists.* 9. pp. 1332–1343.
14. Mardanov, M.C., Mansimov, K.B. & Melikov, T.G. (2013) *Issledovanie osobykh upravleniy i neobkhodimye usloviya optimal'nosti vtorogo poryadka v sistemakh s zapazdyvaniem* [Investigation of special departments and the necessary conditions for optimality in the second order systems with delay]. Baku: ELM.
15. Mansimov, K.B. & Mardanov, M.C. (2010) *Kachestvennaya teoriya optimal'nogo upravleniya sistemami Gursa-Darbu* [Qualitative theory of optimal control of the Goursat-Darboux systems]. Baku: ELM.
16. Mardanov, M.C. (2010) *Issledovanie optimal'nykh protsessov s zapazdyvaniyami pri nalichii ogranicheniy* [Study of optimal processes with delays with restrictions]. Baku: ELM.
17. Parayev, Yu.I. (2014) Optimal control of a two-sector economy. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science.* 3(28). pp. 4–11. (In Russian).
18. Parayev, Yu.I., Grekova, T.I. & Danliyuk, E.Yu. (2011) The analytical solution of the optimal control problem sector economy on a finite interval of time. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science.* 4(17). pp. 5–15. (In Russian).
19. Gabasov, R. & Kirillova, F.M. (1973) *Optimizatsiya lineynykh sistem* [Optimization of linear systems]. Minsk: BSU.
20. Alakceyev, V.M., Tikhomirov, V.M. & Fomin, C.V. (1979) *Optimal'noe upravlenie* [Optimal control]. Moscow: Nauka.

В.И. Смагин

АДАПТИВНОЕ ПРОГНОЗИРУЮЩЕЕ УПРАВЛЕНИЕ В ДИСКРЕТНЫХ СИСТЕМАХ С НЕИЗВЕСТНЫМ ВХОДОМ

Работа выполнена при финансовой поддержке РФФИ № 17-08-00920.

Рассматривается задача синтеза адаптивного прогнозирующего управления, построенного на основе слежения за желаемым выходом системы при наличии в описании модели неизвестных параметров и неизвестного входа. Прогнозирование осуществляется на основе вычисления оценок состояний объекта, параметров и неизвестного входа. Рассматривается задача адаптивного управления экономической системой производства, хранения и поставок товара потребителям.

Ключевые слова: дискретные системы; прогнозирующее управление; адаптивное управление; неизвестный вход; модель производства.

При синтезе управлений широко используется метод управления динамическими объектами с применением прогнозирующих моделей – Model Predictive Control (МРС) [1]. Область применения МРС охватывает задачи управления технологическими процессами, производственными системами, управления запасами, задачи управления портфелем ценных бумаг [1–6] и др.

В работе рассматривается задача синтеза адаптивного прогнозирующего управления для динамических объектов с неизвестными параметрами и неизвестным входом, при этом применяются методы идентификации неизвестных параметров объекта [7, 8], методы вычисления оценок вектора состояния объекта, использующие оценки неизвестного входа [9–12]. Дано применение метода к задаче синтеза адаптивного прогнозирующего управления производством, хранением и поставками товара с учетом возможного наличия в модели объекта неизвестного входа.

1. Постановка задачи

Модель объекта, наблюдений и выхода описывается следующими линейно-разностными уравнениями:

$$x_{t+1} = A(\theta_t)x_t + B(\theta_t)u_t + I r_t + w_t, \quad x_{t|t=0} = x_0, \quad (1)$$

$$\psi_t = Hx_t + v_t, \quad (2)$$

$$y_t = Gx_t, \quad (3)$$

где $x_t \in R^n$ – состояние объекта; $u_t \in R^m$ – управляющее воздействие (известный вход); $\psi_t \in R^l$ – наблюдения; $r_t \in R^q$ – неизвестный вход; $y_t \in R^p$ – управляемый выход; θ_t – неизвестный вектор; $A(\theta_t)$, $B(\theta_t)$, I , H , G – матрицы соответствующих размерностей. Случайные возмущения w_t , шумы измерения v_t и вектор начальных условий x_0 подчиняются гауссовскому распределению с характеристиками:

$$M\{w_t\} = 0, \quad M\{v_t\} = 0, \quad M\{w_t w_k^T\} = W \delta_{t,k}, \quad M\{v_t v_k^T\} = V \delta_{t,k}, \quad M\{w_t v_k^T\} = 0,$$

$$M\{x_0\} = \bar{x}_0, \quad M\{(x_0 - \bar{x}_0)(x_0 - \bar{x}_0)^T\} = P_{x_0}.$$

Ограничения на векторы состояния и управления представимы в виде

$$a_1(t) \leq S_1 x_t \leq a_2(t), \quad \varphi_1(x_t, t) \leq S_2 u_t \leq \varphi_2(x_t, t), \quad (4)$$

где S_1 и S_2 – структурные матрицы полного ранга, состоящие из нулей и единиц, определяющие компоненты векторов x_t и u_t , на которые накладываются ограничения; $a_1(t)$, $a_2(t)$, $\varphi_1(x_t, t)$, $\varphi_2(x_t, t)$ – заданные векторы и вектор-функции соответствующих размерностей.

Задача состоит в том, чтобы по наблюдениям ψ_t определить стратегию управления в условиях неполной информации о модели, при которой вектор выхода системы y_t будет близок к заданному вектору $\bar{y}_t$.

2. Прогнозирование поведения объекта в условиях неполной информации о модели

Прогнозирование поведения объекта и вектора выхода системы осуществляется с использованием экстраполятора Калмана на основе оценок вектора состояния $\hat{x}_{t|t}$, вектора выхода $\hat{y}_{t|t}$ и оценки вектора неизвестного входа $\hat{r}_t$:

$$\hat{x}_{t+1|t} = \hat{A}_t \hat{x}_{t|t-1} + \hat{B}_t u_t + \hat{I}_t r_t + K_t (\psi_t - H \hat{x}_{t|t-1}), \quad \hat{x}_{0|-1} = \bar{x}_0, \quad (5)$$

$$\hat{y}_{t+1|t} = G \hat{x}_{t+1|t}, \quad (6)$$

$$K_t = \hat{A}_t P_t H^T (H P_t H^T + V)^{-1}, \quad (7)$$

$$P_{t+1} = W + \hat{A}_t P_t \hat{A}_t^T - \hat{A}_t P_t H^T (H P_t H^T + V)^{-1} H P_t \hat{A}_t^T, \quad P_0 = P_{x_0}, \quad (8)$$

где $\hat{A}_t = A(\hat{\theta}_t)$ и $\hat{B}_t = B(\hat{\theta}_t)$ – матрицы, полученные в результате идентификации модели.

Идентификацию модели можно осуществить с помощью фильтра Калмана и модифицированного метода наименьших квадратов.

3. Идентификация модели с использованием фильтра Калмана

Идентификацию модели осуществим с помощью фильтра Калмана. В этом случае для идентификации необходимо иметь определенный уровень априорной информации о неизвестном векторе θ_t . Предположим, что закон поведения неизвестного вектора задается следующим уравнением:

$$\theta_{t+1} = \Theta_t \theta_t + \tau_t, \quad \theta_{t|t=0} = \theta_0, \quad (9)$$

где Θ_t – матрица, определяющая динамику неизвестного вектора в момент времени t ; τ_t – случайная гауссовская величина с нулевым средним и ковариацией $M\{\tau_t \tau_k^T\} = T_t \delta_{t,k}$; θ_0 – случайный вектор начальных условий с известными характеристиками

$$M\{\theta_0\} = \bar{\theta}_0, \quad M\{(\theta_0 - \bar{\theta}_0)(\theta_0 - \bar{\theta}_0)^T\} = P_{\theta_0}.$$

Будем предполагать, что матрицы $A(\theta_t)$ и $B(\theta_t)$ линейно зависят от θ_t . Тогда можно представить вектор x_{t+1} в виде линейной функции от θ_t :

$$x_{t+1} = A(\theta_t)x_t + B(\theta_t)u_t + I_t r_t + w_t = \Omega(x_t, u_t)\theta_t + \rho(x_t, u_t) + I_t r_t + w_t. \quad (10)$$

Умножив (10) слева на матрицу H и добавив v_{t+1} к левой и правой частям (10), получим

$$\psi_{t+1} = Hx_{t+1} + v_{t+1} = H\Omega(x_t, u_t)\theta_t + H\rho(x_t, u_t) + Hw_t + HI_t r_t + v_{t+1}. \quad (11)$$

Далее, используя алгоритм фильтрации Калмана к модели (9) и каналу наблюдений (11), при этом заменив в (11) x_t и r_t на их оценки, получим оценку неизвестного параметра в момент времени $t+1$, которая определяется из следующих выражений:

$$\hat{\theta}_{t+1} = \Theta_t \hat{\theta}_t + \bar{K}_t (\psi_{t+1} - H\hat{\Omega}_t \hat{\theta}_t - H\hat{\rho}_t - \hat{r}_t), \quad \hat{\theta}_0 = \bar{\theta}_0, \quad (12)$$

$$\bar{K}_t = \bar{P}_t \hat{\Omega}_t^T (\hat{\Omega}_t \bar{P}_t \hat{\Omega}_t^T + HWH^T + V)^{-1}, \quad (13)$$

$$\bar{P}_t = \Theta_t \bar{P}_t \Theta_t^T + T_t, \quad \bar{P}_{t+1} = (E_n - \bar{K}_t \hat{\Omega}_t) \bar{P}_t, \quad \bar{P}_0 = P_{\theta_0}, \quad (14)$$

где $\hat{\Omega}_t = \Omega(\hat{x}_{t|t-1}, u_t)$, $\hat{\rho}_t = \rho(\hat{x}_{t|t-1}, u_t)$. Оценка вектора состояния $\hat{x}_{t+1|t}$ определяется по формуле

$$\hat{x}_{t+1|t} = A(\hat{\theta}_t) \hat{x}_{t|t-1} + B(\hat{\theta}_t) u_t + \hat{I}_t r_t = \hat{\Omega}_t \hat{\theta}_t + \hat{\rho}_t + \hat{I}_t r_t, \quad (15)$$

где $\hat{A}_t = A(\hat{\theta}_t)$, $\hat{B}_t = B(\hat{\theta}_t)$.

Для построения оценок неизвестного входа будем использовать модифицированный метод МНК [9], в основу которого положена минимизация следующего критерия:

$$J(\hat{r}_{t-1}) = \sum_{i=1}^t \left\{ \left\| \psi_i - H\hat{x}_{i|t-1} \right\|_{C_R}^2 + \left\| \hat{r}_{i-1} \right\|_{D_R}^2 \right\}, \quad (16)$$

где C_R и D_R – симметричные, положительно определенные матрицы.

Оптимизация критерия до текущего момента времени t сводится к минимизации критерия в каждый момент времени $i = \overline{1, t}$:

$$J(\hat{r}_{t-1}) = \min_{\hat{r}_0} \min_{\hat{r}_1} \dots \min_{\hat{r}_{t-1}} \sum_{i=1}^t \left\{ \left\| \psi_i - H\hat{x}_{i|t-1} \right\|_{C_R}^2 + \left\| \hat{r}_{i-1} \right\|_{D_R}^2 \right\}. \quad (17)$$

Оптимальная оценка неизвестного входа на шаге $t=1$

$$J(\hat{r}_0) = \min_{\hat{r}_0} \left\{ \left\| \psi_1 - H\hat{x}_{1|0} \right\|_{C_R}^2 + \left\| \hat{r}_0 \right\|_{D_R}^2 \right\}.$$

Учитывая, что $\hat{x}_{1|0} = Ax_0 + Bu_0 + I\hat{r}_0$, имеем

$$J(\hat{r}_0) = \min_{\hat{r}_0} \left\{ \left\| \psi_1 - HAx_0 - HBu_0 - H\hat{r}_0 \right\|_{C_R}^2 + \left\| \hat{r}_0 \right\|_{D_R}^2 \right\}. \quad (18)$$

После преобразований получаем

$$J(\hat{r}_0) = \min_{\hat{r}_0} \left\{ \hat{r}_0^T \left(I^T H^T C_R H I + D_R \right) \hat{r}_0 - 2\hat{r}_0^T I^T H^T C_R (\psi_1 - HAx_0 - HBu_0) + \alpha_0 \right\},$$

где α_0 – величина, не зависящая от $\hat{r}_0$.

Оптимальная оценка находится из условия

$$\frac{\partial J(\hat{r}_0)}{\partial \hat{r}_0} = 2 \left(I^T H^T C_R H I + D_R \right) \hat{r}_0 - 2I^T H^T C_R (\psi_1 - HAx_0 - HBu_0) = 0,$$

откуда получаем оптимальную оценку неизвестного входа в момент времени $t=1$:

$$\hat{r}_0 = S_R (\psi_1 - HAx_0 - HBu_0), \quad (19)$$

где $S_R = (I^T H^T C_R H I + D_R)^{-1} I^T H^T C_R$. Подставляя полученное выражение для $\hat{r}_0$ в (18), вычисляем оптимальное значение критерия в момент времени $t=1$:

$$J(\hat{r}_0) = (\psi_1 - HAx_0 - HBu_0)^T M_R (\psi_1 - HAx_0 - HBu_0), \quad (20)$$

где $M_R = C_R - 2C_R H I S_R + S_R^T (I^T H^T C_R H I + D_R) S_R$.

В момент времени $t=2$ оптимальная оценка неизвестного входящего сигнала находится исходя из оптимизации следующего критерия:

$$J(\hat{r}_1) = \min_{\hat{r}_0} \min_{\hat{r}_1} \left\{ \left\| \psi_2 - H\hat{x}_{2|1} \right\|_{C_R}^2 + \left\| \hat{r}_1 \right\|_{D_R}^2 + \left\| \psi_1 - H\hat{x}_{1|0} \right\|_{C_R}^2 + \left\| \hat{r}_0 \right\|_{D_R}^2 \right\}.$$

Выражение для $J(\hat{r}_1)$ может быть преобразовано следующим образом:

$$\begin{aligned} J(\hat{r}_1) &= \min_{\hat{r}_1} \left\{ \left\| \psi_2 - H\hat{x}_{2|1} \right\|_{C_R}^2 + \left\| \hat{r}_1 \right\|_{D_R}^2 + J(\hat{r}_0) \right\} = \\ &= \min_{\hat{r}_1} \left\{ \left\| \psi_2 - H\hat{x}_{1|0} - HBu_1 - H\hat{r}_1 \right\|_{C_R}^2 + \left\| \hat{r}_1 \right\|_{D_R}^2 + \left\| \psi_1 - HAx_0 - HBu_0 \right\|_{M_R}^2 \right\} = \\ &= \min_{\hat{r}_1} \left\{ \hat{r}_1^T \left(I^T H^T C_R H I + D_R \right) \hat{r}_1 - 2\hat{r}_1^T I^T H^T C_R (\psi_2 - H\hat{x}_{1|0} - HBu_1) + \alpha_1 \right\}, \end{aligned}$$

где α_1 – величина, не зависящая от $\hat{r}_1$. Дифференцируя по $\hat{r}_1$, по аналогии с операциями, проведенными на первом шаге, имеем

$$\hat{r}_1 = S_R (\psi_2 - H\hat{x}_{1|0} - HBu_1), \quad (21)$$

$$J(\hat{r}_1) = (\psi_2 - H\hat{x}_{1|0} - HBu_1)^T M_R (\psi_2 - H\hat{x}_{1|0} - HBu_1). \quad (22)$$

Для последующих шагов применяя метод математической индукции, получим

$$\hat{r}_t = S_R (\psi_{t+1} - H\hat{x}_{t|t-1} - HBu_t). \quad (23)$$

5. Синтез адаптивного управления

Реализуем прогнозирование поведения объекта и выхода системы с использованием следующих формул:

$$\hat{x}_{t+i|t} = \left(\prod_{k=1}^{i-1} \hat{A}_{t+i-k} \right) \hat{x}_{t+1|t} + \left(\prod_{l=1}^{i-1} \hat{A}_{t+i-l} \right) \hat{B}_{t+i-k} u_{t+i-k|t} + \left(\prod_{l=1}^{i-1} \hat{A}_{t+i-l} \right) I \hat{r}_{t+i-k}, \quad (24)$$

$$\hat{y}_{t+i|t} = G_{t+i} \left(\prod_{k=1}^{i-1} \hat{A}_{t+i-k} \right) \hat{x}_{t+1|t} + \sum_{k=1}^{i-1} G_{t+i} \left(\prod_{l=1}^{k-1} \hat{A}_{t+i-l} \right) \hat{B}_{t+i-k} u_{t+i-k|t} + \sum_{k=1}^{i-1} G_{t+i} \left(\prod_{l=1}^{k-1} \hat{A}_{t+i-l} \right) I \hat{r}_{t+i-k}, \quad (25)$$

где $u_{t+k|t}$ – управление, используемое для прогнозирования.

Уравнения для прогнозируемых векторов состояния и выхода представляются в векторно-матричной форме

$$\begin{aligned} \hat{X}_t &= \hat{\Psi} \hat{x}_{t+1|t} + \hat{P} U_t + \hat{S} \hat{R}_t, \\ \hat{Y}_t &= \hat{\Lambda} \hat{x}_{t+1|t} + \hat{\Phi} U_t + \hat{Q} \hat{R}_t, \end{aligned} \quad (26)$$

где векторы $\hat{X}_t, \hat{Y}_t, U_t, \hat{R}_t$ имеют вид

$$\hat{X}_t = \begin{bmatrix} \hat{x}_{t+1|t} \\ \vdots \\ \hat{x}_{t+N|t} \end{bmatrix}, \quad \hat{Y}_t = \begin{bmatrix} \hat{y}_{t+1|t} \\ \vdots \\ \hat{y}_{t+N|t} \end{bmatrix}, \quad U_t = \begin{bmatrix} u_{t+1|t} \\ \vdots \\ u_{t+N|t} \end{bmatrix}, \quad \hat{R}_t = \begin{bmatrix} \hat{r}_{t+1|t} \\ \vdots \\ \hat{r}_{t+N|t} \end{bmatrix},$$

матрицы $\hat{\Psi}_t, \hat{\Lambda}_t, \hat{S}, \hat{Q}$ определяются следующим образом:

$$\begin{aligned} \hat{\Psi}_t &= \begin{bmatrix} E_n \\ \hat{A}_{t+1} \\ \hat{A}_{t+2} \hat{A}_{t+1} \\ \vdots \\ \prod_{k=1}^{N-1} \hat{A}_{t+N-k} \end{bmatrix}, \quad \hat{\Lambda}_t = \begin{bmatrix} G_{t+1} \\ G_{t+2} \hat{A}_{t+1} \\ G_{t+3} \hat{A}_{t+2} \hat{A}_{t+1} \\ \vdots \\ G_{t+N} \prod_{k=1}^{N-1} \hat{A}_{t+N-k} \end{bmatrix}, \\ \hat{S} &= \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ I & 0 & 0 & \dots & 0 \\ \hat{A}I & I & 0 & \dots & 0 \\ \dots & \dots & \dots & \ddots & \dots \\ \hat{A}^{N-2}I & \hat{A}^{N-3}I & \dots & I & 0 \end{bmatrix}, \quad \hat{Q} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ GI & 0 & 0 & \dots & 0 \\ G\hat{A}I & GI & 0 & \dots & 0 \\ \dots & \dots & \dots & \ddots & \dots \\ G\hat{A}^{N-2}I & G\hat{A}^{N-3}I & \dots & GI & 0 \end{bmatrix}. \end{aligned} \quad (27)$$

В случае $M < N$ матрицы $\hat{P}_t$ и $\hat{\Phi}_t$ вводятся следующим образом:

$$\hat{P}_t = \begin{bmatrix} 0 & 0 & \dots & 0 \\ \hat{B}_{t+1} & 0 & \dots & 0 \\ \hat{A}_{t+2} \hat{B}_{t+1} & \hat{B}_{t+2} & \dots & \dots \\ \dots & \dots & \ddots & 0 \\ \left(\prod_{k=1}^{M-1} \hat{A}_{t+M-k+1} \right) \hat{B}_{t+1} & \left(\prod_{k=1}^{M-2} \hat{A}_{t+M-k+1} \right) \hat{B}_{t+2} & \dots & \hat{B}_{t+M} \\ \left(\prod_{k=1}^M \hat{A}_{t+M-k+2} \right) \hat{B}_{t+1} & \left(\prod_{k=1}^{M-1} \hat{A}_{t+M-k+1} \right) \hat{B}_{t+2} & \dots & \hat{A}_{t+M+1} \hat{B}_{t+M} \\ \dots & \dots & \dots & \dots \\ \left(\prod_{k=1}^{N-2} \hat{A}_{t+N-k} \right) \hat{B}_{t+1} & \left(\prod_{k=1}^{N-3} \hat{A}_{t+N-k} \right) \hat{B}_{t+2} & \dots & \left(\sum_p^{N-M-1} \prod_{k=1}^p \hat{A}_{t+N-k} \right) \hat{B}_{t+M} \end{bmatrix},$$

$$\hat{\Phi}_t = \begin{bmatrix} 0 & 0 & \dots & 0 \\ G\hat{B}_{t+1} & 0 & \dots & 0 \\ G\hat{A}_{t+2}B_{t+1} & G\hat{B}_{t+2} & \dots & \dots \\ \dots & \dots & \ddots & 0 \\ G\left(\prod_{k=1}^{M-1}\hat{A}_{t+M-k+1}\right)\hat{B}_{t+1} & G\left(\prod_{k=1}^{M-2}\hat{A}_{t+M-k+1}\right)\hat{B}_{t+2} & \dots & G\hat{B}_{t+M} \\ G\left(\prod_{k=1}^M\hat{A}_{t+M-k+2}\right)\hat{B}_{t+1} & G\left(\prod_{k=1}^{M-1}\hat{A}_{t+M-k+1}\right)\hat{B}_{t+2} & \dots & G\hat{A}_{t+M+1}\hat{B}_{t+M} \\ \dots & \dots & \dots & \dots \\ G\left(\prod_{k=1}^{N-2}\hat{A}_{t+N-k}\right)\hat{B}_{t+1} & G\left(\prod_{k=1}^{N-3}\hat{A}_{t+N-k}\right)\hat{B}_{t+2} & \dots & G\left(\sum_p^{N-M-1}\prod_{k=1}^p\hat{A}_{t+N-k}\right)\hat{B}_{t+M} \end{bmatrix}. \quad (28)$$

При $M = N$:

$$\hat{P}_t = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ \hat{B}_{t+1} & 0 & 0 & \dots & 0 \\ \hat{A}_{t+2}\hat{B}_{t+1} & \hat{B}_{t+2} & 0 & \dots & 0 \\ \dots & \dots & \dots & \ddots & \dots \\ \left(\prod_{k=1}^{N-2}\hat{A}_{t+N-k}\right)\hat{B}_{t+1} & \left(\prod_{k=1}^{N-3}\hat{A}_{t+N-k}\right)\hat{B}_{t+2} & \dots & \hat{B}_{t+N-1} & 0 \end{bmatrix},$$

$$\hat{\Phi}_t = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ G\hat{B}_{t+1} & 0 & 0 & \dots & 0 \\ G\hat{A}_{t+2}\hat{B}_{t+1} & G\hat{B}_{t+2} & 0 & \dots & 0 \\ \dots & \dots & \dots & \ddots & \dots \\ G\left(\prod_{k=1}^{N-2}\hat{A}_{t+N-k}\right)\hat{B}_{t+1} & G\left(\prod_{k=1}^{N-3}\hat{A}_{t+N-k}\right)\hat{B}_{t+2} & \dots & G\hat{B}_{t+N-1} & 0 \end{bmatrix}. \quad (29)$$

Синтез управления осуществляется на основе минимизации критерия

$$J(\hat{x}_{t+1|t}, U_t) = \frac{1}{2} \sum_{k=1}^N \|\hat{y}_{t+k|t} - \bar{y}_{t+k}\|_C^2 + \frac{1}{2} \sum_{k=1}^M \|u_{t+k|t} - u_{t+k-1|t}\|_D^2,$$

который можно записать в следующем виде:

$$J(\hat{x}_{t+1|t}, U_t) = \frac{1}{2} U_t^T F_t U_t + U_t^T f_t + \alpha_t, \quad (30)$$

где α_t – слагаемое, не зависящее от управления,

$$F = \Phi^T \bar{C} \Phi + \bar{D}, \quad f = \Gamma \begin{bmatrix} \hat{x}_{t+1|t} \\ \hat{R}_t \\ \bar{Y}_t \end{bmatrix} - \begin{bmatrix} Du_t \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \Gamma = \begin{bmatrix} \Phi^T \bar{C} \Lambda & \Phi^T \bar{C} \hat{Q} & -\Phi^T \bar{C} \end{bmatrix}, \quad \bar{Y}_t = \begin{bmatrix} \bar{y}_{t+1|t} \\ \vdots \\ \bar{y}_{t+N|t} \end{bmatrix},$$

матрицы $\bar{C}$ и $\bar{D}$ имеют вид

$$\bar{C} = \begin{bmatrix} C & 0 & \vdots & 0 \\ 0 & C & \vdots & 0 \\ \dots & \dots & \ddots & \dots \\ 0 & 0 & \vdots & C \end{bmatrix}, \quad \bar{D} = \begin{bmatrix} 2D & -D & 0 & \vdots & 0 \\ -D & 2D & -D & \vdots & 0 \\ \dots & \ddots & \ddots & \ddots & \dots \\ 0 & \dots & -D & 2D & -D \\ 0 & \dots & 0 & -D & 2D \end{bmatrix}. \quad (31)$$

Окончательно адаптивное управление примет вид

$$u_{t+1|t}^* = (E_n \ 0 \ \dots \ 0)U_t^*,$$

где U_t^* определяется численно на основе оптимизации критерия (30) с учетом ограничений (4).

6. Применение алгоритма к задаче управления системой производства, хранения и поставок товара потребителям

Рассмотрим задачу управления производством, хранением и поставками товаров потребителям [4]. Модель объекта с дополнительно включенными неизвестными составляющими возмущений имеет вид

$$\begin{aligned} q_{t+1} &= \bar{A}(\theta)q_t + b_t + \varphi_t + \xi_t, \quad q_0 = \bar{q}_0, \\ z_{t+1} &= z_t + \bar{B}(\theta)\omega_t + d_t - \varphi_t + \zeta_t, \quad z_0 = \bar{z}_0, \end{aligned} \quad (32)$$

где $q_t \in R^s$, $q_{i,t}$ – количество товара i -го типа у потребителя в момент времени t ($t = \overline{1, T}$, $i = \overline{1, s}$); $z_{i,t}$ – количество товаров i -го типа на складе производителя; $\omega_{i,t}$ – объем производства товаров i -го типа; $\varphi_{i,t}$ – объем поставок товаров i -го типа; b_t , d_t – неизвестные составляющие возмущений; ξ_t , ζ_t – векторные гауссовские случайные последовательности ($M\{\xi_t\} = 0$, $M\{\zeta_t\} = 0$, $M\{\xi_t \xi_k^T\} = \Sigma \delta_{t,k}$, $M\{\zeta_t \zeta_k^T\} = \Xi \delta_{t,k}$, $M\{\xi_t \zeta_k^T\} = 0$); $\bar{A}(\theta)$ и $\bar{B}(\theta)$ – матрицы; θ_t – неизвестный постоянный вектор.

В каждый момент времени t должны выполняться ограничения

$$z_{\min} \leq z_t \leq z_{\max}, \quad 0 \leq \omega_t \leq \omega_{\max}, \quad 0 \leq \varphi_t \leq z_t.$$

Задача состоит в том, чтобы по наблюдениям определить стратегию управления производством, хранением и поставками товара, обеспечивающую количество товара у потребителя q_t , близкое к заданному вектору $\bar{q}$, при неполной информации о параметрах модели.

Моделирование проводится для следующих исходных данных:

$$\begin{aligned} \bar{A} &= \begin{bmatrix} \theta_1 & 0 \\ -0,25 & \theta_2 \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} \theta_3 & 0,1 \\ 0,2 & 0,8 \end{bmatrix}, \quad z_{\min} = \begin{bmatrix} 0,1 \\ 0,1 \end{bmatrix}, \quad z_{\max} = \begin{bmatrix} 1,5 \\ 2,5 \end{bmatrix}, \quad \omega_{\max} = \begin{bmatrix} 0,8 \\ 0,9 \end{bmatrix}, \quad z_0 = \begin{bmatrix} 0,2 \\ 0,2 \end{bmatrix}, \quad q_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\ V &= \text{diag}\{0,0005; 0,0005; 0,0005; 0,0005\}, \quad \bar{q} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad b = \begin{bmatrix} 0,04 \\ 0,02 \end{bmatrix}, \quad d = \begin{bmatrix} 0,05 \\ 0,03 \end{bmatrix}, \\ \omega_0 &= \varphi_0 = \begin{bmatrix} 0,1 \\ 0,1 \end{bmatrix}, \quad \theta = \begin{bmatrix} 0,75 \\ 0,9 \\ 0,3 \end{bmatrix}, \quad \theta_0 = \begin{bmatrix} 0,3 \\ 0,3 \\ 0,5 \end{bmatrix}, \quad C = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, \quad D = \text{diag}\{2; 2; 2; 2\}, \\ T &= 60, \quad M = N = 4, \quad P_{x_0} = E_4, \quad \bar{P}_{t_0} = E_4, \quad H = E_4, \quad W = 0, \quad T = 0, \quad \Theta_t = E_3, \quad T_\tau = 0, \quad R = E_3. \end{aligned} \quad (33)$$

Результаты численного моделирования алгоритма адаптивного управления, использующего оценки параметров θ и r , приведены в виде графиков переходных процессов на рис. 1–3.

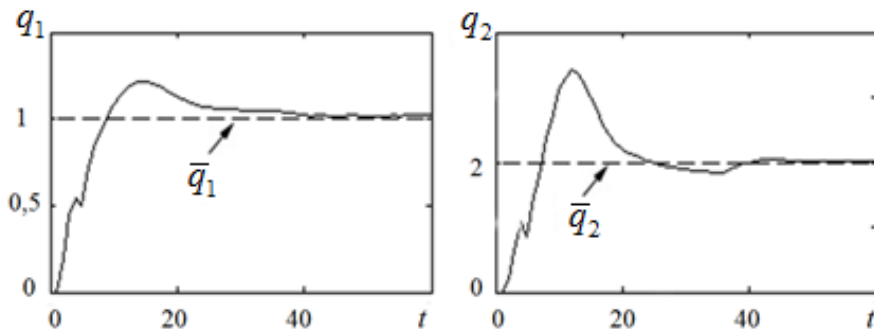


Рис. 1. Динамика изменения количества товаров у потребителя
(пунктирная линия – заданный уровень количества товара у потребителя,
сплошная – количество товара у потребителя)

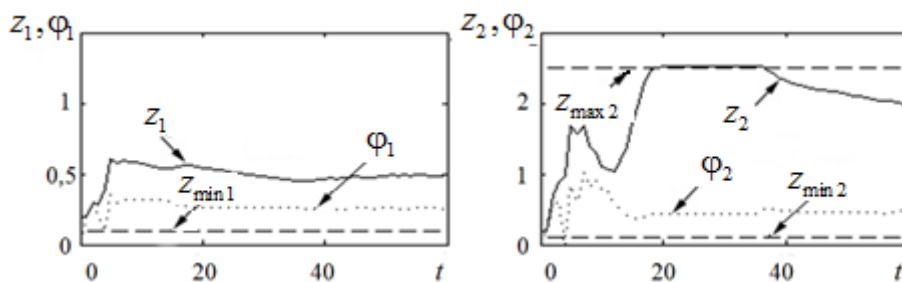


Рис. 2. Динамика изменения количества товаров на складе и объемов поставок (сплошная линия – количество товара на складе производителя, точки – объем поставок товаров, пунктирная линия – ограничения)

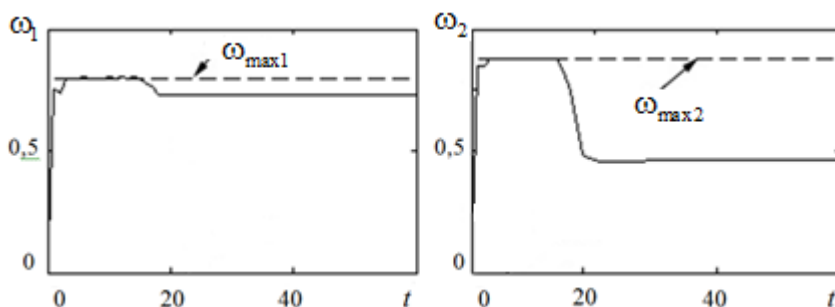


Рис. 3. Процессы изменения объемов производства товаров (сплошная линия – объем производства товара, пунктирная – ограничения)

Результаты моделирования показали, что дополнение алгоритмов адаптивного управления, использующих идентификацию неизвестных параметров, процедурами оценивания неизвестного входа позволяет реализовать цель управления экономическим объектом в условиях неполной информации о модели объекта.

Заключение

Разработаны алгоритмы синтеза адаптивного прогнозирующего управления дискретным объектом в условиях неполной информации о параметрах модели и при наличии в модели объекта неизвестного входа.

Дано применение алгоритма адаптивного прогнозирующего управления к экономической системе производства, хранения и поставок товаров потребителям.

ЛИТЕРАТУРА

1. Maciejowski J.M. Predictive control with constraints. Upper Saddle River (NJ) : Prentice Hall, 2002. 331 p.
2. Перепелкин Е.А. Прогнозирующее управление экономической системой производства, хранения и поставок товаров потребителям // Экономика и математические методы. 2004. Т. 40, № 1. С. 125–128.
3. Киселева М.Ю., Смагин В.И. Управление производством, хранением и поставками товаров на основе прогнозирующей модели выхода системы // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2009. № 2 (7). С. 24–30.
4. Приступа М.Ю., Смагин В.И. Прогнозирующее управление дискретными системами с неизвестным входом и его применение к задаче управления экономическим объектом // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2012. № 1 (18). С. 5–15.
5. Smagin V.I., Koshkin G.M., Kim K.S. Locally Optimal Inventory Control with Time Delay in Deliveries and Incomplete Information on Demand // Proceedings Second International Symposium on Stochastic Models in Reliability Engineering, Life Science and Operations Management. February 15–18, 2016. Beer Sheva, Israel, 2016. P. 570–574.
6. Домбровский В.В., Домбровский Д.В., Ляшенко Е.А. Управление с прогнозирующей моделью системами со случайными зависимыми параметрами при ограничениях и применение к оптимизации инвестиционного портфеля // Автоматика и телемеханика. 2006. № 12. С. 71–85.

7. Astrom K., Eykhoff P. System identification. A survey // Automatica. 1971. V. 7. P. 123–162.
8. Рубан А.И. Идентификация и чувствительность сложных систем. Томск : Изд-во Томск. ун-та, 1982. 302 с.
9. Janczak D., Grishin Yu. State estimation of linear dynamic system with unknown input and uncertain observation using dynamic programming // Control and Cybernetics. 2006. No. 4. P. 851–862.
10. Hsieh C-S. On the optimality of two-stage Kalman filtering for systems with unknown inputs // Asian Journal of Control. 2010. No. 4. P. 510–523.
11. Witczak M. Fault diagnosis and fault-tolerant control strategies for non-linear systems. Chapter 2. Unknown input observers and filters. Lecture Notes in Electrical Engineering. Springer International Publishing, Switzerland, 2014. P. 19–56.
12. Смагин В.И. Оценивание состояний нестационарных дискретных систем при неизвестном входе с использованием компенсаций // Известия вузов. Физика. 2015. Т. 58, № 7. С. 122–127.

Смагин Валерий Иванович, д-р техн. наук, профессор. E-mail: vsm@mail.tsu.ru
Национальный исследовательский Томский государственный университет

Поступила в редакцию 15 апреля 2017 г.

Smagin Valery I. (National Research Tomsk State University, Russian Federation).

Adaptive model predictive control discrete systems with unknown input.

Keywords: discrete systems; model predictive control; adaptive control; unknown input; production model.

DOI: 10.17223/19988605/40/3

The model of the object, observations and output is described by the following linear difference equations:

$$x_{t+1} = A(\theta_t)x_t + B(\theta_t)u_t + I r_t + w_t, \quad x_{t|t=0} = x_0,$$

$$\psi_t = Hx_t + v_t,$$

$$y_t = Gx_t,$$

where $x_t \in R^n$ is the state of the object, $u_t \in R^m$ is the control (known input), $\psi_t \in R^l$ is the observations vector, $r_t \in R^q$ is an unknown input, $y_t \in R^p$ is the controlled output, θ_t is an unknown parameter vector, $A(\theta_t)$, $B(\theta_t)$, I , H , G are matrices. The random perturbations w_t , the measurement noise v_t , and the vector of initial conditions x_0 are not correlated to each other and obey a Gaussian distribution with characteristics:

$$M\{w_t\} = 0, M\{v_t\} = 0, M\{w_t w_k^T\} = W \delta_{t,k}, M\{v_t v_k^T\} = V \delta_{t,k}, M\{w_t v_k^T\} = 0,$$

$$M\{x_0\} = \bar{x}_0, M\{(x_0 - \bar{x}_0)(x_0 - \bar{x}_0)^T\} = P_{x_0}.$$

Restrictions on the state and control vectors are representable in the form

$$a_1(t) \leq S_1 x_t \leq a_2(t), \quad \varphi_1(x_t, t) \leq S_2 u_t \leq \varphi_2(x_t, t),$$

where S_1 and S_2 are structure matrices of full rank consisting of zeros and ones defining the components of the vectors x_t and u_t , on which constraints are imposed; $a_1(t)$, $a_2(t)$, $\varphi_1(x_t, t)$, $\varphi_2(x_t, t)$ are the given vectors and vector-functions.

Based on the minimization of the criterion

$$J(\hat{x}_{t+1|t}, U_t) = \frac{1}{2} \sum_{k=1}^N \|\hat{y}_{t+k|t} - \bar{y}_{t+k}\|_C^2 + \frac{1}{2} \sum_{k=1}^M \|u_{t+k|t} - u_{t+k-1|t}\|_D^2$$

algorithms for the synthesis of adaptive predictive control of a discrete object under conditions of incomplete information about the parameters of the model and in the presence of an unknown input in the model of the object are considered.

The application of the algorithm of adaptive predictive control to the economic system of production, storage and delivery of goods to consumers is given.

REFERENCES

1. Maciejowski, J.M. (2002) *Predictive control with constraints*. Prentice Hall.
2. Perepiolkin, E.A. (2004) Forecast-making control of an economic system of production, storage and deliveries of goods to consumers. *Ekonomika i matematicheskie metody – Economics and Mathematical Methods*. 40(1). pp. 125–128. (In Russian).
3. Kiseleva, M.Yu. & Smagin, V.I. (2009) Control of goods production, storage and delivery based on prediction model systems output. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science*. 2(7). pp. 24–30. (In Russian).
4. Pristupa, M.Yu. & Smagin, V.I. (2012) Model predictive control discrete systems with unknown input and its application to control problem of economic object. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science*. 1(18). pp. 5–15. (In Russian).
5. Smagin, V.I., Koshkin, G.M. & Kim, K.S. (2016) Locally optimal inventory control with time delay in deliveries and incomplete information on demand. *Proceedings Second International Symposium on Stochastic Models in Reliability Engineering, Life Science and Operations Management*. Beer Sheva, Israel. February 15–18, 2016. pp. 570–574.

6. Dombrovsky, V.V., Dombrovsky, D.V. & Lyashenko, E.A. (2006) Model predictive control of systems with random dependent parameters under constraints and its application to the investment portfolio optimization. *Automation and Remote Control*. 67(12). pp. 1927–1939. DOI: 10.1134/S000511790612006X
7. Astrom, K. & Eykhoff, P. (1971) System identification. A survey. *Automatica*. 7. pp.123-162. DOI: 10.1016/0005-1098(71)90059-8
8. Ruban, A.I. (1982) *Identifikatsiya i chuvstvitel'nost' slozhnykh sistem* [Identification and sensitivity of complex systems]. Tomsk: Tomsk State University.
9. Janczak, D. & Grishin, Yu. (2006) State estimation of linear dynamic system with unknown input and uncertain observation using dynamic programming. *Control and Cybernetics*. 4. pp. 851–862.
10. Hsieh, C-S. (2010) On the optimality of two-stage Kalman filtering for systems with unknown inputs. *Asian Journal of Control*. 4. pp. 510–523. DOI: 10.1002/asjc.205
11. Witczak, M. (2014) *Fault diagnosis and fault-tolerant control strategies for non-linear systems. Chapter 2. Unknown input observers and filters*. Switzerland: Springer International Publishing. pp. 19–56.
12. Smagin, V.I. (2015) State estimation for nonstationary discrete systems with unknown input using compensations. *Russian Physics Journal*. 58(7). pp. 1010–1017. DOI: 10.1007/s11182-015-0602-x

ОБРАБОТКА ИНФОРМАЦИИ

УДК 519.21

DOI: 10.17223/19988605/40/4

А.М. Горцев, М.Е. Завгородняя

ОЦЕНИВАНИЕ ПАРАМЕТРА НЕПРОДЛЕВАЮЩЕГОСЯ МЕРТВОГО ВРЕМЕНИ
СЛУЧАЙНОЙ ДЛИТЕЛЬНОСТИ В ПУАССОНОВСКОМ ПОТОКЕ СОБЫТИЙ

Рассматривается простейший поток событий с параметром λ , функционирующий в условиях непродлевающегося мертвого времени, порождаемого текущим наблюдаемым событием. Случайное мертвое время распределено по экспоненциальному закону с параметром α . Показано, что оценивание неизвестных параметров λ и α ($\lambda \neq \alpha$) методом максимального правдоподобия либо методом моментов неосуществимо. Оценивание является возможным указанными методами только тогда, когда один из параметров является известным либо известно, что $\lambda = \alpha$. При этом метод максимального правдоподобия обеспечивает более приемлемые оценки, чем метод моментов.

Ключевые слова: простейший поток; непродлевающееся мертвое время; оценки параметров; метод максимального правдоподобия; метод моментов.

Распространенными математическими моделями физических явлений и процессов являются потоки событий. В частности, такие модели применяются при исследовании информационных потоков сообщений в телекоммуникационных системах, в спутниковых сетях связи и т.п. Задачи по оценке параметров случайных потоков событий возникают в оптических и лазерных системах, функционирующих в режиме счета фотонов. В большинстве публикаций авторы рассматривают математические модели потоков событий, когда события потока доступны наблюдению. Однако на практике возникают ситуации, когда наступившее событие влечет за собой ненаблюдаемость последующих событий. Причиной ненаблюдаемости выступает мертвое время регистрирующих приборов [1, 2], в течение которого зарегистрированное событие обрабатывается; другие же события, поступившие в этот период, теряются. Регистрирующие приборы при этом делятся на два вида: с непродлевающимся мертвым временем и продлевающимся [Там же]. Задачи по оценке параметров и состояний потока событий в условиях наличия мертвого времени фиксированной длительности рассматривались в работах [3–33]. При этом в [3, 5–8, 10, 12–16, 18–31] получены результаты для непродлевающегося мертвого времени, в [4, 9, 11, 17] – для продлевающегося.

При непродлеваемом мертвом времени фиксированной длительности T решены задачи по нахождению оценки $\hat{T}$: в [3] – для пуассоновского потока, в [5, 6] – для синхронного альтернирующего потока, в [7, 8, 10] – для асинхронного альтернирующего потока, в [12] – для синхронного потока, в [13–15] – для асинхронного потока, в [16, 18] – для полусинхронного потока, в [19–21] – для обобщенного асинхронного потока, в [22, 23] – для обобщенного полусинхронного потока, в [24, 25] – для МАР-потока, в [26, 27] – для модулированного синхронного потока, в [28, 29] – для модулированного обобщенного полусинхронного потока, в [30–33] – для модулированного МАР-потока; при продлеваемом мертвом времени фиксированной длительности T – в [4] (для пуассоновского потока), в [9] – для альтернирующего асинхронного потока, в [11] – для синхронного потока, в [17] – для полусинхронного потока.

Однако достаточно открытым остается вопрос изучения потоков событий, когда мертвое время является случайной величиной с тем или иным законом распределения. В настоящей статье в текущем времени строятся оценки параметров наблюдаемого потока, порожденного пуассоновским потоком, функционирующим в условиях непродлевающегося мертвого времени случайной длительности.

1. Математическая модель наблюдаемого потока

Рассматривается пуассоновский поток событий интенсивности λ . После каждого зарегистрированного в момент времени t_k события наступает период мертвого времени случайной длительности, в течение которого другие события потока недоступны наблюдению и не вызывают его продления (непродлевающееся мертвое время). По окончании периода мертвого времени первое наступившее событие снова создает период мертвого времени случайной длительности и т.д. Принимается, что длительность мертвого времени распределена по экспоненциальному закону с параметром α . Вариант возникновения ситуации приведен на рис. 1, где прямоугольниками обозначены периоды мертвого времени длительности $T_1, T_2, \dots$; $t_1, t_2, \dots$ – моменты наступления событий в наблюдаемом потоке.

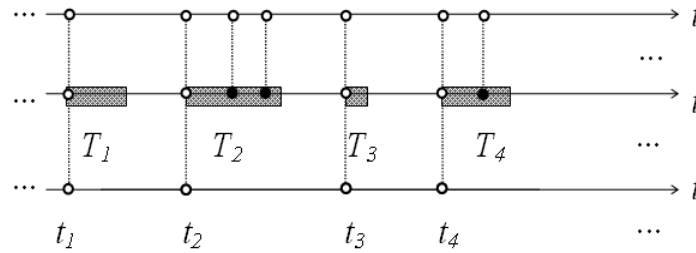


Рис. 1. Формирование наблюдаемого потока событий

Рассматривается установившийся (стационарный) режим функционирования наблюдаемого потока событий, поэтому переходными процессами на полуинтервале наблюдения $(t_0, t]$, где t_0 – начало наблюдений, t – окончание наблюдений, пренебрегаем. Тогда без потери общности можно положить $t_0 = 0$.

Предполагается, что параметры наблюдаемого потока λ и α являются неизвестными. Необходимо в момент окончания наблюдений (в момент времени t) на основании выборки $t_1, t_2, \dots, t_n$ наблюдаемых моментов наступления событий потока осуществить методом максимального правдоподобия и методом моментов построение оценок $\hat{\lambda}$ и $\hat{\alpha}$ и произвести сравнение качества полученных оценок.

2. Оценки максимального правдоподобия параметров λ и α

Обозначим через $\tau_k = t_{k+1} - t_k$, $k = 1, 2, \dots$, значение длительности k -го интервала между соседними событиями наблюдаемого потока ($\tau_k \geq 0$). Так как рассматривается стационарный режим, то плотность вероятности значений длительности k -го интервала есть $p_\alpha(\tau_k) = p_\alpha(\tau)$, $\tau \geq 0$, для любого k (индекс α подчеркивает, что плотность вероятности зависит от случайной длительности мертвого времени) в наблюдаемом потоке событий. В силу этого момент времени t_k без потери общности можно положить равным нулю, т.е. момент наступления события есть $\tau = 0$. В [3] получено выражение для плотности вероятности $p(\tau | T)$, когда длительность мертвого времени является детерминированной величиной:

$$p(\tau | T) = 0, \tau < T, \quad p(\tau | T) = \lambda e^{-\lambda(\tau - T)}, \tau \geq T. \quad (1)$$

Тогда плотность вероятности $p_\alpha(\tau)$ запишется в виде

$$p_\alpha(\tau) = \int_0^\tau p(T) p(\tau | T) dT. \quad (2)$$

Подставляя в (2) выражение (1) и учитывая, что $p(T) = \alpha e^{-\alpha T}$, находим

$$p_\alpha(\tau) = \frac{\alpha \lambda}{\lambda - \alpha} (e^{-\alpha \tau} - e^{-\lambda \tau}), \tau \geq 0. \quad (3)$$

Пусть $\tau_1 = t_2 - t_1, \dots, \tau_n = t_{n+1} - t_n$, $\tau_1 \geq 0, \dots, \tau_n \geq 0$ – последовательность измеренных в результате наблюдения за потоком в течение полуинтервала наблюдения $(0, t]$ значений длительностей интервалов между соседними событиями потока. В силу предпосылок последовательность $\{\tau_k\}$, $k = \overline{1, n}$, образует

конкретную реализацию выборки из совокупности независимых случайных величин. Тогда функция правдоподобия примет вид

$$L(\alpha, \lambda | \tau_1, \dots, \tau_n) = \left(\frac{\lambda \alpha}{\lambda - \alpha} \right)^n \prod_{k=1}^n (e^{-\alpha \tau_k} - e^{-\lambda \tau_k}). \quad (4)$$

Здесь возможны три варианта: $\lambda > \alpha$, $\lambda = \alpha$, $\lambda < \alpha$ ($\lambda > 0$, $\alpha > 0$).

Рассмотрим вариант $\lambda > \alpha$. Тогда, логарифмируя (4), после чего взяв первые производные по λ и α , получим систему уравнений правдоподобия для определения оценок $\hat{\lambda}$, $\hat{\alpha}$

$$-\frac{n\alpha}{\lambda(\lambda - \alpha)} + \sum_{k=1}^n \tau_k \frac{e^{\alpha \tau_k}}{e^{\lambda \tau_k} - e^{\alpha \tau_k}} = 0, \quad (5)$$

$$\frac{n\lambda}{\alpha(\lambda - \alpha)} - \sum_{k=1}^n \tau_k \frac{e^{\lambda \tau_k}}{e^{\lambda \tau_k} - e^{\alpha \tau_k}} = 0. \quad (6)$$

Преобразовывая (5), (6), находим

$$\lambda + \alpha = \hat{\tau} \alpha \lambda, \quad (7)$$

$$\frac{n}{\alpha[(2/\hat{\tau}) - \alpha]\hat{\tau}} + \sum_{k=1}^n \tau_k \left[\exp \left\{ \alpha \tau_k \frac{\alpha - (2/\hat{\tau})}{\alpha - (1/\hat{\tau})} \right\} - 1 \right]^{-1} = 0. \quad (8)$$

В (7), (8) величина $\hat{\tau} = (1/n) \sum_{k=1}^n \tau_k$. Отметим, что уравнение (8) есть уравнение относительно α , решение которого определяет оценку $\hat{\alpha}$. Подстановкой $\hat{\alpha}$ в уравнение (7) находится оценка $\hat{\lambda}$. Если решение уравнений (7), (8) существует, то $\hat{\alpha} < (2/\hat{\tau})$, $\hat{\lambda} > (2/\hat{\tau})$.

Эквивалентные преобразования уравнения (5), (6) приводят к уравнениям

$$\lambda + \alpha = \hat{\tau} \alpha \lambda, \quad (9)$$

$$\frac{n}{\lambda[(2/\hat{\tau}) - \lambda]\hat{\tau}} + \sum_{k=1}^n \tau_k \left[\exp \left\{ \lambda \tau_k \frac{\lambda - (2/\hat{\tau})}{\lambda - (1/\hat{\tau})} \right\} - 1 \right]^{-1} = 0. \quad (10)$$

Уравнение (10) есть уравнение относительно λ , решение которого определяет оценку $\hat{\lambda}$. Подстановкой $\hat{\lambda}$ в уравнение (9) находится оценка $\hat{\alpha}$. Если решение уравнений (9), (10) существует, то $\hat{\alpha} < (2/\hat{\tau})$, $\hat{\lambda} > (2/\hat{\tau})$.

Замечание 1. Уравнения (7), (8) и уравнения (9), (10) полностью идентичны.

Замечание 2. Аналогичный результат получается для варианта $\lambda < \alpha$, за исключением того что $\hat{\alpha} > (2/\hat{\tau})$, $\hat{\lambda} < (2/\hat{\tau})$.

Рассмотрим вариант $\alpha = \lambda$. Тогда (4) выпишется в виде

$$L(\alpha = \lambda | \tau_1, \dots, \tau_n) = \lambda^{2n} \exp \left\{ -\lambda \sum_{k=1}^n \tau_k \right\} \prod_{k=1}^n \tau_k.$$

При этом из уравнения правдоподобия находится оценка $\hat{\lambda} = 2/\hat{\tau}$.

Теорема 1. Метод максимального правдоподобия вне зависимости от соотношения неизвестных параметров λ и α обеспечивает оценки $\hat{\lambda}$ и $\hat{\alpha}$ в виде $\hat{\alpha} = \hat{\lambda} = 2/\hat{\tau}$, т.е. его работоспособность распространяется только на вариант $\alpha = \lambda$.

Доказательство. Рассмотрим вариант $\lambda > \alpha$. В силу идентичности уравнений (8), (10) выпишем их в виде

$$\frac{n}{x[(2/\hat{\tau}) - x]\hat{\tau}} + \sum_{k=1}^n \tau_k \left[\exp \left\{ x \tau_k \frac{x - (2/\hat{\tau})}{x - (1/\hat{\tau})} \right\} - 1 \right]^{-1} = 0. \quad (11)$$

Пусть x^* – корень уравнения (11). Тогда $\alpha^* = x^*$ – корень уравнения (8). С другой стороны, $\lambda^* = x^*$ – корень уравнения (10). Таким образом, имеем $\alpha^* = \lambda^*$. Тогда из (7) следует $\hat{\lambda} = \alpha^* / (\hat{\tau}\alpha^* - 1)$, из (9) следует $\hat{\alpha} = \lambda^* / (\hat{\tau}\lambda^* - 1)$. Так как $\alpha^* = \lambda^*$, то $\hat{\lambda} = \hat{\alpha}$. Из уравнений (7), (8) вытекает $\hat{\alpha} < (2/\hat{\tau})$, $\hat{\lambda} > (2/\hat{\tau})$. Тогда, так как $\hat{\alpha} = \hat{\lambda}$, имеем $\hat{\lambda} < (2/\hat{\tau})$, $\hat{\lambda} > (2/\hat{\tau})$, поэтому $\hat{\alpha} = \hat{\lambda} = (2/\hat{\tau})$ – единственная точка, в которой уравнения (7), (8), а также уравнения (9), (10) не противоречивы.

Аналогично рассматривается вариант $\lambda < \alpha$. Теорема 1 доказана.

3. ММ-оценки параметров λ и α

Получим оценки параметров λ и α методом моментов. Теоретическое среднее значение длительности интервала между соседними моментами наступления событий наблюдаемого потока (начальный момент первого порядка):

$$T_1 = \int_0^{\infty} \tau p_{\alpha}(T) d\tau = \frac{1}{\alpha} + \frac{1}{\lambda}.$$

Оценка $\hat{T}_1 = \frac{1}{n} \sum_{k=1}^n \tau_k = \hat{\tau}_1$ ($\hat{\tau}_1 = \hat{\tau}$).

Начальный момент второго порядка

$$T_2 = \int_0^{\infty} \tau^2 p_{\alpha}(T) d\tau = 2 \left(\frac{1}{\alpha^2} + \frac{1}{\alpha\lambda} + \frac{1}{\lambda^2} \right).$$

Оценка $\hat{T}_2 = \frac{1}{n} \sum_{k=1}^n \tau_k^2 = \hat{\tau}_2$. Уравнения моментов при этом примут вид

$$\lambda + \alpha = \hat{\tau}_1 \alpha \lambda, \quad 2 \left(\frac{1}{\alpha^2} + \frac{1}{\alpha\lambda} + \frac{1}{\lambda^2} \right) = \hat{\tau}_2 \alpha^2 \lambda^2. \quad (12)$$

Преобразовывая (12), находим

$$\lambda = \alpha / [\hat{\tau}_1 \alpha - 1], \quad [\hat{\tau}_1^2 - (1/2)\hat{\tau}_2] \alpha^2 - \hat{\tau}_1 \alpha + 1 = 0. \quad (13)$$

Решением квадратного уравнения (13) являются корни α_1 и α_2 , которые определяют оценки $\hat{\alpha}_1 = \alpha_1$, $\hat{\alpha}_2 = \alpha_2$. В свою очередь, оценки $\hat{\alpha}_1$ и $\hat{\alpha}_2$ путем подстановки в первое уравнение (13) определяют соответствующие оценки $\hat{\lambda}_1$ и $\hat{\lambda}_2$.

Эквивалентные преобразования уравнения (12) приводят к уравнениям

$$\alpha = \lambda / [\hat{\tau}_1 \lambda - 1], \quad [\hat{\tau}_1^2 - (1/2)\hat{\tau}_2] \lambda^2 - \hat{\tau}_1 \lambda + 1 = 0. \quad (14)$$

Решением квадратного уравнения (14) являются корни λ_1 и λ_2 , которые определяют оценки $\hat{\lambda}_1 = \lambda_1$, $\hat{\lambda}_2 = \lambda_2$. В свою очередь, оценки $\hat{\lambda}_1$ и $\hat{\lambda}_2$ путем подстановки в первое уравнение (14) определяют соответствующие оценки $\hat{\alpha}_1$ и $\hat{\alpha}_2$.

Замечание 3. Уравнения (13) и (14) полностью идентичны.

Теорема 2. Решение уравнений моментов (12) не существует.

Доказательство. В силу идентичности уравнений (13), (14), полученных из (12), выпишем квадратное уравнение в виде

$$\psi(x) = [\hat{\tau}_1^2 - (1/2)\hat{\tau}_2] x^2 - \hat{\tau}_1 x + 1 = 0. \quad (15)$$

Корнями уравнения (15) являются

$$x_1 = \frac{\hat{\tau}_1 - \sqrt{2\hat{\tau}_2 - 3\hat{\tau}_1^2}}{2\hat{\tau}_1^2 - \hat{\tau}_2}, \quad x_2 = \frac{\hat{\tau}_1 + \sqrt{2\hat{\tau}_2 - 3\hat{\tau}_1^2}}{2\hat{\tau}_1^2 - \hat{\tau}_2}.$$

При этом должно выполняться $x_1 > 0$, $x_2 > 0$, так как $\alpha > 0$, $\lambda > 0$.

Если дискриминант $D = 2\hat{\tau}_2 - 3\hat{\tau}_1^2 < 0$, то решение уравнения (15) не существует. Случай $D = 0$, который приводит к оценкам $\hat{\alpha} = \hat{\lambda} = (2/\hat{\tau})$, имеет нулевую вероятность. Таким образом, если $D \leq 0$, то решение уравнений (12) не существует. Рассмотрим случай $D > 0$: 1) пусть $d = \hat{\tau}_1^2 - (1/2)\hat{\tau}_2 < 0$, тогда решение уравнений (12) не существует; 2) случай $d = 0$ имеет нулевую вероятность. Таким образом, если $D > 0$ и $d \leq 0$, то решение уравнений (12) не существует.

Рассмотрим последнюю ситуацию: $D > 0$ и $d > 0$. Так как x_1 – корень уравнения (13), то $\alpha_1 = x_1$. С другой стороны, x_1 – корень уравнения (14), тогда $\lambda_1 = x_1$, при этом $(1/\hat{\tau}_1) < x_1 < (2/\hat{\tau}_1)$. Отсюда следует $\alpha_1 = \lambda_1$. Из первого равенства системы (13) находим $\hat{\lambda}_1 = \alpha_1/[\hat{\tau}_1\alpha_1 - 1]$, из первого равенства системы (14) получаем $\hat{\alpha}_1 = \lambda_1/[\hat{\tau}_1\lambda_1 - 1]$. Так как $\alpha_1 = \lambda_1$, то $\hat{\alpha}_1 = \hat{\lambda}_1$. Из уравнений (13) следует $(1/\hat{\tau}_1) < \hat{\alpha}_1 < (2/\hat{\tau}_1)$, $\hat{\lambda}_1 > (2/\hat{\tau}_1)$. Из уравнений (14) вытекает $(1/\hat{\tau}_1) < \hat{\lambda}_1 < (2/\hat{\tau}_1)$, $\hat{\alpha}_1 > (2/\hat{\tau}_1)$. Так как уравнения (13), (14) эквивалентны, то единственная точка, которая одновременно удовлетворяет этим уравнениям, есть $\hat{\alpha}_1 = \hat{\lambda}_1 = (2/\hat{\tau}_1)$. Последнее равенство возможно, если $D = 0$. Выполнение последнего равенства имеет нулевую вероятность, так что решения уравнений (12) не существует.

Аналогичное утверждение имеет место для корня x_2 . Теорема 2 доказана.

Таким образом, при оценке неизвестных параметров λ и α метод моментов неработоспособен.

Замечание 4. Для варианта $\alpha = \lambda$ имеем $T_1 = (2/\alpha)$. Тогда уравнение моментов принимает вид $(2/\alpha) = \hat{\tau}_1$, откуда следует $\hat{\alpha} = (2/\hat{\tau}_1)$.

4. МП- и ММ-оценки параметра α

Пусть параметр λ является известным.

МП-оценка параметра α . Для оценки параметра α имеем уравнение (6). Обозначим левую часть уравнения (6) через $f(\alpha)$. После преобразования уравнение (6) примет вид

$$f(\alpha) = \sum_{k=1}^n \left\{ \frac{\lambda}{\alpha(\lambda - \alpha)} - \tau_k \frac{e^{\lambda\tau_k}}{e^{\lambda\tau_k} - e^{\alpha\tau_k}} \right\} = 0, \quad \alpha \geq 0,$$

$$\lim_{\alpha \rightarrow 0} f(\alpha) = \infty, \quad \lim_{\alpha \rightarrow \lambda} f(\alpha) = \frac{n}{2\lambda} (2 - \lambda\hat{\tau}_1), \quad \lim_{\alpha \rightarrow \infty} f(\alpha) = -0. \quad (16)$$

Замечание 5. Функция $f(\alpha)$, по крайней мере, один раз пересекает ось абсцисс, т.е. решение уравнения (16) существует.

Замечание 6. Решение уравнения (16) возможно только численно.

ММ-оценка параметра α . Для оценки параметра α имеем первое уравнение моментов (12), из которого получаем $\hat{\alpha} = \lambda/[\hat{\tau}_1\lambda - 1]$.

С целью установления качества получаемых оценок параметра α методом максимального правдоподобия и методом моментов поставлен статистический эксперимент.

Отдельный j -й эксперимент ($j = \overline{1, N}$) заключается в следующем: 1) при заданных значениях α , λ и заданном времени моделирования T_m осуществляется имитационное моделирование наблюдаемого потока; выходом имитационной модели является последовательность значений $\tau_1, \tau_2, \dots, \tau_n$; 2) численно решается уравнение (16), т.е. находится оценка $\hat{\alpha}_{МП}^{(j)}$; 3) по первой формуле (14) находится оценка $\hat{\alpha}_{ММ}^{(j)}$; 4) осуществляется повторение N раз шагов 1–3 для получения выборок достаточного объема.

Результатом выполнения описанного алгоритма являются две выборки $(\hat{\alpha}_{МП}^{(1)}, \dots, \hat{\alpha}_{МП}^{(N)})$, $(\hat{\alpha}_{ММ}^{(1)}, \dots, \hat{\alpha}_{ММ}^{(N)})$, на основании которых вычисляются выборочные средние

$$\hat{\bar{\alpha}}_{МП} = \frac{1}{N} \sum_{j=1}^N \hat{\alpha}_{МП}^{(j)}, \quad \hat{\bar{\alpha}}_{ММ} = \frac{1}{N} \sum_{j=1}^N \hat{\alpha}_{ММ}^{(j)}$$

и выборочные вариации получаемых оценок

$$\hat{V}_{МП} = \frac{1}{N} \sum_{j=1}^N (\hat{\alpha}_{МП}^{(j)} - \alpha)^2, \quad \hat{V}_{ММ} = \frac{1}{N} \sum_{j=1}^N (\hat{\alpha}_{ММ}^{(j)} - \alpha)^2,$$

где α – известное истинное значение оцениваемого параметра (это значение задается в исходных данных имитационной модели наблюдаемого потока).

Программа расчета реализована на языке программирования C# в среде Microsoft Visual Studio C#. При проведении статистического эксперимента выбраны следующие параметры имитационной ли: $\lambda = 1$, $\alpha = 2$. В табл. 1, 2 в качестве иллюстрации приведены результаты эксперимента. Результаты получены для $N = 100$.

Т а б л и ц а 1

Результаты статистического эксперимента $T_m \leq 60$

T_m	20	30	40	50	60
$\hat{\alpha}_{МП}$	2,628	2,701	2,349	2,295	2,278
$\hat{V}_{МП}$	3,274	3,561	1,474	1,016	0,734
$\hat{\alpha}_{ММ}$	63,135	3,806	2,685	2,501	2,352
$\hat{V}_{ММ}$	359361,2	19,696	24,892	3,184	1,069
$\hat{V}_{МП} - \hat{V}_{ММ}$	-359357,9	-16,135	-23,418	-2,168	-0,335

Т а б л и ц а 2

Результаты статистического эксперимента $100 \leq T_m \leq 500$

T_m	100	200	300	400	500
$\hat{\alpha}_{МП}$	2,184	2,056	2,051	1,973	2,010
$\hat{V}_{МП}$	0,350	0,137	0,063	0,061	0,040
$\hat{\alpha}_{ММ}$	2,211	2,104	2,080	1,971	2,023
$\hat{V}_{ММ}$	0,446	0,255	0,096	0,076	0,050
$\hat{V}_{МП} - \hat{V}_{ММ}$	-0,096	-0,118	-0,033	-0,015	-0,010

В первой строке таблиц указано время моделирования T_m (в табл. 1: $T_m = 20, 30, \dots, 60$; в табл. 2: $T_m = 100, 200, \dots, 500$); во второй и четвертой строках приведены значения выборочных средних $\hat{\alpha}_{МП}$ и $\hat{\alpha}_{ММ}$; в третьей и пятой – значения выборочных вариаций $\hat{V}_{МП}$ и $\hat{V}_{ММ}$; в шестой – значения разностей $\hat{V}_{МП} - \hat{V}_{ММ}$.

Анализ численных результатов показывает, что в смысле введенного критерия (выборочная вариация оценки $\hat{\alpha}$) оценки $\hat{\alpha}_{МП}$ всегда лучше оценок $\hat{\alpha}_{ММ}$. При этом качество оценок улучшается при увеличении времени моделирования T_m , что является естественным.

Заключение

По результатам проведенного исследования можно сделать следующие выводы: 1) оценивание неизвестных параметров наблюдаемого потока α , λ по наблюдениям моментов наступления событий методом максимального правдоподобия либо методом моментов является невозможным; 2) в случае, когда известно, что $\lambda = \alpha$, либо известен параметр λ (параметр α), осуществимо оценивание параметра α (параметра λ), при этом метод максимального правдоподобия обеспечивает более приемлемые оценки, чем метод моментов.

ЛИТЕРАТУРА

1. Курочкин С.С. Многомерные статистические анализаторы. М. : Атомиздат, 1968. 446 с.
2. Апанасович В.В., Коляда А.А., Чернявский А.Ф. Статистический анализ случайных потоков в физическом эксперименте. Минск : Университетское, 1988. 256 с.

3. Горцев А.М., Климов И.С. Оценка интенсивности пуассоновского потока событий в условиях частичной его ненаблюдаемости // Радиотехника. 1991. № 12. С. 3–7.
4. Горцев А.М., Климов И.С. Оценивание периода ненаблюдаемости и интенсивности пуассоновского потока событий // Радиотехника. 1996. № 2. С. 8–11.
5. Горцев А.М., Нежелская Л.А. Оценивание длительности мертвого времени и параметров синхронного альтернирующего потока событий // Вестник Томского государственного университета. Приложение. 2003. № 6. С. 232–239.
6. Василевская Т.П., Горцев А.М., Нежелская Л.А. Оценивание длительности мертвого времени и параметров синхронного альтернирующего потока с проявлением либо не проявлением событий // Вестник Томского государственного университета. 2004. Приложение. № 9 (II). С. 129–138.
7. Горцев А.М., Ниссенбаум О.В. Оценивание длительности мертвого времени и параметров асинхронного альтернирующего потока событий при непродлеваемом мертвом времени // Известия высших учебных заведений. Физика. 2005. № 10. С. 35–49.
8. Горцев А.М., Ниссенбаум О.В. Оценивание длительности мертвого времени и параметров асинхронного альтернирующего потока событий с иницированием лишнего события // Вестник Томского государственного университета. 2004. № 284. С. 137–145.
9. Горцев А.М., Паршина М.Е. Оценивание параметров альтернирующего потока событий в условиях «мертвого времени» // Известия высших учебных заведений. Физика. 1999. № 4. С. 8–13.
10. Горцев А.М., Завгородняя М.Е. Оценка параметров альтернирующего потока событий при условии его частичной наблюдаемости // Оптика атмосферы и океана. 1997. Т. 10, № 3. С. 273–280.
11. Bushlanov I.V., Gortsev A.M., Nezhelskaya L.A. Estimating parameters of the synchronous twofold-stochastic flow of events // Automation and Remote Control. 2008. V. 69, No. 9. P. 1517–1533.
12. Горцев А.М., Нежелская Л.А. Оценивание длительности «мертвого времени» и интенсивностей синхронного дважды стохастического потока событий // Радиотехника. 2004. № 10. С. 8–16.
13. Vasileva L.A., Gortsev A.M. Estimation of the dead time of an asynchronous double stochastic flow of events under incomplete observability // Automation and Remote Control. 2003. V. 64, No.12. P. 1890–1898.
14. Vasileva L.A., Gortsev A.M. Estimation of parameters of a double-stochastic flow of events under conditions of its incomplete observability // Automation and Remote Control. 2002. V. 63, No. 3. P. 511–515.
15. Горцев А.М., Куснатдинов Р.Т. Оценивание состояний МС-потока событий при его частичной наблюдаемости // Известия высших учебных заведений. Физика. 1998. № 4. С. 22–30.
16. Gortsev A.M., Nezhelskaya L.A. Estimation of the dead-time period and parameters of a semi-synchronous double-stochastic stream of events // Measurement Techniques. 2003. V. 46, No. 6. P. 536–545.
17. Горцев А.М., Нежелская Л.А. Полусинхронный дважды стохастический поток событий при продлеваемом мертвом времени // Вычислительные технологии. 2008. Т. 13, № 1. С. 31–41.
18. Нежелская Л.А. Оптимальное оценивание состояний полусинхронного потока событий в условиях его частичной наблюдаемости // Вестник Томского государственного университета. 2000. № 269. С. 95–98.
19. Горцев А.М., Леонова М.А., Нежелская Л.А. Совместная плотность вероятностей длительности интервалов обобщенного асинхронного потока событий при непродлеваемом мертвом времени // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2012. № 4 (21). С. 14–25.
20. Леонова М.А., Нежелская Л.А. Оценка максимального правдоподобия длительности мертвого времени в обобщенном асинхронном потоке событий // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2013. № 2 (23). С. 54–63.
21. Горцев А.М., Леонова М.А., Нежелская Л.А. Сравнение МП- и ММ-оценок длительности мертвого времени в обобщенном асинхронном потоке событий // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2013. № 4 (25). С. 32–42.
22. Горцев А.М., Калягин А.А., Нежелская Л.А. Оценка максимального правдоподобия длительности мертвого времени в обобщенном полусинхронном потоке // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2015. № 1 (30). С. 27–37.
23. Горцев А.М., Калягин А.А., Нежелская Л.А. Совместная плотность вероятностей длительности интервалов обобщенного полусинхронного потока событий при непродлеваемом мертвом времени // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2014. № 2 (27). С. 19–29.
24. Gortsev A.M., Solov'ev A.A. Joint probability density of interarrival interval of a flow of physical events with unextendable dead time period // Russian Physics Journal. 2014. V. 57, No. 7. P. 973–983.
25. Горцев А.М., Соловьев А.А. Оценка максимального правдоподобия длительности непродлеваемого мертвого времени в потоке физических событий // Известия высших учебных заведений. Физика. 2015. Т. 58, № 11. С. 141–149.
26. Сиротина М.Н., Горцев А.М. Оценка максимального правдоподобия длительности мертвого времени в модулированном синхронном дважды стохастическом потоке событий // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2016. № 1 (34). С. 50–64.
27. Gortsev A., Sirotina A. Joint probability density function of modulated synchronous flow interval duration under conditions of fixed dead time // Communications in Computer and Information Science. 2015. V. 564. P. 41–52.
28. Бахолдина М.А., Горцев А.М. Оптимальная оценка состояний модулированного обобщенного полусинхронного потока событий при непродлеваемом мертвом времени // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2014. № 1 (26). С. 13–24.
29. Bakholdina M., Gortsev A. Joint probability density of the intervals length of modulated semi-synchronous integrated flow of events in conditions of a constant dead time and the flow recurrence conditions // Communications in Computer and Information Science. 2015. V. 564. P. 13–27.

30. Nezhelskaya L. Optimal state estimation in modulated MAP event flows with unextendable dead time // Communications in Computer and Information Science. 2014. V. 487. P. 342–350.
31. Нежелская Л.А. Условия рекуррентности потока физических событий при непродлеваемом мертвом времени // Известия высших учебных заведений. Физика. 2015. Т. 58, № 12. С. 168–175.
32. Nezhel'skaya L. Probability density function for modulated MAP event flows with unextendable dead time // Communications in Computer and Information Science. 2015. V. 564. P. 141–151.
33. Nezhel'skaya L. Estimation of the unextendable dead time period in a flow of physical events by the method of maximum likelihood // Russian Physics Journal. 2016. V. 59, No. 5. P. 651–662.

Горцев Александр Михайлович, д-р техн. наук, профессор. E-mail: gam@fpmk.tsu.ru

Национальный исследовательский Томский государственный университет

Завгородняя Мария Евгеньевна, канд. техн. наук. E-mail: mari.zavgor@mail.ru

Национальный исследовательский Томский государственный университет

Поступила в редакцию 21 марта 2017 г.

Gortsev Alexander M., Zavgorodnyaya Maria E. (National Research Tomsk State University, Russian Federation).

Estimation of the parameter of unextendable dead time random duration in the Poisson flow of events.

Keywords: the simplest flow; an unextendable dead time; parameter estimates; the maximum likelihood method; the method of moments.

We consider a Poisson flow of intensity events λ . After each event registered at time t_k , a dead time period of random duration occurs during which other events of the flow are unavailable to the observation and do not cause it to be prolonged (unextendable dead time). At the end of the dead time period, the first event that occurred again creates a period of dead time of random duration, etc. It is assumed that the dead time duration is distributed exponentially with the parameter α .

The stationary mode of operation of the observed event flow is studied, therefore we neglect the transient processes on the observation half-line $(t_0, t]$, where t_0 is the beginning of the observations, t is the end of the observations, then without loss of generality we can set $t_0 = 0$.

It is assumed that the parameters of the observed flow are unknown. It is necessary, at the end of the observations (at time t), based on the sample $t_1, t_2, \dots, t_n$, the observed moments of the onset of the flow events, to make the maximum likelihood method and the method of moments constructing the estimates $\hat{\lambda}$, $\hat{\alpha}$ and to compare the quality of the estimates obtained.

The likelihood function is written out

$$L(\alpha, \lambda | \tau_1, \dots, \tau_n) = \left(\frac{\lambda \alpha}{\lambda - \alpha} \right)^n \prod_{k=1}^n (e^{-\alpha \tau_k} - e^{-\lambda \tau_k})$$

and likelihood equations

$$-\frac{n\alpha}{\lambda(\lambda - \alpha)} + \sum_{k=1}^n \tau_k \frac{e^{\alpha \tau_k}}{e^{\lambda \tau_k} - e^{\alpha \tau_k}} = 0, \quad \frac{n\lambda}{\alpha(\lambda - \alpha)} - \sum_{k=1}^n \tau_k \frac{e^{\lambda \tau_k}}{e^{\lambda \tau_k} - e^{\alpha \tau_k}} = 0,$$

where, $\tau_k = t_{k+1} - t_k$, $k = 1, 2, \dots$, is the sequence of the values of the lengths of intervals between adjacent events of the flow measured during the observation of the flow during the observation interval $(0, t]$. It is shown that the maximum likelihood method, regardless of the ratio of the unknown parameters λ and α , provides estimates $\hat{\lambda}$ and $\hat{\alpha}$ in the form $\hat{\alpha} = \hat{\lambda} = 2 / \hat{\tau}$, i.e. its performance only applies to the option $\alpha = \lambda$.

For the method of moments it is proved that the equations of moments

$$\lambda + \alpha = \hat{\tau}_1 \alpha \lambda; \quad 2 \left(\frac{1}{\alpha^2} + \frac{1}{\alpha \lambda} + \frac{1}{\lambda^2} \right) = \hat{\tau}_2 \alpha^2 \lambda^2$$

relatively unknown parameters and have no solution.

For the case of a known parameter λ , an unknown parameter α is estimated both by the maximum likelihood method and by the moment method. In this situation, the numerical results show the advantage of the maximum likelihood method.

REFERENCES

1. Kurochkin, S.S. (1968) *Mnogomernye statisticheskie analizatory* [Multivariate statistical analyzers]. Moscow: Atomizdat.
2. Apanasovich, V.V., Kolyada, A.A. & Chernyavskiy, A.F. (1988) *Statisticheskiy analiz sluchaynykh potokov v fizicheskom eksperimente* [The statistical analysis of series of random events in physical experiment]. Minsk: University Press.
3. Gortsev, A.M. & Klimov, I.S. (1991) Otsenka intensivnosti puassonovskogo potoka sobyitiy v usloviyakh chastichnoy ego nenablyudaemosti [Intensity estimation of the Poisson flow of events in condition of its incomplete observability]. *Radiotekhnika*. 12. pp. 3–7.
4. Gortsev, A.M. & Klimov, I.S. (1996) Otsenivanie perioda nenablyudaemosti i intensivnosti puassonovskogo potoka sobyitiy [The estimation of intensity process and period of unobservability of the Poisson flow of events]. *Radiotekhnika*. 2. pp. 8–11.
5. Gortsev, A.M. & Nezhelskaya, L.A. (2003) Dead time period and parameter estimation of synchronous alternating flow of events. *Vestnik Tomskogo gosudarstvennogo universiteta. Prilozhenie – Tomsk State University Journal. Supplement*. 6. pp. 232–239. (In Russian).
6. Vasilevskaya, T.P., Gortsev, A.M. & Nezhelskaya, L.A. (2004) Dead time and parameters estimation of synchronous alternating flow with or without event manifestation. *Vestnik Tomskogo gosudarstvennogo universiteta. Prilozhenie – Tomsk State University Journal. Supplement*. 9. pp. 129–138.
7. Gortsev, A.M. & Nissenbaum, O.V. (2005) Estimation of the dead time period and parameters of an asynchronous alternative flow of events with unextendable dead time period. *Russian Physics Journal*. 10. pp. 35–49. DOI: 10.1007/s11182-006-0023-y

8. Gortsev, A.M. & Nissenbaum, O.V. (2004) Dead time and parameter estimation of asynchronous alternating flow with additional event initiation. *Tomsk State University Journal*. 284. pp. 137–145. (In Russian).
9. Gortsev, A.M. & Parshina, M.E. (1999) Parameter estimation of alternating flow of events under conditions of dead time. *Russian Physics Journal*. 4. pp. 8–13.
10. Gortsev, A.M. & Zavgorodnyaya, M.E. (1997) Otsenka parametrov al'terniruyushchego potoka sobytii pri uslovii ego chastichnoy nablyu-daemosti [Parameter estimation of alternating flow of events under conditions of particulate observability]. *Optika atmosfery i okeana – Atmospheric and Oceanic Optics*. 10(3). pp. 273–280.
11. Bushlanov, I.V., Gortsev, A.M. & Nezhelskaya, L.A. (2008) Estimating parameters of the synchronous twofold-stochastic flow of events. *Automation and Remote Control*. 69(9). pp. 1517–1533. DOI: 10.1134/S0005117908090075
12. Gortsev, A.M. & Nezhelskaya, L.A. (2004) Estimation of the dead time period and intensities of the synchronous double stochastic event flow. *Radioengineering*. 10. pp. 8–16.
13. Vasileva, L.A. & Gortsev, A.M. (2003) Estimation of the dead time of an asynchronous double stochastic flow of events under incomplete observability. *Automation and Remote Control*. 64(12). pp. 1890–1898. DOI: 10.1023/B:AURC.0000008427.99676.df
14. Vasileva, L.A. & Gortsev, A.M. (2002) Estimation of parameters of a double-stochastic flow of events under conditions of its incomplete observability. *Automation and Remote Control*. 63(3). pp. 511–515. DOI: 10.1023/A:1014718921138
15. Gortsev, A.M. & Kusnatdinov, R.T. (1998) Estimation of states of the MC-flow of events under its partial observation. *Russian Physics Journal*. 41(4). pp. 22–30.
16. Gortsev, A.M. & Nezhelskaya, L.A. (2003) Estimation of the dead-time period and parameters of a semi-synchronous double-stochastic stream of events. *Measurement Techniques*. 46(6). pp. 536–545. DOI: 10.1023/A:1025499509015
17. Gortsev, A.M. & Nezhelskaya, L.A. (2008) Semi-synchronous twice-stochastic event flow in conditions of prolonged dead time. *Vychislitel'nye tekhnologii – Computational Technologies*. 13(1). pp. 31–41. (In Russian).
18. Nezhelskaya, L.A. (2000) Optimal state estimation of semi-synchronous flow in conditions of its incomplete observability. *Vestnik Tomskogo gosudarstvennogo universiteta – Tomsk State University Journal*. 269. pp. 95–98. (In Russian).
19. Gortsev, A.M., Leonova, M.A. & Nezhelskaya, L.A. (2012) Joint probability density function of interval duration of generic asynchronous event flow in conditions of fixed dead time. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science*. 4(21). pp. 14–25. (In Russian).
20. Leonova, M.A. & Nezhelskaya, L.A. (2013) Maximum likelihood estimation of dead time value at a generalized asynchronous flow of events. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science*. 2(23). pp. 54–63. (In Russian).
21. Gortsev, A.M., Leonova, M.A. & Nezhelskaya, L.A. (2013) Comparison of MP- and MM-estimations of dead time in generic asynchronous flow. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science*. 4(25). pp. 32–42. (In Russian).
22. Gortsev, A.M., Kalyagin, A.A. & Nezhelskaya, L.A. (2015) Maximum likelihood estimation of dead time of generic semichronous flow. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science*. 1(30). pp. 27–37. (In Russian).
23. Gortsev, A.M., Kalyagin, A.A. & Nezhelskaya, L.A. (2014) The joint probability density of duration of the intervals in a generalized semi-synchronous flow of events with unprolonging dead time. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science*. 2(27). pp. 19–29. (In Russian).
24. Gortsev, A.M. & Solovyev, A.A. (2014) Joint probability density of interarrival interval of a flow of physical events with unextendable dead time period. *Russian Physics Journal*. 57(7). pp. 973–983. (In Russian).
25. Gortsev, A.M. & Solovyev, A.A. (2015) Maximum likelihood estimation of fixed dead time in physical flow of events. *Russian Physics Journal*. 58(11). pp. 141–149. (In Russian).
26. Sirotina, M.N. & Gortsev, A.M. (2016) Maximum likelihood function estimation of dead time duration in modulated synchronous doubly stochastic flow of events. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science*. 1(34). pp. 50–64. (In Russian).
27. Gortsev, A. & Sirotina, A. (2015) Joint probability density function of modulated synchronous flow interval duration under conditions of fixed dead time. *Communications in Computer and Information Science*. 564. pp. 41–52. DOI: 10.1007/978-3-319-25861-4_4
28. Bakholdina, M.A. & Gortsev, A.M. (2014) Optimal states estimation of the modulated semi-synchronous integrated flow of events in the condition of constant dead time. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science*. 1(26). pp. 13–24.
29. Bakholdina, M. & Gortsev, A. (2015) Joint probability density of the intervals length of modulated semi-synchronous integrated flow of events in conditions of a constant dead time and the flow recurrence conditions. *Communications in Computer and Information Science*. 564. pp. 13–27. DOI: 10.1007/978-3-319-13671-4_3
30. Nezhelskaya, L. (2014) Optimal state estimation in modulated MAP event flows with unextendable dead time. *Communications in Computer and Information Science*. 487. pp. 342–350. DOI: 10.1007/978-3-319-13671-4_39
31. Nezhelskaya, L.A. (2015) Conditions for recurrence of flow of physical events with unextendable dead time period. *Russian Physics Journal*. 58(12). pp. 168–175. (In Russian). DOI: 10.1007/s11182-016-0727-6
32. Nezhelskaya, L. (2015) Probability density function for modulated MAP event flows with unextendable dead time. *Communications in Computer and Information Science*. 564. pp. 141–151. DOI: 10.1007/978-3-319-25861-4_12
33. Nezhelskaya, L. (2016) Estimation of the unextendable dead time period in a flow of physical events by the method of maximum likelihood. *Russian Physics Journal*. 59(5). pp. 651–662. (In Russian). DOI: 10.1007/s11182-016-0818-4

Г.А. Медведев

МОДЕЛЬ ВРЕМЕННОЙ СТРУКТУРЫ ДОХОДНОСТИ ЛОНГСТАФФА–ШВАРЦА И ЕЕ РАСШИРЕНИЕ

Рассматривается модель Лонгстаффа–Шварца как в пространстве латентных переменных состояний, так и в пространстве наблюдаемых (или оцениваемых) переменных состояний. Получены аналитические выражения для кривых доходности до погашения и форвардных кривых в обоих случаях. Предложено расширение модели на произвольную размерность пространства состояний. В рамках этого расширения предложен способ получения аналитических решений уравнений относительно функций временной структуры доходности, когда исходные уравнения динамики краткосрочной процентной ставки приводят к нелинейным системам уравнений Риккати относительно этих функций, не позволяющих известными методами получить аналитическое решение. Этот способ основан на очевидном утверждении, что если задан процесс краткосрочной процентной ставки, то соответствующая ему временная структура доходности не зависит от того, каким способом описывается пространство переменных состояния финансового рынка. Приводятся численные примеры.

Ключевые слова: временная структура доходности; краткосрочная процентная ставка; модель Лонгстаффа–Шварца; системы уравнений Риккати.

Временная структура бескупонной доходности является одной из востребованных характеристик, которая используется при определении стоимости финансовых активов. Однако до сих пор временные структуры в аналитической форме удавалось получить только для аффинных систем доходности, как правило, для однофакторных моделей. Поскольку однофакторные модели описывают ситуацию на финансовом рынке недостаточно точно, возникают модели с использованием большего числа факторов. Одной из первых таких моделей является двухфакторная модель Лонгстаффа–Шварца (1992). Она основана на использовании так называемых латентных (скрытых) факторов, т.е. переменных состояния, которые непосредственно на рынке не наблюдаются. Считается, что динамика этих факторов описывается процессами «с квадратным корнем» Кокса–Ингерсолла–Росса (CIR). Затем, если удастся связать эти процессы с реально наблюдаемыми показателями, то можно придать анализу такой модели реальный смысл. Это свойство может быть использовано в более общих обстоятельствах, когда рассматривается модель с произвольным числом латентных факторов.

1. Модель Лонгстаффа–Шварца

Известной версией двухфакторной модели CIR является модель Лонгстаффа–Шварца [1]. В этой модели в качестве исходных переменных состояния выбираются два независимых процесса CIR, имеющих смысл неких ненаблюдаемых экономических факторов:

$$\begin{aligned} dx &= (a - bx)dt + \sqrt{x}dW_1, \\ dy &= (d - ey)dt + \sqrt{y}dW_2. \end{aligned} \quad (1)$$

В дальнейшем считается, что взвешенная сумма этих факторов образует краткосрочную процентную ставку

$$r(t) = \alpha x(t) + \beta y(t), \quad (2)$$

где α и β – неотрицательные константы. Использование исходных уравнений и формулы Ито позволяет написать

$$\begin{aligned} dr(t) &= \alpha dx(t) + \beta dy(t) = \\ &= [\alpha a + \beta d - bx(t) - ey(t)]dt + \alpha \sqrt{x(t)} dW_1(t) + \beta \sqrt{y(t)} dW_2(t). \end{aligned}$$

Заметим, что стационарные математические ожидания процессов $x(t)$ и $y(t)$ определяются равенствами $E[x(t)] = a/b$ и $E[y(t)] = d/e$. По тем же причинам квадрат $dr(t)$ имеет вид

$$(dr(t))^2 = \alpha^2 x(t) dt + \beta^2 y(t) dt.$$

Локальная (по времени) дисперсия изменения процентной ставки за интервал времени $(t, t + dt)$ на единицу времени равна

$$V(t) = \alpha^2 x(t) + \beta^2 y(t). \quad (3)$$

Из (2) и (3) следует, что между парами (x, y) и (r, V) имеется взаимно однозначная связь, определяемая соотношениями

$$\begin{aligned} r &= \alpha x + \beta y, & V &= \alpha^2 x + \beta^2 y, \\ x &= \frac{\beta r - V}{\alpha(\beta - \alpha)}, & y &= \frac{V - \alpha r}{\beta(\beta - \alpha)}. \end{aligned} \quad (4)$$

Поэтому более естественно рассматривать в качестве переменных состояния не ненаблюдаемые неопределенные «экономические факторы» (x, y) , а имеющие вполне определенный смысл процентную ставку r и ее локальную дисперсию V . Однако следует заметить, что если областью задания переменных (x, y) является весь первый квадрант плоскости (X, Y) , то переменные (r, V) принимают значения только в ограниченной области первого квадранта, определяемой неравенствами $\alpha r < V < \beta r$ (или $V/\beta < r < V/\alpha$), $0 < \alpha < \beta$, в которой малым ставкам могут соответствовать только малые дисперсии (или малые дисперсии имеют только малые процентные ставки). Имея ввиду существующее линейное преобразование

$$\begin{pmatrix} r \\ V \end{pmatrix} = \begin{pmatrix} \alpha & \beta \\ \alpha^2 & \beta^2 \end{pmatrix} \begin{pmatrix} X \\ Y \end{pmatrix}, \quad (5)$$

вместо системы уравнений (1) по формуле Ито получим соответствующую систему для (r, V) :

$$d \begin{pmatrix} r \\ V \end{pmatrix} = K \left(\theta - \begin{pmatrix} r \\ V \end{pmatrix} \right) dt + \sigma(r, V) d \begin{pmatrix} W_1 \\ W_2 \end{pmatrix}, \quad (6)$$

где

$$\begin{aligned} K &= \frac{1}{\alpha - \beta} \begin{pmatrix} e\alpha - b\beta & b - e \\ \alpha\beta(e - b) & b\alpha - e\beta \end{pmatrix}, \quad \theta = \begin{pmatrix} \alpha & \beta \\ \alpha^2 & \beta^2 \end{pmatrix} \begin{pmatrix} a/b \\ d/e \end{pmatrix}, \\ \sigma(r, V) &= \begin{pmatrix} \alpha & \beta \\ \alpha^2 & \beta^2 \end{pmatrix} \times \begin{pmatrix} \sqrt{\frac{(\beta r - V)}{\alpha(\beta - \alpha)}} & 0 \\ 0 & \sqrt{\frac{(V - \alpha r)}{\beta(\beta - \alpha)}} \end{pmatrix}. \end{aligned}$$

Для определения дифференциальных уравнений для функций временной структуры $A(\tau)$ и $B(\tau)$ применим технику, описанную в [2]. Для получения системы уравнений (4), (5) из [2] воспользуемся разложением по переменным r и V элементов стохастического дифференциального уравнения (6). При составлении уравнений (5) из [2] весовые коэффициенты ϕ_i выбирались таким образом, чтобы было выполнено необходимое условие $\lim_{\tau \rightarrow 0} y(\tau, x) = r$. Так что если в качестве переменных состояния принимаются

(r, V) , то $\phi_r = 1$, а $\phi_V = 0$. Если в качестве переменных состояния используются (x, y) , то $\phi_x = \alpha$, а $\phi_y = \beta$. Заметим, что

$$\begin{aligned} K\theta &= \begin{pmatrix} \alpha a + \beta d \\ \alpha^2 a + \beta^2 d \end{pmatrix}, \quad K_r = \frac{1}{\alpha - \beta} \begin{pmatrix} e\alpha - b\beta \\ \alpha\beta(e - b) \end{pmatrix}, \quad K_V = \frac{1}{\alpha - \beta} \begin{pmatrix} b - e \\ b\alpha - e\beta \end{pmatrix}, \\ \sigma(r, V)\sigma(r, V)^T &= \beta_0 + \beta_r r + \beta_V V, \\ \beta_0 &= 0, \quad \beta_r = -\alpha\beta \begin{pmatrix} 0 & 1 \\ 1 & \alpha + \beta \end{pmatrix}, \quad \beta_V = \begin{pmatrix} 1 & \alpha + \beta \\ \alpha + \beta & \alpha^2 + \alpha\beta + \beta^2 \end{pmatrix}, \\ \sigma(r, V)\lambda(r, V)^T &= \xi + \eta_r r + \eta_V V, \\ \xi &= 0, \quad \eta_r = \frac{1}{\beta - \alpha} \begin{pmatrix} \beta \lambda_r - \alpha \lambda_V \\ \alpha\beta(\lambda_r - \lambda_V) \end{pmatrix}, \quad \eta_V = \frac{1}{\alpha - \beta} \begin{pmatrix} \lambda_r - \lambda_V \\ \alpha \lambda_r - \beta \lambda_V \end{pmatrix}. \end{aligned}$$

Здесь предполагается, что рыночные цены риска определяются соотношением $\lambda(r, V) = (\lambda_r, \lambda_V)\sigma(r, V)$, где λ_r и λ_V – константы. Подставляя полученные разложения в уравнения (4), (5) из [2], находим систему уравнений для функций временной структуры $A(\tau)$, $B_r(\tau)$ и $B_V(\tau)$:

$$A'(\tau) = -(\alpha a + \beta d) B_r(\tau) - (\alpha^2 a + \beta^2 d) B_V(\tau), \quad A(0) = 0, \quad (7)$$

$$B_r'(\tau) = 1 - B(\tau)^T(\eta_r + K_r) - B(\tau)^T \beta_r B(\tau)/2 = \quad (8)$$

$$= 1 - \frac{1}{\alpha - \beta} ((e\alpha - b\beta - \beta\lambda_r + \alpha\lambda_V) B_r(\tau) + \alpha\beta(e - b - \lambda_r + \lambda_V) B_V(\tau)) +$$

$$+ \alpha\beta B_V(\tau)(B_r(\tau) + (\alpha + \beta) B_V(\tau)/2), \quad B_r(0) = 0,$$

$$B_V'(\tau) = -B(\tau)^T(\eta_V + K_V) - \frac{1}{2} B(\tau)^T \beta_V B(\tau) = \quad (9)$$

$$= -\frac{1}{\alpha - \beta} ((b - e + \lambda_r - \lambda_V) B_r(\tau) + (b\alpha - e\beta + \alpha\lambda_r - \beta\lambda_V) B_V(\tau)) -$$

$$- \frac{1}{2} B_r(\tau)^2 - (\alpha + \beta) B_r(\tau) B_V(\tau) - \frac{1}{2} (\alpha^2 + \alpha\beta + \beta^2) B_V(\tau)^2, \quad B_V(0) = 0.$$

К сожалению, полученная система обыкновенных дифференциальных уравнений для (B_r, B_V) относится к системам уравнений Риккати, является нелинейной и не известно какого-либо способа для получения ее аналитического решения. Вместе с тем задача определения временной структуры рассматриваемой модели при использовании исходных ненаблюдаемых переменных состояния (x, y) имеет аналитическое решение. Действительно, для системы (1) получаются следующие данные для составления уравнений (4), (5) из [2]:

$$K = \begin{pmatrix} b & 0 \\ 0 & e \end{pmatrix}, \quad \theta = \begin{pmatrix} a/b \\ d/e \end{pmatrix}, \quad K\theta = \begin{pmatrix} a \\ d \end{pmatrix},$$

$$\sigma(x, y)\sigma^T(x, y) = \begin{pmatrix} x & 0 \\ 0 & y \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} y,$$

$$\beta_0 = 0, \quad \beta_x = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad \beta_y = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

Если положить, что рыночные цены риска определяются соотношением $\lambda(x, y) = (\lambda_x, \lambda_y)\sigma(x, y)$, где λ_x и λ_y – константы, тогда

$$\sigma(x, y)\lambda^T(x, y) = \begin{pmatrix} \lambda_x x \\ \lambda_y y \end{pmatrix} = \begin{pmatrix} \lambda_x \\ 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ \lambda_y \end{pmatrix} y,$$

$$\xi = 0, \quad \eta_x = \begin{pmatrix} \lambda_x \\ 0 \end{pmatrix}, \quad \eta_y = \begin{pmatrix} 0 \\ \lambda_y \end{pmatrix}.$$

Система дифференциальных уравнений (4), (5) из [2] для этих данных имеет вид

$$A'(\tau) = -a B_x(\tau) - d B_y(\tau), \quad A(0) = 0, \quad (10)$$

$$B_x'(\tau) = \alpha - (\lambda_x + b) B_x(\tau) - B_x^2(\tau)/2, \quad B_x(0) = 0, \quad (11)$$

$$B_y'(\tau) = \beta - (\lambda_y + e) B_y(\tau) - B_y^2(\tau)/2, \quad B_y(0) = 0. \quad (12)$$

Два последних уравнения уже не составляют систему, а являются независимыми скалярными уравнениями Риккати, имеющими аналитическое решение

$$B_x(\tau) = \alpha \left(\frac{\varepsilon_x}{e^{\varepsilon_x \tau} - 1} + J_x \right)^{-1}, \quad \varepsilon_x = \sqrt{(b + \lambda_x)^2 + 2\alpha}, \quad J_x = (\varepsilon_x + \lambda_x + b)/2; \quad (13)$$

$$B_y(\tau) = \beta \left(\frac{\varepsilon_y}{e^{\varepsilon_y \tau} - 1} + J_y \right)^{-1}, \quad \varepsilon_y = \sqrt{(e + \lambda_y)^2 + 2\beta}, \quad J_y = (\varepsilon_y + \lambda_y + e)/2. \quad (14)$$

Подстановка этих выражений в уравнение для функции $A(\tau)$ приводит к следующему результату:

$$A(\tau) = -a\alpha \frac{J_x \tau - \ln[1 + (e^{\varepsilon_x \tau} - 1)J_x / \varepsilon_x]}{J_x(J_x - \varepsilon_x)} - \\ - d\beta \frac{J_y \tau - \ln[1 + (e^{\varepsilon_y \tau} - 1)J_y / \varepsilon_y]}{J_y(J_y - \varepsilon_y)}. \quad (15)$$

Наконец, кривая доходности вычисляется в аналитическом виде

$$y(\tau | x, y) = \frac{x B_x(\tau) + y B_y(\tau) - A(\tau)}{\tau} = \\ = \frac{x\alpha}{\tau} \left(\frac{\varepsilon_x}{e^{\varepsilon_x \tau} - 1} + J_x \right)^{-1} + \frac{y\beta}{\tau} \left(\frac{\varepsilon_y}{e^{\varepsilon_y \tau} - 1} + J_y \right)^{-1} + \\ + a\alpha \frac{J_x \tau - \ln[1 + (e^{\varepsilon_x \tau} - 1)J_x / \varepsilon_x]}{J_x(J_x - \varepsilon_x)\tau} + d\beta \frac{J_y \tau - \ln[1 + (e^{\varepsilon_y \tau} - 1)J_y / \varepsilon_y]}{J_y(J_y - \varepsilon_y)\tau}. \quad (16)$$

Для получения форвардной кривой можно воспользоваться формулой (6) из [2], что дает

$$f(\tau | x, y) = x \frac{dB_x(\tau)}{d\tau} + y \frac{dB_y(\tau)}{d\tau} - \frac{dA(\tau)}{d\tau} = \\ = x\alpha + y\beta + (a - xb - x\lambda_x)B_x(\tau) + \\ + (d - ye - y\lambda_y)B_y(\tau) - x B_x^2(\tau)/2 - y B_y^2(\tau)/2, \quad (17)$$

где функции $B_x(\tau)$ и $B_y(\tau)$ явно вычисляются по формулам (13) и (14).

Заметим, что

$$\lim_{\tau \rightarrow 0} y(\tau | x, y) = \lim_{\tau \rightarrow 0} f(\tau | x, y) = x\alpha + y\beta = r, \quad (18)$$

$$\lim_{\tau \rightarrow \infty} y(\tau | x, y) = \lim_{\tau \rightarrow \infty} f(\tau | x, y) = \frac{a\alpha}{J_x} + \frac{d\beta}{J_y}. \quad (19)$$

Вместе с тем, какими бы переменными состояния ни описывалось поведение процесса процентных ставок в рассматриваемой модели, доходность до погашения для некоторого срока до погашения τ должна быть одинаковой для любых способов описания процесса процентных ставок, если преобразование переменных состояний задает взаимно однозначное соответствие переменных. В нашем случае для переменных состояния (x, y) и (r, V) это имеет место, поскольку матрица линейного преобразования (5) является невырожденной. Следовательно, должно выполняться следующее равенство:

$$y(\tau | x, y) = y(\tau | r, V).$$

Поэтому для получения аналитического выражения для доходности до погашения $y(\tau | r, V)$ достаточно выразить в формуле (13) переменные состояния x и y через r и V по формуле (4). Тогда после упрощения получим

$$y(\tau | r, V) = \frac{1}{\tau} \left(\frac{\alpha^2 B_y(\tau) - \beta^2 B_x(\tau)}{\alpha\beta(\alpha - \beta)} r - \frac{\alpha B_y(\tau) - \beta B_x(\tau)}{\alpha\beta(\alpha - \beta)} V - A(\tau) \right), \quad (20)$$

где функции временной структуры $A(\tau)$, $B_x(\tau)$ и $B_y(\tau)$ вычисляются по формулам (13)–(15). Выражение доходности до погашения (20) предпочтительнее формулы (16), поскольку в нем используются переменные r и V , которые можно наблюдать или оценивать, в то время как в (16) используются ненаблюдаемые переменные x и y , не имеющие явной финансовой трактовки.

Аналогично для получения аналитического выражения для форвардной кривой $f(\tau | r, V)$ достаточно в формуле (17) переменные состояния x и y выразить через r и V по формулам (4). Это дает

$$f(\tau | r, V) = a B_x(\tau) + d B_y(\tau) + \\ + \left(1 + \beta \frac{(b + \lambda_x) B_x(\tau) + B_x(\tau)^2 / 2}{\alpha(\alpha - \beta)} - \alpha \frac{(e + \lambda_y) B_y(\tau) + B_y(\tau)^2 / 2}{\beta(\alpha - \beta)} \right) r -$$

$$-\left(\frac{(b + \lambda_x)B_x(\tau) + B_x(\tau)^2 / 2}{\alpha(\alpha - \beta)} - \frac{(e + \lambda_y)B_y(\tau) + B_y(\tau)^2 / 2}{\beta(\alpha - \beta)} \right) V. \quad (21)$$

Таким образом, кривые доходности в пространстве (r, V) могут быть рассчитаны по формулам (20), (21) в аналитическом виде или численно с помощью «аналитически неразрешимых» нелинейных уравнений (7)–(9). Вычисления показывают, что оба этих способа приводят к полностью совпадающим кривым. Поэтому функции $B_r(\tau)$ и $B_l(\tau)$, удовлетворяющие системе уравнений (8), (9), можно записать в следующем аналитическом виде:

$$B_r(\tau) = \frac{\alpha^2 B_y(\tau) - \beta^2 B_x(\tau)}{\alpha\beta(\alpha - \beta)}, \quad B_l(\tau) = \frac{\beta B_x(\tau) - \alpha B_y(\tau)}{\alpha\beta(\alpha - \beta)}, \quad (22)$$

где функции $B_x(\tau)$ и $B_y(\tau)$ вычисляются по формулам (13)–(15). Справедливость этого утверждения легко проверяется путем подстановки выражений (22) в уравнения (8) и (9).

Заметим, что аналитические выражения для кривой доходности до погашения для рассматриваемой модели были найдены Ф. Лонгстаффом и Э. Швартцем (1992) в другом, более громоздком виде.

На рисунке 1 кривые $y(\tau | r, V)$ и $f(\tau | r, V)$ представлены для нейтральной к риску среды ($\lambda_x = \lambda_y = 0$) при следующих значениях параметров:

$$r = 0,06, V = 0,03, a = 0,3, b = 4, d = 0,5, e = 1,7, \alpha = 0,3, \beta = 0,7.$$

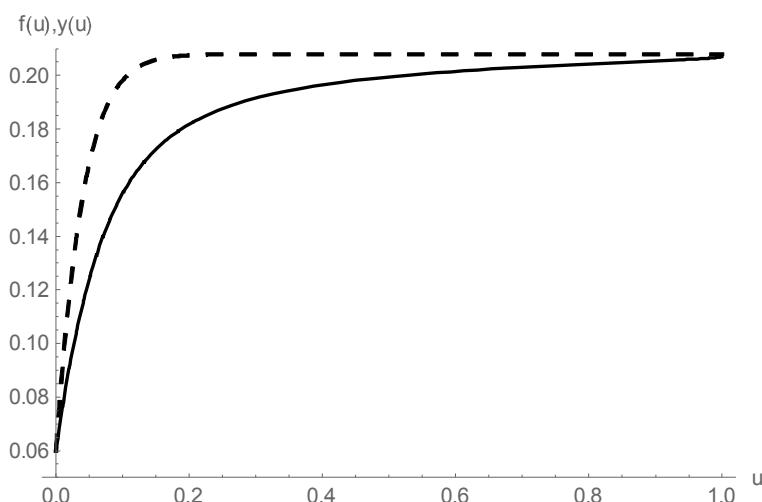


Рис. 1. Кривая доходности до погашения $y(u)$ и кривая форвардных ставок $f(u)$

Для представления кривых «целиком» для всего интервала возможных значений сроков до погашения $\tau \in (0, \infty)$ использовано нелинейное преобразование сроков до погашения: $u = 1 - e^{-\rho \tau}$, которое отображает положительную полуось $(0, \infty)$ в единичный интервал $(0, 1)$. Принятое при расчетах численное значение $\rho = \ln 10 / 30 = 0,07675$ соответствует тому, что сроки до погашения от 0 до 30 лет отображаются в интервал $(0; 0,9)$. Кривые стартуют из точки $y(0 | r, V) = f(0 | r, V) = r = 0,06$ и при $u \rightarrow 1$ стремятся к одному и тому же пределу $y(1 | r, V) = f(1 | r, V) = 0,2079$. Сплошной линией показана кривая доходности до погашения $y(u)$, а пунктирной – кривая форвардных ставок $f(u)$.

2. Расширение модели Лонгстаффа–Швартца

Рассмотренную модель Лонгстаффа–Швартца можно отнести к классу двухфакторных моделей Кокса–Ингерсолла–Росса. Однако полученные результаты позволяют расширить эту модель на случай n факторов. Действительно, пусть $X = (x_1, x_2, \dots, x_n)^T$ обозначает n -мерный вектор неких латентных переменных состояния рынка.

Уравнения динамики этих переменных состояния, расширяя на этот случай модель Лонгстаффа – Швартца (1), запишем в виде

$$dx_i = \kappa_i(a_i - x_i)dt + \sqrt{x_i}dW_i, \quad 1 \leq i \leq n. \quad (23)$$

Предположим, что латентные переменные X связаны невырожденным линейным преобразованием

$$Z = HX \quad (24)$$

с вектором других переменных состояния $Z = (z_1, z_2, \dots, z_n)^T$, компоненты которого наблюдаемы и имеют вполне определенный экономический или статистический смысл. Например, z_1 – это краткосрочная процентная ставка, z_2 – ее мгновенная дисперсия и т.д., H – невырожденная матрица с компонентами h_{is} , $1 \leq i, s \leq n$. Тогда в соответствии со стохастическим анализом Ито для динамики вектора переменных состояния Z получается уравнение, аналогичное уравнению (6):

$$dZ = K(\theta - Z)dt + H D(\sqrt{H^{-1}Z})dW, \quad (25)$$

где $K = HD(\kappa)H^{-1}$, $\theta = Ha$. В (23) κ – вектор с компонентами κ_i ; a – вектор с компонентами a_i ; $D(\kappa)$ – диагональная матрица, на главной диагонали которой располагаются элементы вектора κ ; $D(\sqrt{H^{-1}Z})$ – диагональная матрица, на главной диагонали которой располагаются квадратные корни элементов вектора $H^{-1}Z$.

Относительно матрицы преобразования H сделаем следующие предположения. Удобно компоненту z_1 вектора состояний Z отождествлять с краткосрочной процентной ставкой r , так что согласно представлению (24) первая строка матрицы H должна состоять из элементов, обеспечивающих равенство $z_1 \equiv r = \sum_{i=1}^n h_{1i}x_i$. Кроме того, элементы матрицы H должны быть такими, чтобы было выполнено необходимое условие (18) для кривых доходности $\lim_{\tau \rightarrow 0} y(\tau | X) = r$.

Обозначим кривую доходности до погашения в случае, когда переменные состояния определяются вектором X как $y(\tau|X)$, и, соответственно, кривую доходности до погашения в случае, когда переменные состояния определяются вектором Z как $y(\tau|Z)$. Характер динамики переменных состояния в обоих случаях обеспечивает тот факт, что кривые доходности будут относиться к аффинному классу, т.е. $y(\tau|X) = A(\tau) + X^T B(\tau)$ и $y(\tau|Z) = a(\tau) + Z^T b(\tau)$. Очевидно, что в каких бы координатах ни описывалось текущее состояние какого-либо конкретного процесса краткосрочной процентной ставки, кривые доходности для этого процесса должны совпадать, т.е. $y(\tau|X) = y(\tau|Z)$ или $A(\tau) + X^T B(\tau) = a(\tau) + Z^T b(\tau)$. Поскольку эти равенства должны иметь место для любых сроков до погашения τ и любых значений переменных состояния X и Z (и нулевых тоже), то имеем $a(\tau) = A(\tau)$, $X^T B(\tau) = Z^T b(\tau) = X^T H^T b(\tau)$. Таким образом, если функции временной структуры $A(\tau)$ и $B(\tau)$ для кривой доходности $y(\tau|X)$ известны, для кривой доходности $y(\tau|Z)$ функции временной структуры $a(\tau)$ и $b(\tau)$ находятся из соотношений

$$a(\tau) = A(\tau), \quad b(\tau) = (H^T)^{-1}B(\tau). \quad (26)$$

Для определения дифференциальных уравнений относительно функций временной структуры $A(\tau)$ и $B(\tau)$ снова применим подход, описанный в [2]. Для системы (23) получаются следующие данные для составления уравнений (4), (5) из [2]:

$$K = \begin{pmatrix} \kappa_1 & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & \kappa_n \end{pmatrix}, \quad \theta = \begin{pmatrix} a_1 \\ \dots \\ a_n \end{pmatrix}, \quad K\theta = \begin{pmatrix} \kappa_1 a_1 \\ \dots \\ \kappa_n a_n \end{pmatrix}.$$

$$\sigma(X)\sigma^T(X) = \begin{pmatrix} x_1 & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & x_n \end{pmatrix} = \begin{pmatrix} 1 & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & 0 \end{pmatrix} x_1 + \dots + \begin{pmatrix} 0 & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & 1 \end{pmatrix} x_n,$$

$$\beta_0 = 0, \quad \beta_1 = \begin{pmatrix} 1 & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & 0 \end{pmatrix}, \dots, \beta_n = \begin{pmatrix} 0 & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & 1 \end{pmatrix}.$$

Если положить, что рыночные цены риска определяются соотношением $\lambda(X) = (\lambda_1 \lambda_2 \dots \lambda_n) \sigma(X)$, где $\lambda_1, \lambda_2, \dots, \lambda_n$ – константы, тогда

$$\sigma(X)\lambda^T(X) = \begin{pmatrix} \lambda_1 x_1 \\ \dots \\ \lambda_n x_n \end{pmatrix} = \begin{pmatrix} \lambda_1 \\ \dots \\ 0 \end{pmatrix} x_1 + \dots + \begin{pmatrix} 0 \\ \dots \\ \lambda_n \end{pmatrix} x_n,$$

$$\xi = 0, \quad \eta_1 = \begin{pmatrix} \lambda_1 \\ \dots \\ 0 \end{pmatrix}, \dots, \eta_n = \begin{pmatrix} 0 \\ \dots \\ \lambda_n \end{pmatrix}.$$

Следовательно, система дифференциальных уравнений (4), (5) из [2] в этом случае имеет вид

$$A'(\tau) = - \sum_{i=1}^n a_i \kappa_i B_i(\tau), \quad A(0) = 0, \quad (27)$$

$$B_i'(\tau) = h_{1i} - (\kappa_i + \lambda_i) B_i(\tau) - B_i^2(\tau)/2, \quad B_i(0) = 0, \quad 1 \leq i \leq n. \quad (28)$$

Полученные уравнения с точностью до параметров совпадают с уравнениями (11) и (12), поэтому их решения имеют тот же вид, что и (13):

$$B_i(\tau) = h_{1i} \left(\frac{\varepsilon_i}{e^{\varepsilon_i \tau} - 1} + J_i \right)^{-1}, \quad \varepsilon_i = \sqrt{(\kappa_i + \lambda_i)^2 + 2h_{1i}}, \quad J_i = (\varepsilon_i + \lambda_i + \kappa_i)/2. \quad (29)$$

Таким образом,

$$y(\tau|X) = \frac{1}{\tau} \sum_{i=1}^n h_{1i} \left(x_i \left(\frac{\varepsilon_i}{e^{\varepsilon_i \tau} - 1} + J_i \right)^{-1} + a_i \kappa_i \frac{J_i \tau - \ln[1 + (e^{\varepsilon_i \tau} - 1)J_i / \varepsilon_i]}{J_i(J_i - \varepsilon_i)} \right). \quad (30)$$

Форвардная кривая $f(\tau|X)$ определяется по формуле (17) следующим выражением:

$$f(\tau|X) = \sum_{i=1}^n x_i (h_{1i} - (\kappa_i + \lambda_i) B_i(\tau) - B_i^2(\tau)/2) + \sum_{i=1}^n a_i \kappa_i B_i(\tau), \quad (31)$$

где функции $B_i(\tau)$ вычисляются по формулам (29).

Кривые $y(\tau|X)$ и $f(\tau|X)$ обладают следующими предельными свойствами:

$$\lim_{\tau \rightarrow 0} y(\tau|X) = \lim_{\tau \rightarrow 0} f(\tau|X) = \sum_{i=1}^n h_{1i} x_i = r, \quad (32)$$

$$\lim_{\tau \rightarrow \infty} y(\tau|X) = \lim_{\tau \rightarrow \infty} f(\tau|X) = \sum_{i=1}^n \frac{a_i \kappa_i h_{1i}}{J_i}. \quad (33)$$

Таким образом, после того как определены кривые доходности $y(\tau|X)$ и $f(\tau|X)$ в пространстве переменных X , кривые доходности $y(\tau|Z)$ и $f(\tau|Z)$ в пространстве переменных Z определяются с помощью соотношений (26) по формулам

$$y(\tau|Z) = Z^T (H^T)^{-1} B(\tau)/\tau - A(\tau)/\tau, \quad (34)$$

$$f(\tau|Z) = Z^T (H^{-1})^T \frac{dB(\tau)}{d\tau} - \frac{dA(\tau)}{d\tau}, \quad (35)$$

где функции $A(\tau)$ и $B(\tau)$ определяются соотношениями (27)–(29).

Таким образом, если наблюдаемые (или оцениваемые) переменные состояния рынка Z имеют динамику, описываемую уравнением (25), то уравнение временной структуры бескупонных доходностей будет иметь вид

$$-\frac{dA(\tau)}{d\tau} + Z^T \frac{dB(\tau)}{d\tau} + (Z^T - \theta^T) K^T B(\tau) + \frac{1}{2} B(\tau)^T H D(H^{-1}Z) H^T B(\tau) -$$

$$- Z^T H_1^T = -\lambda^T D(H^{-1}Z) B(\tau),$$

где $D(H^{-1}Z)$ – диагональная матрица, на главной диагонали которой располагаются элементы вектора $H^{-1}Z$; H_1 – первая строка матрицы H . Считая компоненты вектора Z независимыми переменными, из этого равенства можно получить следующую систему уравнений для функций временной структуры $a(\tau)$ и $b(\tau)$:

$$a'(\tau) = -(K\theta)^T b(\tau), \quad a(0) = 0, \quad (36)$$

$$b_i'(\tau) = h_{1i} - \xi_i b(\tau) - b^T(\tau) \zeta_i b(\tau)/2, \quad b_i(0) = 0, \quad 1 \leq i \leq n, \quad (37)$$

где $\xi_i = (K_i - D(G_i)\lambda)^T$, а K_i и G_i – i -е столбцы матриц K и H^{-1} соответственно; $\zeta_i = HD(G_i)H^T$, $D(G_i)$ – по-прежнему диагональная матрица, на главной диагонали которой располагаются элементы вектора G_i . Никаких прямых методов аналитического решения системы уравнений Риккати (37) неизвестно. Однако, как следует из вышеприведенного анализа, переход к переменным состояния $X = H^{-1}Z$ позволяет аналитически решить задачу определения кривых доходности $y(\tau|X)$ и $f(\tau|X)$ в виде (30) и (31), а затем выразить кривые доходности $y(\tau|Z)$ и $f(\tau|Z)$ в аналитическом виде по формулам (34), (35).

3. Числовой пример

В заключение рассмотрим числовой пример. Для простоты будем решать задачу в риск-нейтральной постановке, когда рыночные цены риска $\lambda = 0$. Пусть наблюдение за процессами на финансовом рынке позволяет сформировать для описания динамики наблюдаемых переменных состояния $Z = (z_1, z_2, z_3)^T$, для которых $z_1 = r$ – краткосрочная процентная ставка, следующую систему стохастических дифференциальных уравнений типа (6):

$$dZ = \begin{pmatrix} 4 & -1 & -2 \\ 2 & 1 & -2 \\ 1 & -1 & 1 \end{pmatrix} \begin{pmatrix} 4 - z_1 \\ 2 - z_2 \\ 3 - z_3 \end{pmatrix} dt + \begin{pmatrix} \sqrt{z_1 - z_3} & \sqrt{z_1 - z_2} & \sqrt{z_3 + z_2 - z_1} \\ \sqrt{z_1 - z_3} & 0 & \sqrt{z_3 + z_2 - z_1} \\ 0 & \sqrt{z_1 - z_2} & \sqrt{z_3 + z_2 - z_1} \end{pmatrix} dW. \quad (38)$$

Для составления уравнений относительно функций временной структуры $a(\tau)$ и $b(\tau)$ выпишем необходимые элементы системы:

$$K\theta = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}, \quad K_1 = \begin{pmatrix} 4 \\ 2 \\ 1 \end{pmatrix}, \quad K_2 = \begin{pmatrix} -1 \\ 1 \\ -1 \end{pmatrix}, \quad K_3 = \begin{pmatrix} -2 \\ -2 \\ 1 \end{pmatrix},$$

$$\sigma(Z) \sigma(Z)^T = \begin{pmatrix} z_1 & z_2 & z_3 \\ z_2 & z_2 & z_3 + z_2 - z_1 \\ z_3 & z_3 + z_2 - z_1 & z_3 \end{pmatrix},$$

$$\beta_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix}, \quad \beta_2 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad \beta_3 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

Подставив эти выражения в систему уравнений (4), (5) из [2], получим уравнения для $a(\tau)$ и $b(\tau)$:

$$\begin{aligned} a'(\tau) &= -8b_1(\tau) - 4b_2(\tau) - 5b_3(\tau), \quad a(0) = 0, \\ b_1'(\tau) &= 1 - 4b_1(\tau) - 2b_2(\tau) - b_3(\tau) - 0,5b_1(\tau)^2 + b_2(\tau)b_3(\tau), \quad b_1(0) = 0, \\ b_2'(\tau) &= b_1(\tau) - b_2(\tau) + b_3(\tau) - 0,5b_2(\tau)^2 - b_1(\tau)b_2(\tau) - b_2(\tau)b_3(\tau), \quad b_2(0) = 0, \\ b_3'(\tau) &= 2b_1(\tau) + 2b_2(\tau) - b_3(\tau) + 0,5b_3(\tau)^2 - b_1(\tau)b_3(\tau) - b_2(\tau)b_3(\tau), \quad b_3(0) = 0. \end{aligned} \quad (39)$$

Система дифференциальных уравнений для компонент векторной функции $b(\tau)$ нелинейна и методы получения ее решения в аналитическом виде неизвестны, хотя численное ее решение можно найти просто, используя, например, программу Wolfram Mathematica.

Для получения аналитического решения, следуя представленной выше идее описания динамики рыночных процессов в другом пространстве переменных состояния, рассмотрим следующее линейное преобразование:

$$Z = HX = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} x_1 + x_2 + x_3 \\ x_1 + x_2 \\ x_2 + x_3 \end{pmatrix},$$

тогда

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & -1 \\ 1 & -1 & 0 \\ -1 & 1 & 1 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} z_1 - z_3 \\ z_1 - z_2 \\ z_3 + z_2 - z_1 \end{pmatrix}. \quad (40)$$

Экономический смысл переменных X мы не определяем, а пользуемся этим преобразованием только для получения аналитических решений для кривых доходностей, в итоговых выражениях для кривых доходностей, в которых сами переменные X не используются. При таком преобразовании согласно стохастическому анализу Ито функция дрейфа в уравнении (38) для переменных состояния X приобретает вид

$$\begin{pmatrix} 1 & 0 & -1 \\ 1 & -1 & 0 \\ -1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 4 & -1 & -2 \\ 2 & 1 & -2 \\ 1 & -1 & 1 \end{pmatrix} \begin{pmatrix} 4 - z_1 \\ 2 - z_2 \\ 3 - z_3 \end{pmatrix} = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 - x_1 \\ 2 - x_2 \\ 1 - x_3 \end{pmatrix}.$$

Заметим, что матрицу волатильности в уравнении (38) можно записать как

$$\begin{pmatrix} \sqrt{z_1 - z_3} & \sqrt{z_1 - z_2} & \sqrt{z_3 + z_2 - z_1} \\ \sqrt{z_1 - z_3} & 0 & \sqrt{z_3 + z_2 - z_1} \\ 0 & \sqrt{z_1 - z_2} & \sqrt{z_3 + z_2 - z_1} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} \sqrt{z_1 - z_3} & 0 & 0 \\ 0 & \sqrt{z_1 - z_2} & 0 \\ 0 & 0 & \sqrt{z_3 + z_2 - z_1} \end{pmatrix}.$$

Значит, с учетом соотношений (40) при преобразовании переменных состояния к X матрица волатильности преобразуется к диагональному виду

$$\begin{pmatrix} 1 & 0 & -1 \\ 1 & -1 & 0 \\ -1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \sqrt{z_1 - z_3} & \sqrt{z_1 - z_2} & \sqrt{z_3 + z_2 - z_1} \\ \sqrt{z_1 - z_3} & 0 & \sqrt{z_3 + z_2 - z_1} \\ 0 & \sqrt{z_1 - z_2} & \sqrt{z_3 + z_2 - z_1} \end{pmatrix} = \begin{pmatrix} \sqrt{x_1} & 0 & 0 \\ 0 & \sqrt{x_2} & 0 \\ 0 & 0 & \sqrt{x_3} \end{pmatrix}.$$

Следовательно, в пространстве состояний X уравнение динамики рыночных процессов (38) приобретает вид трехфакторной модели Лонгстаффа–Шварца (1):

$$dX = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 - x_1 \\ 2 - x_2 \\ 1 - x_3 \end{pmatrix} dt + \begin{pmatrix} \sqrt{x_1} dW_1 \\ \sqrt{x_2} dW_2 \\ \sqrt{x_3} dW_3 \end{pmatrix}. \quad (41)$$

Это означает, что система (38) превращается в набор независимых уравнений

$$\begin{aligned} dx_1 &= 3(1 - x_1)dt + \sqrt{x_1} dW_1, \\ dx_2 &= 2(2 - x_2)dt + \sqrt{x_2} dW_2, \\ dx_3 &= (1 - x_3)dt + \sqrt{x_3} dW_3. \end{aligned}$$

С учетом того, что $z_1 = x_1 + x_2 + x_3 = r$, а также используя свойства системы (41) для построения уравнений (4), (5) из [2] для функций временной структуры $A(\tau)$ и $B(\tau) = (B_1(\tau) \ B_2(\tau) \ B_3(\tau))^T$ в пространстве переменных состояния X , получим также набор независимых уравнений для функций $B(\tau)$ типа (11), (12):

$$\begin{aligned} A'(\tau) &= -3B_1(\tau) - 4B_2(\tau) - B_3(\tau), \quad A(0) = 0, \\ B_1'(\tau) &= 1 - 3B_1(\tau) - B_1(\tau)^2/2, \quad B_1(0) = 0, \\ B_2'(\tau) &= 1 - 2B_2(\tau) - B_2(\tau)^2/2, \quad B_2(0) = 0, \\ B_3'(\tau) &= 1 - B_3(\tau) - B_3(\tau)^2/2, \quad B_3(0) = 0, \end{aligned}$$

которые имеют аналитические решения типа (13), (14):

$$B_1(\tau) = \left(\frac{\sqrt{11}}{e^{\tau\sqrt{11}} - 1} + \frac{3 + \sqrt{11}}{2} \right)^{-1}, \quad B_2(\tau) = \left(\frac{\sqrt{6}}{e^{\tau\sqrt{6}} - 1} + \frac{2 + \sqrt{6}}{2} \right)^{-1}, \quad B_3(\tau) = \left(\frac{\sqrt{3}}{e^{\tau\sqrt{3}} - 1} + \frac{1 + \sqrt{3}}{2} \right)^{-1}. \quad (42)$$

Наконец, на основании равенств (26) получаем выражения для $a'(\tau)$, $b_1(\tau)$, $b_2(\tau)$ и $b_3(\tau)$, которые являются аналитическими решениями нелинейных уравнений (36), (37):

$$\begin{aligned} a'(\tau) &= -3B_1(\tau) - 4B_2(\tau) - B_3(\tau), \\ b_1(\tau) &= B_1(\tau) + B_2(\tau) - B_3(\tau), \end{aligned}$$

$$b_2(\tau) = -B_2(\tau) + B_3(\tau),$$

$$b_3(\tau) = -B_1(\tau) + B_3(\tau),$$

где $B_1(\tau)$, $B_2(\tau)$ и $B_3(\tau)$ вычисляются по формулам (41). Справедливость этих решений проверяется просто подстановкой таким образом найденных $b_1(\tau)$, $b_2(\tau)$ и $b_3(\tau)$ в уравнения (39). После определения функций временной структуры по формулам (34) и (35) можно построить кривые доходности до погашения $y(\tau|Z)$ и форвардные кривые $f(\tau|Z)$. Аналитические выражения $y(\tau|Z)$ и $f(\tau|Z)$ здесь не выписываются ввиду их громоздкости. На рисунке 2 они представлены в виде графиков для стартовых значений компонент вектора Z : $z_1 = r = 10$ (в %), $z_2 = 0$, $z_3 = 0$. Для графиков рис. 2 переменная срока до погашения преобразована так же, как и на рис. 1, только значение параметра ρ здесь равно 0,2303.

Заметим, что согласно свойствам (32), (33) кривые $Y(u|Z)$ и $F(u|Z)$ стартуют из общей точки $Y(0|Z) = F(0|Z) = r = 10$ и при $u \rightarrow 1$ стремятся к общему значению

$$Y(1|Z) = F(1|Z) = \frac{3}{J_1} + \frac{4}{J_2} + \frac{1}{J_3} = \frac{6}{3 + \sqrt{11}} + \frac{8}{2 + \sqrt{6}} + \frac{2}{1 + \sqrt{3}} = 3,4799.$$

Форвардная кривая $F(u|Z)$ в окрестности стартовой точки изменяется быстрее, чем кривая доходности до погашения $Y(u|Z)$ (по теории ее производная по абсолютной величине в два раза больше производной кривой доходности до погашения). Кривые имеют минимумы, причем кривая доходности до погашения $Y(u|Z)$ имеет минимум в точке пересечения с кривой $F(u|Z)$, что соответствует теории.

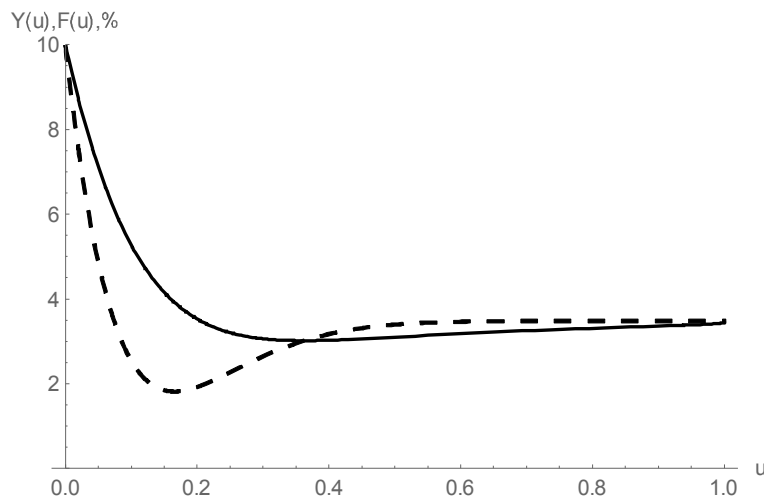


Рис. 2. Кривая доходности до погашения $Y(u)$ и кривая форвардных ставок $F(u)$ для случая, когда динамика переменных состояния определяется уравнением (38)

Заключение

В статье на основе анализа модели Лонгстаффа–Шварцца предложена ее расширенная версия для произвольного числа переменных состояния, которую можно использовать для получения аналитического решения уравнений относительно функций временной структуры доходности, что позволяет получать аналитические выражения для кривых доходности до погашения и форвардных кривых процентных ставок. При этом использован очевидный принцип, заключающийся в том, что если задан процесс краткосрочной процентной ставки, то выражения для кривых доходности, соответствующих этому процессу, не зависят от того, каким способом описывается пространство переменных состояния финансового рынка. Показано, что применение этого принципа в расширении модели Лонгстаффа–Шварцца позволяет получить аналитические решения системы нелинейных уравнений Риккати произвольного порядка, аналитические методы решения которых в литературе не описаны. Предложенное расширение модели Лонгстаффа–Шварцца свободно от недостатка, присущего этой модели и заключающегося в том, что при переходе от латентных переменных к реальным наблюдаемым переменным ограничивается область возможных значений реальных переменных состояний. В расширении такого эффекта нет.

ЛИТЕРАТУРА

1. Longstaff F.A., Schwartz E.S. Interest Rate Volatility and the Term Structure: A Two-Factor General Equilibrium Model // *Journal of Finance*. 1992. V. 47, No. 4. P. 1259–1282.
2. Медведев Г.А. О временной структуре доходности. 1. Модель Васичека // *Вестник Томского государственного университета. Управление, вычислительная техника и информатика*. 2012. № 1 (18). С. 102–111.

Медведев Геннадий Алексеевич, д-р. физ.-мат. наук, профессор. E-mail: MedvedevGA@bsu.by
Белорусский государственный университет (г. Минск, Беларусь)

Поступила в редакцию 21 марта 2017 г.

Medvedev Gennady A. (Belarusian State University, Belarus).

The Longstaff-Schwartz model of yield term structure and its expansion.

Keywords: yield term structure; short-term interest rate; Longstaff-Schwartz model; system of Riccati equations.

DOI: 10.17223/19988605/40/5

The Longstaff-Schwartz model is considered both in the space of latent state variables and in the space of observable (or estimated) state variables. Analytical expressions for yield curves to maturity and forward curves are obtained in both cases. Based on the analysis of the Longstaff-Schwartz model, an extended version is proposed for an arbitrary number of state variables. Within the framework of this expansion, a method is proposed for obtaining analytical solutions of equations with respect to the functions of the yield term structure, when the initial equations for the dynamics of the short-term interest rate lead to non-linear systems of Riccati equations with respect to these functions, which do not allow an analytical solution to be obtained by known methods. This allows you to obtain analytical expressions for yield curves to maturity and forward interest rate curves. At the same time, the obvious principle is used: if the process of a short-term interest rate is specified, then the expressions for yield curves corresponding to this process do not depend on how the space of state variables of the financial market is described. It is shown that the application of this principle to the extension of the Longstaff-Schwartz model allows one to obtain analytical solutions of a system of nonlinear Riccati equations of arbitrary order, whose analytical methods are not described in the literature. The proposed extension of the Longstaff-Schwartz model is free from the drawback inherent in this model, which consists in the fact that in the transition from latent variables to real observable variables, the domain of definition of these real state variables is limited. In the expansion there is no such effect. Numerical examples are given.

REFERENCES

1. Longstaff, F.A. & Schwartz, E.S. (1992) Interest Rate Volatility and the Term Structure: A Two-Factor General Equilibrium Model. *Journal of Finance*. 47(4). pp. 1259–1282. DOI: 10.1111/j.1540-6261.1992.tb04657.x
2. Medvedev, G.A. (2012) On term structure of yield rates. 1. Vasiček model. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science*. 1(18). pp. 102–111. (In Russian).

А.Н. Тырсин, К.К. Костин

ОЦЕНИВАНИЕ ЛОГИСТИЧЕСКОЙ РЕГРЕССИИ КАК ЭКСТРЕМАЛЬНАЯ ЗАДАЧА

Работа выполнена при финансовой поддержке РФФИ № 16-06-00048.

Рассматриваются особенности оценивания логистической регрессии как экстремальной задачи, реализующей критерий максимального правдоподобия. Показано, что эта задача имеет существенные отличия по сравнению с поиском локального экстремума. Описан устойчивый алгоритм вычисления коэффициентов разделяющего уравнения логистической регрессии. Приведены результаты статистического моделирования сравнительного анализа оценок логистической регрессии предложенного алгоритма с известными алгоритмами.

Ключевые слова: логистическая регрессия; метод максимального правдоподобия; алгоритм спуска; случайный поиск; модель; целевая функция.

Логистическая регрессия является одним из распространенных методов классификации данных в разных областях [1. С. 181–242; 2. С. 283–369; 3. С. 152–214; 4. С. 321–329].

Основная цель логистической регрессии состоит в разделении множества исходных значений линейной границей (разделяющей прямой, плоскостью или гиперплоскостью) на две соответствующих двум заданным классам области. Логистическая регрессия прогнозирует вероятность некоторого события, находящуюся в пределах от 0 до 1 [4. С. 322–324]. Следует указать, что число классов L может быть больше двух, в этом случае они имеют мультиномиальную регрессию, которую можно построить с помощью $L - 1$ независимых логистических регрессий. Поэтому будем рассматривать задачу классификации с двумя классами.

В настоящее время построение логистической регрессии обычно осуществляют в виде оптимизационной задачи, реализующей критерий максимального правдоподобия [5. С. 124]. Ряд авторов [6. С. 31–32; 7] предлагают оценивать коэффициенты разделяющего уравнения методом Ньютона–Рафсона. Однако исследования на модельных данных этого и других алгоритмов оценивания логистической регрессии, реализованных в статистических пакетах STATISTICA и SPSS, показали неустойчивость оценок. Этот недостаток отмечается также в [7].

Цель статьи – исследование особенностей процедуры оценивания логистической регрессии как экстремальной задачи, а также разработка устойчивого алгоритма вычисления коэффициентов разделяющего уравнения.

1. Построение логистической регрессии по критерию максимального правдоподобия

Пусть имеется выборка прецедентов (обучающая выборка):

$$(\mathbf{x}_i, y_i), i = 1, 2, \dots, n, \quad (1)$$

$$\text{где } \mathbf{x}_i = \begin{pmatrix} x_{i1} \\ x_{i2} \\ \dots \\ x_{im} \end{pmatrix} = \begin{pmatrix} 1 \\ x_{i2} \\ \dots \\ x_{im} \end{pmatrix} - \text{вектор значений } i\text{-го объекта, } \forall i \ x_{i1} = 1; \mathbf{X} = \begin{pmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \dots \\ \mathbf{x}_n^T \end{pmatrix} = \begin{pmatrix} 1 & x_{12} & \dots & x_{1m} \\ 1 & x_{22} & \dots & x_{2m} \\ \dots & \dots & \dots & \dots \\ 1 & x_{n2} & \dots & x_{nm} \end{pmatrix};$$

$y_i \in \{-1; 1\}$ – бинарная переменная, указывающая на принадлежность i -го объекта соответствующему классу, например, для определенности первому классу при $y_i = -1$ и второму – при $y_i = 1$; $M = m - 1$ – число признаков у каждого объекта; n – количество наблюдений.

Классификацию осуществляют с помощью функции $h(\mathbf{x}) = \frac{1}{1 + \exp\{-\mathbf{b}^T \mathbf{x}\}}$, принимающей значения в интервале (0; 1). Пороговым значением является $h(\mathbf{x}) = 0,5$. В [6. С. 29–32] показано, что максимизация логарифма правдоподобия эквивалентна минимизации функционала

$$Q(\mathbf{b}) = \sum_{i=1}^n \ln(1 + e^{-y_i \mathbf{b}^T \mathbf{x}_i}) \rightarrow \min_{\mathbf{b} \in \mathbf{R}^m}, \quad (2)$$

где $\mathbf{b}^T = (b_1 \ b_2 \ \dots \ b_m)$ – искомый вектор коэффициентов разделяющей линейной границы, описываемой в общем случае уравнением гиперплоскости

$$b_1 + \sum_{j=2}^m b_j x_j = 0.$$

Аналитически задача минимизации (2) не решается. Поэтому для оценивания вектора коэффициентов $\mathbf{b}$ используют итерационные алгоритмы спуска для решения экстремальных задач. В [6. С. 32–34] описан итерационный алгоритм Ньютона–Рафсона решения задачи (2). Он состоит в следующем.

В качестве нулевого приближения можно взять решение задачи классификации методом многомерной линейной регрессии

$$\mathbf{b}^{(0)} = (\mathbf{X}^T \mathbf{X})^{-1} (\mathbf{X}^T \mathbf{Y}). \quad (3)$$

Затем начинается итерационный процесс, на k -м шаге которого уточняется вектор коэффициентов $\mathbf{b}^{(k)}$:

$$\mathbf{b}^{(k)} = \mathbf{b}^{(k-1)} - h_k (\mathbf{Q}''(\mathbf{b}^{(k-1)}))^{-1} \mathbf{Q}'(\mathbf{b}^{(k-1)}),$$

где $\mathbf{Q}'(\mathbf{b}^{(k)})$ – вектор первых производных (градиент) функционала $Q(\mathbf{b})$ в точке $\mathbf{b}^{(k)}$; $\mathbf{Q}''(\mathbf{b}^{(k)})$ – матрица вторых производных (гессиан) функционала $Q(\mathbf{b})$ в точке $\mathbf{b}^{(k)}$; h_k – величина шага, который можно положить равным 1, но его подбор на каждом шаге способен увеличить скорость сходимости. Метод Ньютона–Рафсона описан в ряде учебников по методам оптимизации, например в [8. С. 223–226].

Статистическое моделирование методом Монте–Карло [9] показало неустойчивость в некоторых случаях работы этого алгоритма, а также других алгоритмов, реализованных в статистических пакетах. Отметим основные причины этого.

В итерационных алгоритмах поиска минимума требуется задать начальное приближение вектора $\mathbf{b}^{(0)}$. Плохое начальное приближение, значительно отличающееся от точки локального минимума, может значительно ухудшить условия сходимости итерационного алгоритма. В результате статистического моделирования выяснилось, что использование в качестве начального приближения $\mathbf{b}^{(0)}$ решения задачи классификации в виде линейной регрессии часто дает неудовлетворительный результат. Покажем это на примере.

Пример 1. Сгенерируем выборку прецедентов $(\mathbf{x}_i, y_i)$, $\mathbf{x}_i = \begin{pmatrix} x_{i1} \\ x_{i2} \\ x_{i3} \end{pmatrix} = \begin{pmatrix} 1 \\ x_{i2} \\ x_{i3} \end{pmatrix}$, $i = 1, 2, \dots, n$, следующим

образом. Объем выборки $n = 100$, число признаков $M = 2$, первая половина выборки ($i = 1, 2, \dots, 50$) относится к первому классу ($y_i = -1$), а вторая – ко второму классу ($y_i = 1$). Пусть в каждом классе признаки являются взаимно независимыми гауссовскими случайными величинами. Это означает, что первый и второй классы представляют собой случайные выборки из двумерных гауссовских случайных векторов $\mathbf{X}^{(1)} = (X_2^{(1)}, X_3^{(1)})$ и $\mathbf{X}^{(2)} = (X_2^{(2)}, X_3^{(2)})$ соответственно. Зададим у всех компонент одно и то же среднее квадратическое отклонение $\sigma_X = 1,5$ и математические ожидания: $M[X_2^{(1)}] = 10$, $M[X_3^{(1)}] = 16$, $M[X_2^{(2)}] = 4$, $M[X_3^{(2)}] = 4$. На рисунке 1 приведен результат классификации с помощью метода наименьших квадратов на основе (3). Разделяющая прямая не только не классифицирует выборку прецедентов на два класса, но и значительно удалена от всех объектов. Хотя классы в этом примере легкоразделимы. Такое неудачное начальное приближение значительно затрудняет работу итерационных алгоритмов решения задачи (2).

Вторая причина неустойчивости решений задачи (2) состоит в том, что итерационные алгоритмы спуска рассчитаны на поиск локального минимума. А задача (2) при корректной классификации всех прецедентов имеет нижнюю грань на бесконечности, равную нулю. Причем качество построения разделяющей границы при формальном спуске уже не будет связано со значением целевой функции $Q(\mathbf{b})$. Поэтому требуется корректировка стратегии спуска.

Третья причина состоит в том, что при достаточно малых значениях целевой функции $Q(\mathbf{b})$ некоторые величины $-y_i \mathbf{b}^T \mathbf{x}_i$ могут принимать очень большие значения, что приводит к неустрашимым вычислительным погрешностям и даже к останову работы алгоритма из-за переполнения памяти.

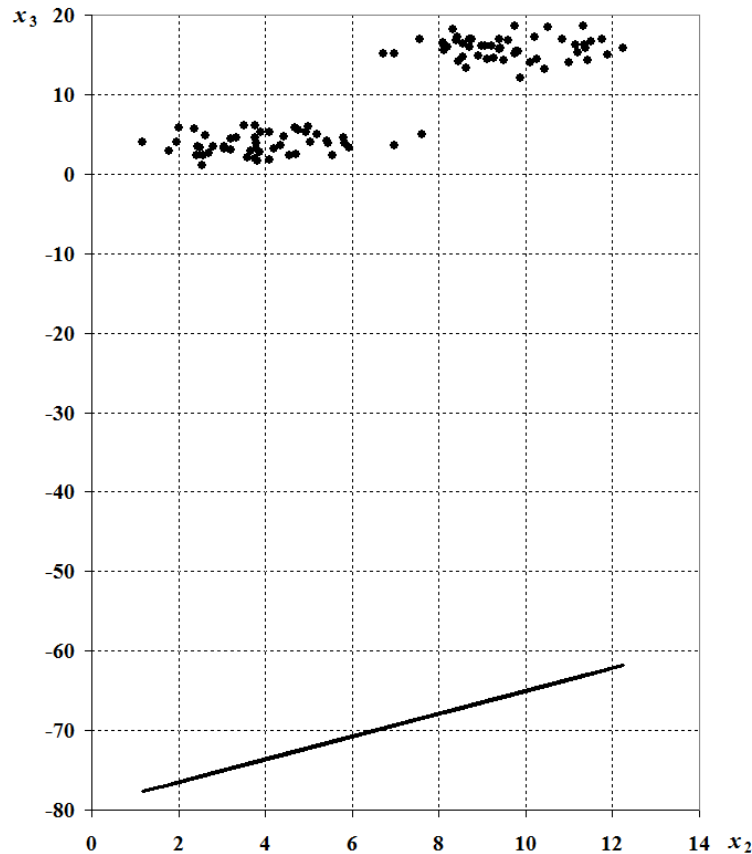


Рис. 1. Пример построения разделяющей границы с помощью линейной регрессии

Таким образом, приведенные недостатки непосредственного использования итерационных алгоритмов спуска требуют исследования особенностей задачи (2).

2. Исследование целевой функции задачи оценивания коэффициентов разделяющей границы по критерию максимального правдоподобия

Исследуем свойства целевой функции $Q(\mathbf{b})$. Рассмотрим вначале произвольное i -е слагаемое целевой функции $Q(\mathbf{b})$, т.е. функцию

$$z_i(\mathbf{b}) = \ln(1 + e^{-y_i \mathbf{b}^T \mathbf{x}_i}) = \ln \left[1 + \exp \left\{ -y_i \sum_{k=1}^m b_k x_{ik} \right\} \right].$$

Найдем ее частные производные:

$$\frac{\partial z_i}{\partial b_k} = -\frac{y_i x_{ik} e^{-y_i \mathbf{b}^T \mathbf{x}_i}}{1 + e^{-y_i \mathbf{b}^T \mathbf{x}_i}}, \quad k = 1, 2, \dots, m.$$

Так как все частные производные непрерывно дифференцируемы и $\frac{\partial z_i}{\partial b_1} \neq 0$, то функция $z_i(\mathbf{b})$ не имеет стационарных точек, а значит, у нее нет экстремумов.

Далее найдем вторые производные $\frac{\partial^2 z_i}{\partial b_k \partial b_j}$. Они равны

$$\frac{\partial^2 z_i}{\partial b_k \partial b_j} = y_i^2 x_{ik} x_{ij} U_i(\mathbf{b}), \quad k, j = 1, 2, \dots, m,$$

где $U_i(\mathbf{b}) = \frac{e^{-y_i \mathbf{b}^T \mathbf{x}_i}}{(1 + e^{-y_i \mathbf{b}^T \mathbf{x}_i})^2} > 0$.

Поскольку все вторые производные всюду непрерывны, то $\frac{\partial^2 z_i}{\partial b_k \partial b_j} = \frac{\partial^2 z_i}{\partial b_j \partial b_k}$. Исследуем функцию $z_i(\mathbf{b})$ на выпуклость. Ее гессиан равен

$$H(z_i) = (q_{kj}(\mathbf{b})), \quad q_{kj}(\mathbf{b}) = \frac{\partial^2 z_i}{\partial b_k \partial b_j} = y_i^2 x_{ik} x_{ij} U_i(\mathbf{b}), \quad k, j = 1, 2, \dots, m.$$

Воспользуемся критерием Сильвестра [10. С. 507]. Определим главные миноры $A_1, A_2, \dots, A_m$ гессиана $H(z_i)$. Поскольку $x_{i1} = 1$, $y_i^2 = 1$, то $A_1 = y_i^2 x_{i1}^2 U_i(\mathbf{b}) = U_i(\mathbf{b}) > 0$. Можно показать, что все остальные главные миноры

$$A_k = (U_i(\mathbf{b}))^k \cdot \begin{vmatrix} 1 & x_{i1}x_{i2} & \dots & x_{i1}x_{ik} \\ x_{i2}x_{i1} & x_{i2}^2 & \dots & x_{i2}x_{ik} \\ \dots & \dots & \dots & \dots \\ x_{ik}x_{i1} & x_{ik}x_{i2} & \dots & x_{ik}^2 \end{vmatrix} = 0, \quad k = 2, 3, \dots, m.$$

Получили, что квадратичная форма относительно переменных $h_1, h_2, \dots, h_m$

$$\Phi(h_1, h_2, \dots, h_m) = \sum_{k=1}^m \sum_{j=1}^m q_{kj}(\mathbf{b}) h_k h_j$$

является неотрицательно определенной. Это означает выпуклость функции $z_i(\mathbf{b})$ на $\mathbf{R}^m$.

Перейдем теперь к рассмотрению функции $Q(\mathbf{b})$. Она является суммой n собственных выпуклых функций $z_i(\mathbf{b})$, следовательно, $Q(\mathbf{b})$ также выпукла [11. С. 49].

Найдем частные производные функции $Q(\mathbf{b})$:

$$\frac{\partial Q}{\partial b_k} = - \sum_{i=1}^n \frac{y_i x_{ik} e^{-y_i \mathbf{b}^T \mathbf{x}_i}}{1 + e^{-y_i \mathbf{b}^T \mathbf{x}_i}}, \quad k = 1, 2, \dots, m.$$

Все частные производные всюду на $\mathbf{R}^m$ непрерывно дифференцируемы, поэтому экстремумы функции $Q(\mathbf{b})$ могут быть только в стационарных точках, в которых градиент равен нулю, т.е.

$$\sum_{i=1}^n \frac{y_i x_{ik} e^{-y_i \mathbf{b}^T \mathbf{x}_i}}{1 + e^{-y_i \mathbf{b}^T \mathbf{x}_i}} = 0, \quad k = 1, 2, \dots, m. \quad (4)$$

Система нелинейных уравнений (4) имеет достаточно сложный вид, установить ее несовместность или найти решения аналитически не представляется возможным. Однако удалось подобрать примеры, когда система (4) имеет решения. Это означает, что целевая функция $Q(\mathbf{b})$ может иметь стационарные точки, а значит, ввиду ее выпуклости может иметь локальные минимумы, не являющиеся глобальным минимумом.

Найдем вторые производные $\frac{\partial^2 Q}{\partial b_k \partial b_j}$. С учетом того, что $\forall i \quad y_i^2 = 1$, они равны

$$\frac{\partial^2 z_i}{\partial b_k \partial b_j} = \sum_{i=1}^n x_{ik} x_{ij} U_i(\mathbf{b}), \quad k, j = 1, 2, \dots, m.$$

Так как все вторые производные всюду непрерывны, то $\frac{\partial^2 Q}{\partial b_k \partial b_j} = \frac{\partial^2 Q}{\partial b_j \partial b_k}$. Гессиан функции $Q(\mathbf{b})$ равен

$$H(Q) = (Q_{kj}(\mathbf{b})), \quad Q_{kj}(\mathbf{b}) = \frac{\partial^2 z_i}{\partial b_k \partial b_j} = \sum_{i=1}^n x_{ik} x_{ij} U_i(\mathbf{b}), \quad k, j = 1, 2, \dots, m.$$

Опять воспользуемся критерием Сильвестра и определим главные миноры $A_1, A_2, \dots, A_m$ гессиана $H(Q)$. Поскольку $x_{i1} = 1$, то $A_1 = \sum_{i=1}^n x_{i1}^2 U_i(\mathbf{b}) = \sum_{i=1}^n U_i(\mathbf{b}) > 0$ как сумма строго положительных чисел. Остальные главные миноры с учетом выпуклости $Q(\mathbf{b})$ равны

$$A_k = \begin{vmatrix} \sum_{i=1}^n U_i(\mathbf{b}) & \sum_{i=1}^n x_{i2} U_i(\mathbf{b}) & \dots & \sum_{i=1}^n x_{ik} U_i(\mathbf{b}) \\ \sum_{i=1}^n x_{i2} U_i(\mathbf{b}) & \sum_{i=1}^n x_{i2}^2 U_i(\mathbf{b}) & \dots & \sum_{i=1}^n x_{i2} x_{ik} U_i(\mathbf{b}) \\ \dots & \dots & \dots & \dots \\ \sum_{i=1}^n x_{ik} U_i(\mathbf{b}) & \sum_{i=1}^n x_{ik} x_{i2} U_i(\mathbf{b}) & \dots & \sum_{i=1}^n x_{ik}^2 U_i(\mathbf{b}) \end{vmatrix} \geq 0, \quad k = 2, 3, \dots, m.$$

Рассмотрим частные случаи.

Пусть $m = 2, n = 2$. После преобразований получим $A_2 = (x_{12} - x_{22})^2 U_1(\mathbf{b}) U_2(\mathbf{b}) \geq 0$.

Пусть $m = 2, n = 3$. После преобразований получим $A_2 = \sum_{i=1}^{n-1} \sum_{j=i+1}^n (x_{i2} - x_{j2})^2 U_i(\mathbf{b}) U_j(\mathbf{b}) \geq 0$.

Таким образом, квадратичная форма в случае нескольких наблюдений может быть положительно определенной или неотрицательно определенной. Это означает, что функция $Q(\mathbf{b})$:

- является выпуклой на $\mathbf{R}^m$ и не имеет максимумов;
- может иметь стационарные точки, в которых могут быть локальные минимумы.

Поскольку равенство нулю главных миноров будет только в некоторых точках, например в вырожденных случаях равенства всех измерений, то функция $Q(\mathbf{b})$ может оказаться в окрестности стационарной точки строго выпуклой. Это означает, что в такой стационарной точке у функции $Q(\mathbf{b})$ будет локальный минимум. Из этого можно сделать выводы.

Использование численных методов спуска первого или второго порядков при наличии у целевой функции стационарных точек может привести к неправильному построению логистической регрессии по двум причинам. Во-первых, можно в качестве решения получить локальный минимум, который будет грубой оценкой коэффициентов уравнения логистической регрессии. Во-вторых, спуск может стремиться к нижней грани целевой функции, а вектор коэффициентов логистической регрессии будет оставаться грубым решением.

3. Устойчивый алгоритм вычисления коэффициентов разделяющего уравнения логистической регрессии

Имеем выборку прецедентов (1), в которой первые N_1 объектов относятся к первому классу ($y_i = -1$), а остальные – ко второму классу ($y_i = 1$).

С учетом вышеизложенного приведем описание алгоритма. Он состоит в следующем.

1. Найдем середину $\tilde{\mathbf{x}}^{(0)}$ отрезка, соединяющего центры тяжести первого и второго классов

$$\tilde{\mathbf{x}}^{(1)} = \frac{1}{N_1} \sum_{i=1}^{N_1} \mathbf{x}_i, \quad \tilde{\mathbf{x}}^{(2)} = \frac{1}{n - N_1} \sum_{i=N_1+1}^n \mathbf{x}_i.$$

2. Осуществляем параллельный сдвиг начала координат в точку $\tilde{\mathbf{x}}^{(0)}$.

3. Через точку $\tilde{\mathbf{x}}^{(0)}$ строим гиперплоскость $S: s_1 + s_2 x_2 + \dots + s_m x_m = 0$, ортогональную вектору $\mathbf{d} = \tilde{\mathbf{x}}^{(2)} - \tilde{\mathbf{x}}^{(1)}$.

4. Получаем начальную оценку $\mathbf{b}^{(0)}$ вектора разделяющей гиперплоскости как $\mathbf{b}^{(0)} = \mathbf{s}/|\mathbf{s}|$, где $\mathbf{s}$ – вектор коэффициентов гиперплоскости S . При этом длина $|\mathbf{b}^{(0)}| = 1$. Зададим параметр $\beta_0 = 1$.

5. С помощью итерационного алгоритма ищем с заданной точностью ε решение задачи (2). На любом k -м шаге алгоритма, многократно генерируя случайный вектор $\mathbf{p}^{(l)}$, решаем задачу

$$\begin{cases} Q(\mathbf{b}^{(k,l)}) \rightarrow \min_{\mathbf{b}^{(k)} \in \mathbf{R}^m}, \\ \mathbf{b}^{(k,l)} = \alpha(\mathbf{b}^{(k,l-1)} + h_k \mathbf{p}^{(l)}), \\ |\mathbf{b}^{(k,l)}| = \beta_{k-1}, \mathbf{b}^{(k,1)} = \mathbf{b}^{(k-1)}. \end{cases}$$

Если требуемая точность решения будет достигнута ($Q(\mathbf{b}^{(k,l)}) < \varepsilon$), то в качестве решения задачи используем последнюю оценку $\mathbf{b}^{(k,l)}$. В противном случае фиксируем последнюю оценку $\mathbf{b}^{(k)} = \mathbf{b}^{(k,l)}$, увеличиваем длину векторов оценок, т.е. $\beta_k = \gamma \cdot \beta_{k-1}$, где $\gamma > 1$ – коэффициент растяжения и возвращаемся к п. 5 ($k := k + 1$).

Следует отметить, что слишком малое значение параметра точности ε может привести к большим длинам оценок векторов $\mathbf{b}^{(k,l)}$ и, как указано выше, к потере точности алгоритма.

4. Вычислительные эксперименты

Проведем с помощью метода статистических испытаний Монте-Карло несколько экспериментов для исследования эффективности предложенного алгоритма по сравнению с алгоритмом «Quasi-Newton», реализованным в статистическом пакете STATISTICA. Он показал наилучшие результаты оценивания среди алгоритмов логистической регрессии в STATISTICA.

Число опытов $L = 200$. Выборка прецедентов имеет тот же вид, что и в примере 1, различия заключаются в объемах выборок первого N_1 и второго N_2 классов ($N_1 + N_2 = n$) и математических ожиданиях $M[X_2^{(1)}]$, $M[X_3^{(1)}]$, $M[X_2^{(2)}]$, $M[X_3^{(2)}]$. Поскольку с точностью до постоянного множителя векторов коэффициентов $\mathbf{b}$ может быть бесконечное множество, то для удобства сравнения результаты оценивания будем задавать в нормированном виде с единичной длиной.

Пример 2. Рассмотрим случай одинаковых объемов выборок в классах. Зададим $N_1 = N_2 = 20$. Результаты оценивания коэффициентов логистической регрессии по предложенному алгоритму и алгоритму «Quasi-Newton» в пакете STATISTICA приведены в табл. 1.

Т а б л и ц а 1

Результаты оценивания логистической регрессии по предложенному алгоритму и алгоритму «Quasi-Newton» в пакете STATISTICA для случая $N_1 = N_2 = 20$

Матожидания признаков	Параметр	Предложенный алгоритм			Алгоритм «Quasi-Newton»		
		b_1	b_2	b_3	b_1	b_2	b_3
$M[X_2^{(1)}] = 6$,	β_j	0,00000	–0,89443	–0,44721	0,00000	–0,89443	–0,44721
$M[X_3^{(1)}] = 3$,	$\bar{b}_j$	–0,00925	–0,88467	–0,46612	–0,13761	–0,78647	–0,28092
$M[X_2^{(2)}] = –6$,	$s(b_j)$	0,26070	0,10410	0,13367	0,47889	0,21434	0,29080
$M[X_3^{(2)}] = –3$	Δb_j	–0,00925	0,00975	–0,01891	–0,13761	0,10795	0,16629
$M[X_2^{(1)}] = 16$,	β_j	0,99723	–0,06648	–0,03324	0,99723	–0,06648	–0,03324
$M[X_3^{(1)}] = 13$,	$\bar{b}_j$	0,99721	–0,06691	–0,03297	0,99701	–0,04299	–0,05718
$M[X_2^{(2)}] = 4$,	$s(b_j)$	0,00047	0,00958	0,00923	0,00063	0,03115	0,03347
$M[X_3^{(2)}] = 7$	Δb_j	–0,00002	–0,00042	0,00027	–0,00023	0,02350	–0,02394

Матожидания признаков	Параметр	Предложенный алгоритм			Алгоритм «Quasi-Newton»		
		b_1	b_2	b_3	b_1	b_2	b_3
$M[X_2^{(1)}] = 26$,	β_j	0,99931	-0,03331	-0,01666	0,99931	-0,03331	-0,01666
$M[X_3^{(1)}] = 23$,	$\bar{b}_j$	0,99930	-0,03345	-0,01648	0,99917	-0,03187	-0,01801
$M[X_2^{(2)}] = 14$,	$s(b_j)$	0,00009	0,00428	0,00429	0,00060	0,01393	0,01250
$M[X_3^{(2)}] = 17$	Δb_j	0,00000	-0,00014	0,00018	-0,00014	0,00144	-0,00135

Пример 3. Рассмотрим случай разных объемов выборок в классах. Зададим $N_1 = 20$, $N_2 = 40$. Результаты оценивания коэффициентов логистической регрессии по предложенному алгоритму и алгоритму «Quasi-Newton» в пакете STATISTICA приведены в табл. 2.

Таблица 2

Результаты оценивания логистической регрессии по предложенному алгоритму и алгоритму «Quasi-Newton» в пакете STATISTICA для случая $N_1 = 20$, $N_2 = 40$

Матожидания признаков	Параметр	Предложенный алгоритм			Алгоритм «Quasi-Newton»		
		b_1	b_2	b_3	b_1	b_2	b_3
$M[X_2^{(1)}] = 6$,	β_j	0,00000	-0,89443	-0,44721	0,00000	-0,89443	-0,44721
$M[X_3^{(1)}] = 3$,	$\bar{b}_j$	0,09740	-0,89644	-0,43234	-0,10119	-0,73443	-0,18157
$M[X_2^{(2)}] = -6$,	$s(b_j)$	0,25587	0,08658	0,14076	0,55009	0,27014	0,41928
$M[X_3^{(2)}] = -3$	Δb_j	0,09740	-0,00201	0,01488	-0,10119	0,15999	0,26565
$M[X_2^{(1)}] = 16$,	β_j	0,99723	-0,06648	-0,03324	0,99723	-0,06648	-0,03324
$M[X_3^{(1)}] = 13$,	$\bar{b}_j$	0,99724	-0,06699	-0,03194	0,99702	-0,03436	-0,06328
$M[X_2^{(2)}] = 4$,	$s(b_j)$	0,00035	0,00852	0,00881	0,00068	0,03844	0,03698
$M[X_3^{(2)}] = 7$	Δb_j	0,00001	-0,00051	0,00130	-0,00022	0,03213	-0,03004
$M[X_2^{(1)}] = 26$,	β_j	0,99931	-0,03331	-0,01666	0,99931	-0,03331	-0,01666
$M[X_3^{(1)}] = 23$,	$\bar{b}_j$	0,99931	-0,03303	-0,01676	0,99894	-0,02170	-0,02813
$M[X_2^{(2)}] = 14$,	$s(b_j)$	0,00010	0,00499	0,00493	0,00090	0,02441	0,02436
$M[X_3^{(2)}] = 17$	Δb_j	0,00001	0,00028	-0,00011	-0,00037	0,01161	-0,01148

В таблицах 1 и 2 приняты следующие обозначения: β_j – теоретические значения коэффициентов логистической регрессии; $\bar{b}_j$ – средние значения оценок коэффициентов логистической регрессии; $s(b_j)$ – средние квадратические значения ошибок оценок коэффициентов логистической регрессии; Δb_j – средние значения ошибок оценок коэффициентов логистической регрессии.

Анализ таблиц показывает, что предложенный алгоритм значительно выигрывает по сравнению с известным.

Заключение

Исследования показали вычислительную неустойчивость алгоритмов оценивания логистической регрессии, реализующих критерий максимального правдоподобия. С целью повышения точности оценок предложен новый итерационный алгоритм. Он включает последовательность задач минимизации нулевого порядка на основе случайного поиска. На каждом k -м шаге длина вектора коэффициентов $\mathbf{b}^{(k)}$ является фиксированной. Ее постепенно увеличивают до достижения целевой функцией требуемого значения. Результаты статистического моделирования показали более высокую точность оценок логистической регрессии предложенного алгоритма по сравнению с известными.

ЛИТЕРАТУРА

1. Azen R., Walker C.M. Categorical Data Analysis for the Behavioral and Social Sciences. Routledge, 2011. 283 p.
2. Lachin J.M. Biostatistical Methods: the Assessment of Relative Risks. 2nd edition. Wiley, 2011. 644 p.
3. Shoukri M.M., Pause C.A. Statistical Methods for Health Sciences. 2nd edition. CRC Press, 1999. 390 p.
4. Магнус Я.Р., Катышев П.К., Пересецкий А.А. Эконометрика. Начальный курс. 6-е изд., перераб. и доп. М. : Дело, 2004. 576 с.
5. Мятлев В.Д., Панченко Л.А., Ризниченко Г.Ю., Терехин А.Т. Теория вероятностей и математическая статистика. Математические модели. М. : Академия, 2009. 320 с.
6. Воронцов К.В. Лекции по алгоритмам восстановления регрессии. 2007. 37 с. URL: <http://www.ccas.ru/voron/download/Regression.pdf> (дата обращения: 15.03.2016).
7. Васильев Н.П., Егоров А.А. Опыт расчета параметров логистической регрессии методом Ньютона–Рафсона для оценки зимостойкости растений // Математическая биология и биоинформатика. 2011. Т. 6, № 2. С. 190–199. URL: http://www.mathbio.org/article_pdf.php?id=82 (дата обращения: 15.03.2016)
8. Пантелеев А.В., Летова Т.А. Методы оптимизации в примерах и задачах. 2-е изд., испр. М. : Высшая школа, 2005. 544 с.
9. Бусленко Н.П., Шрейдер Ю.А. Метод статистических испытаний (Монте-Карло) и его реализация на цифровых вычислительных машинах. М. : ФИЗМАТЛИТ, 1961. 226 с.
10. Ильин В.А., Садовничий В.А., Сендов Бл.Х. Математический анализ. Начальный курс. 2-е изд., перераб. М. : Изд-во МГУ, 1985. 662 с.
11. Рокафеллар Р. Выпуклый анализ : пер. с англ. М. : Мир, 1973. 472 с.

Тырсин Александр Николаевич, д-р техн. наук. E-mail: at2001@yandex.ru

Уральский федеральный университет имени первого Президента России Б.Н. Ельцина (г. Екатеринбург)

Костин Кирилл Константинович. E-mail: lemwwwar@gmail.com

Южно-Уральский государственный университет (национальный исследовательский университет) (г. Челябинск)

Поступила в редакцию 13 марта 2017 г.

Tyrsin Alexander N., Kostin Kirill K. (Ural Federal University named after the first President of Russia B.N. Yeltsin, South Ural State University (National Research University), Russian Federation).

Consideration of estimation of logistic regression as an optimization problem.

Keywords: logistic regression; the maximum likelihood method; the algorithm of descent; random search; model; objective function.

DOI: 10.17223/19988605/40/6

The logistic regression is one of the common methods of data classification in various spheres. The main goal of logistic regression is a separation of multiple input values with a linear boundary (a dividing line, plane or hyperplane) on two classes corresponding to two given region. Logistic regression predicts the probability of some events that are in the range from 0 to 1. Should indicate that the L classes can be more than two, in this case we have a multinomial regression. It can be built using $L - 1$ independent logistic regressions.

The research of an estimation algorithm logistic regression on model data implemented in statistical packages, showed instability of the estimates.

The goal of this article is research of the features of the procedure of estimation of logistic regression as an extreme problem, as well as the development of a stable algorithm of computing the coefficients of the separating equation.

Nowadays the logistic regression is usually built as an optimization problem $Q(\mathbf{b}) = \sum_{i=1}^n \ln(1 + e^{-y_i \mathbf{b}^T \mathbf{x}_i}) \rightarrow \min_{\mathbf{b} \in \mathbb{R}^n}$ that implements the criterion of maximum likelihood. In this article the features of estimation of logistic regression are considered as an optimization problem. There is shown that this problem has significant differences compared to the search of a local extremum.

There are investigated causes of the instability of estimates of the logistic regression. It is shown that using $\mathbf{b}^{(0)}$ as initial approximation for the solve of the classification task as a linear regression often yields unsatisfactory results. The second reason for the instability of estimates of maximum likelihood is that the iterative descent algorithms are designed to find a local minimum. The target function with the correct classification of all precedents has zero as the lower bound at infinity. Moreover the quality of the building of the dividing line in the formal descent wouldn't be associated with the value of the objective function $Q(\mathbf{b})$. So adjust of the strategy of the descent is needed. The third reason is that the some values, for example, $y_i \mathbf{b}^T \mathbf{x}_i$ can be very large when the value of objective function $Q(\mathbf{b})$ is small. It leads to fatal errors for computing and even can stop the operation of the algorithm due to memory overflow.

The stable algorithm for calculating the coefficients of the separating equation of the logistic regression is described. As initial approximation a hyperplane is used. The hyperplane is orthogonal to the segment connecting the centers of gravity of classes, and is passed through the middle of this segment. The iterative algorithm involves a sequence of problems of minimizing zero-order based on a random search. At each k -th step the length of the vector of coefficients $\mathbf{b}^{(k)}$ is fixed. It is gradually increased until the objective function achieves the required value.

The results of statistical modeling of comparative analysis of estimation of logistic regression with the existing algorithm are given.

REFERENCES

1. Azen, R. & Walker, C.M. (2011) *Categorical Data Analysis for the Behavioral and Social Sciences*. Routledge.
2. Lachin, J.M. (2011) *Biostatistical Methods: The Assessment of Relative Risks*. 2nd ed. Wiley.
3. Shoukri, M.M. & Pause, C.A. (1999) *Statistical Methods for Health Sciences*. 2nd ed. CRC Press.
4. Magnus, Ja.R., Katyshev, P.K. & Pereseckij, A.A. (2004) *Ekonometrika. Nachal'nyy kurs* [Econometrics. The Starting Course]. 6th ed. Moscow: Delo. (In Russian).
5. Myatlev, V.D., Panchenko, L.A., Riznichenko, G.Yu. & Terekhin, A.T. (2009) *Teoriya veroyatnostey i matematicheskaya statistika. Matemati-cheskie modeli* [Probability theory and mathematical statistics. Mathematical models]. Moscow: Akademiya.
6. Vorontsov, K.V. (2007) *Leksii po algoritmam vosstanovleniya regressii* [Lectures about the recovery algorithms regression]. [Online] Available from: <http://www.ccas.ru/voron/download/Regression.pdf>.
7. Vasiliev, N.P. & Egorov A.A. (2011) Experience of logistic regression parameters calculation by Newton-Rafson method to estimation resistance to cold of plants. *Matematicheskaya biologiya i bioinformatika – Mathematical Biology and Bioinformatics*. 6(2). pp. 190–199. (In Russian).
8. Pantelev, A.V. & Letova, T.A. (2005) *Metody optimizatsii v primerakh i zadachakh* [Optimization methods in examples and problems]. 2nd ed. Moscow: Vysshaya shkola.
9. Buslenko, N.P. & Shreider Yu.A. (1961) *Metod statisticheskikh ispytaniy (Monte-Karlo) i ego realizatsiya na tsifrovyykh vychislitel'nykh mashinakh* [The method of statistical trials (Monte–Carlo) and its implementation on digital computers]. Moscow: FIZ-MATLIT.
10. Ilyin, V.A., Sadovnichiy, V.A. & Sendov, Bl.H. (1985) *Matematicheskiy analiz. Nachal'nyy kurs* [Mathematical analysis. The starting Course]. 2nd ed. Moscow: Moscow State University.
11. Rockafellar, R.T. (1970) *Vypuklyy analiz* [Convex Analysis]. Translated from English by A. Ioffe, V. Tikhomirov. Princeton: Princeton University Press.

ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ

УДК 681.323

DOI: 10.17223/19988605/40/7

Н.А. Лукин, А.Ю. Филимонов**ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ АЛГОРИТМОВ
НА АРХИТЕКТУРАХ ОДНОРОДНЫХ ВЫЧИСЛИТЕЛЬНЫХ СРЕД**

Рассматриваются проблемы программирования систем с массивным параллелизмом вычислений и подходы к их решению. Приведены примеры реализации нетрадиционных подходов к построению платформ программирования массивно-параллельных систем.

Ключевые слова: однородные вычислительные среды; императивные и декларативные языки программирования параллельных вычислений; системы потоковой обработки.

Однородными вычислительными средами (ОВС) принято называть специализированные вычислительные системы, которые образованы из одинаково соединенных друг с другом универсальных вычислительных элементов, каждый из которых программно настраивается на выполнение арифметической или логической функции, а также функции соединения с соседними элементами [1, 2]. При этом под универсальным процессорным элементом (ПЭ), как правило, подразумевается однобитный процессор, выполняющий последовательную обработку данных, которые поступают на его входы от соседних процессоров.

Основным достоинством однородных вычислительных сред является изначально присущий им массовый параллелизм вычислений. Кроме того, регулярность ОВС позволяет наращивать ее вычислительные возможности путем простого увеличения размеров матрицы, позволяя при этом организовать достаточно простые методы контроля и диагностики ОВС в процессе производства и эксплуатации. Избыточность, изначально присущая однородной среде, позволяет, благодаря наличию резервных элементов в схеме, обеспечить при необходимости высокую живучесть вычислительной системы.

Указанные выше достоинства ОВС позволяют эффективно использовать их при построении функционально-ориентированных процессоров (ФОП) для систем реального времени. При этом номенклатура реализуемых алгоритмов может быть весьма велика:

- цифровая обработка сигналов и изображений;
- распознавание образов;
- решение систем уравнений (алгебраических и дифференциальных);
- логико-комбинаторные алгоритмы, прежде всего, сортировка и поиск.

Программирование ОВС представляет собой технологический процесс настройки массива ПЭ на выполнение операций обработки и передачи данных, который завершается «укладкой» графа алгоритма в вычислительную решетку. Настоящая работа посвящена рассмотрению некоторых компонентов этого технологического процесса, а также проблемам, с которыми сталкиваются создатели подобных платформ.

В первом разделе рассматриваются вопросы применимости универсальных языков для программирования параллельных вычислительных систем и сравниваются основные подходы, которые применяются при создании соответствующих программных платформ.

Второй раздел посвящен анализу опыта применения традиционного подхода к созданию специальных языков программирования параллельных вычислительных систем.

Специфические особенности программирования вычислительных систем, управляемых потоками данных, рассматриваются в третьем разделе.

В четвертом разделе рассмотрен опыт реализации комплексного технологического процесса программирования потоковых вычислительных систем.

В заключении дан краткий анализ перспектив применения современных параллельных программных средств для программирования ОВС.

1. Применение традиционных подходов для программирования однородных вычислительных систем

1.1. Адаптация традиционных языков программирования

Достоинством данного подхода является возможность плавной миграции всего комплекса современных программных продуктов на новые высокопроизводительные параллельные вычислительные системы. Однако на сегодняшний день многочисленные проекты по созданию универсальных векторизирующих компиляторов показывают обнадеживающие результаты только при решении весьма ограниченного класса задач [3], ни один из них не увенчался сколько-нибудь заметными успехами в распараллеливании последовательной программы произвольного вида [4–6].

«Параллельная невыразительность» традиционных языков программирования может быть устранена путем ввода специальных лексических конструкций, которые позволяют программисту описывать и организовывать параллельные вычислительные процессы. Как показывает опыт, при практической реализации данного подхода могут возникать очень серьезные трудности, которые применительно к параллельному языку HPF (High Performance Fortran) описаны в отчетной работе [7] группой основных разработчиков. Примечательно, что программисты в качестве основных недостатков указывали отсутствие возможности оценки (прогнозирования) времени выполнения программы HPF на конкретной вычислительной архитектуре.

Сегодня наибольшее распространение получило применение параллельных расширений (диалектов) существующих последовательных языков для программирования вычислительных структур, которые имеют заранее определенную архитектуру, как правило, это кластеры универсальных процессоров или гибридные вычислители, состоящие из универсальных вычислителей и векторных сопроцессоров (GPU/CPU). Поскольку в этом случае вычислительные комплексы содержат компоненты с классической архитектурой, дополнительные преимущества при их программировании получают именно те платформы, которые максимально используют возможности последовательных языков программирования (Open CL, Open MP, UPC и т.д.).

1.2. Языки параллельного программирования

Работы по созданию универсальных параллельных языков проводились компаниями Cray, SUN и IBM в период с 2002 по 2010 г. по заказу агентства перспективных проектов (Defense Advanced Research Projects Agency – DARPA) Министерства обороны США в рамках программы создания высокопроизводительных вычислительных систем (High Productivity Computing Systems – HPCS) и завершились представлением языков программирования Chapel (2009 г.), Fortress (2008 г.) и X10 (2004 г.) [8]. Эти языки основаны на модели разделяемого адресного пространства Partitioned Global Address Space (PGAS), причем X10 в большей степени, чем Chapel и Fortress, опирается на традиционный синтаксис и базовые понятия последовательных языков. Тот факт, что даже самый близкий к традиционным из указанных языков (X10) не получил заметного распространения, разработчики объясняют неготовностью сообщества программистов жертвовать удобством процесса программирования ради повышения его эффективности [9]. В то же время приведенные в [10, 11] выводы о большей выразительности и универсальности Chapel по сравнению с X10 указывают на ограниченные возможности применения традиционных методов для программирования систем с массовым параллелизмом вычислений.

В условиях отсутствия универсальных инструментов параллельные проблемно-ориентированные языки (ПОЯ) (Domain Specific Language – DSL) позволяют достичь требуемой эффективности и выразительности программирования за счет существенного сокращения области применения языков [12].

Применение ПОЯ, как правило, предполагает полную осведомленность программиста об особенностях организации вычислительного процесса в программируемом устройстве, что позволяет наиболее полно использовать специфические возможности данного вычислителя [14]. Поэтому ПОЯ в первую очередь оказываются полезными в качестве временного решения на этапе макетирования вычислительных систем [13].

1.3. Перспективы применения традиционных программных средств для программирования ОВС

Практически все рассмотренные выше языки, как и подавляющее большинство современных языков программирования, являются императивными и основаны на принципе обработки данных потоками команд (Control Flow). Повсеместное распространение императивных языков программирования определяется их точным соответствием требованиям и особенностям организации вычислительного процесса на вычислителях с классической архитектурой от фон Неймана. Однако в тех случаях, когда архитектура программируемого вычислителя отличается от классической, это преимущество императивных языков программирования превращается в недостаток, который будет проявляться тем сильнее, чем сильнее различаются архитектуры вычислителей. Например, применение императивных языков для программирования систем с архитектурой Data Flow, которую образует массив исполнительных устройств и устройств сопоставления соединенных с контекстно-адресуемым запоминающим устройством, вызывает серьезные проблемы, связанные с побочными эффектами и ограниченной доступностью данных [15]. Примечательно, что в последнее время создаются серьезные проекты, основанные на модели вычислений Data Flow. Так, в частности, разработчики крупного европейского проекта TERAFLUX по созданию высокопроизводительных вычислительных структур отмечают, что применение модели организации вычислительного процесса Data Flow позволяет максимально выявлять и использовать параллелизм вычислений и преодолеть ряд существенных ограничений, присущих традиционной модели Control Flow [16]. Эти результаты, безусловно, следует принимать во внимание при выборе системы программирования ОВС, поскольку модель организации вычислительного процесса Data Flow гораздо лучше, чем Control Flow, подходит для описания вычислений в ОВС, а форма графа потоков данных, используемая для указания информационных связей между операндами выражения, вычисляемого в системе DataFlow, по сути, представляет собой архитектуру ОВС, которая обеспечивает вычисление данного выражения.

2. Применение альтернативных подходов для программирования однородных вычислительных систем

2.1. Программирование вычислений, управляемых потоками данных (Data Flow)

Обобщенные в отчете [15] результаты многочисленных исследований показывают, что ключевым фактором, который способен обеспечить требуемую эффективность программирования вычислителей, управляемых потоком данных, являются радикальные изменения понятия переменной и процедуры присваивания. В обзоре [17] приведены результаты использования адаптированного языка C для программирования алгоритмов обработки изображений на реконфигурируемом вычислителе, построенном на FPGA семейства Virtex (Xilinx, США). Язык программирования SA-C представляет собой диалект, наиболее характерными отличиями от базового языка C являются:

- принцип однократного присваивания значений переменным (single assignment);
- исключение операций с компонентами общей памяти;
- ориентированность на выражения (expression oriented – любая языковая конструкция обязательно возвращает как минимум одно значение).

В ряде случаев для обеспечения выразительности и адекватности описания вычислительного процесса в потоковых вычислительных системах с массивным параллелизмом платформы программирования, основанные на императивных языках, вынуждены перенимать базовые принципы декларативной концепции программирования [18, 19]. К преимуществам использования декларативных языков для

программирования однородных вычислительных систем следует отнести отсутствие конструкций и понятий, которые не имеют адекватного отображения в ОВС, – такие как общая память, переменные, указатели, индексы, циклы. Главными факторами, которые сдерживают применение и распространение декларативной парадигмы программирования, долгое время оставались отсутствие зрелых платформ программирования и психологическая неготовность сообщества программистов к радикальному изменению своего инструментария [20, 21]. Однако за последние несколько лет отношение разработчиков программного обеспечения к декларативным языкам стало меняться, что привело к появлению гибридных языков и новых декларативных платформ для программирования параллельных вычислителей.

2.2. Не-императивные языки программирования

Анализ динамики (по данным Wikipedia) создания новых языков программирования за последние 15 лет показывает устойчивое уменьшение среди новых языков программирования доли чистых императивных языков (–24%) и такой же устойчивый рост доли гибридных языков, которые поддерживают декларативный стиль программирования (+34%). Объяснить это можно двумя причинами:

1) отсутствием заметного прогресса в применении императивных программных систем для программирования вычислителей с массивным параллелизмом вычислений;

2) появлением и все более широким внедрением распределенных облачных и туманных вычислений (cloud, fog computing); основной тенденцией в организации вычислительных процессов становится уход от классической схемы фон Неймана, что уменьшает область эффективного применения чистых императивных языков программирования.

Гибридные языки, по замыслу разработчиков, должны сочетать в себе основные достоинства указанных выше парадигм программирования:

– наглядность (простота восприятия) программного кода, наличие мощных инструментов для его создания и отработки, которые свойственны современным императивным языкам программирования;

– автоматическое выявление ветвей параллельных вычислений, отсутствие побочных эффектов при их выполнении, которые свойственны чистым декларативным языкам программирования.

В качестве примеров реализации подобных решений в последнем разделе будут рассмотрены платформа программирования FPGA-сопроцессоров, основанная на гибридном языке Mittrion-C, и платформа программирования параллельных вычислений, основанная на декларативном языке SequenceL. Поскольку данные платформы способны обеспечить массовый параллелизм вычислительного процесса, основные принципы их построения и опыт реализации могут быть использованы при создании систем программирования однородных вычислительных сред.

3. Платформы программирования систем с массовым параллелизмом вычислений

3.1. Платформа, основанная на гибридном языке программирования

Платформа компании Mittrion включает в себя все компоненты, которые необходимы для реализации полного цикла программирования сопроцессоров, построенных на FPGA:

– транслятор Mittrion-C, который предназначен для лингвистической верификации исходного кода, написанного на языке программирования высокого уровня, и формирования объектного кода программы;

– виртуальный процессор Mittrion, который обеспечивает отладку объектного кода программы на интерпретаторе с использованием FPGA – независимого эмулятора;

– блоки конфигурирования (Processor Configuration Unit, SPR Tool), которые выполняют размещение вычислительных компонентов на FPGA, создание соединений между ними (Place & Route) и обеспечивают формирование загрузочного кода (FPGA bitstream – код укладки программы) с учетом специфики выбранной аппаратной платформы FPGA.

По замыслу разработчиков платформы Mittrion, язык программирования должен обеспечивать интеллектуальную поддержку программиста в обеспечении параллельного выполнения программ и в то же время быть достаточно простым для изучения и использования [22].

Поскольку лексика языка Mitron-C не опирается явно на принятые в декларативных языках механизмы описания шаблонов функций, она, на первый взгляд, гораздо ближе к лексике императивных языков программирования. Декларативный характер Mitron-C подчеркивается также принятыми в данном языке программирования способами определения и использования переменных; присвоение значения переменной в языке программирования Mitron-C представляет собой не выражение, как это принято в императивных языках программирования, а утверждение. Отличие заключается в том, что единожды определенное при помощи процедуры присваивания значение переменной Mitron-C может затем многократно использоваться (например, для определения значений других переменных), но, в то же время, не может изменяться (переопределяться) [22].

Особый интерес при этом, безусловно, представляют оценки выразительности нетрадиционных лингвистических конструкций языка Mitron-C. Действительно, многие испытатели отмечают, что им требовалось определенное время для того, чтобы привыкнуть к специфическим особенностям программирования на Mitron-C [23, 24]. Большинство из них, однако, при этом указывают, что после завершения адаптации процесс программирования уже не у них вызывал значительных затруднений. При этом программисты по достоинству оценили оригинальные решения разработчиков языка, которые обеспечивают возможность управления точностью представления данных [23] и функциональную полноту платформы, которая обеспечивает выполнение предварительной отладки программных решений на интерпретаторе [25].

Предварительные сравнительные оценки средств программирования FPGA-сопроцессоров [24, 26–28] показывают, что платформа Mitron (язык Mitron-C) имеет преимущество в эффективности по отношению к традиционным программным средствам (Impulse-C и Dime-C), но уступает как средствам аппаратного (VHDL и Verilog), так и графического (DLPLogic) программирования. Рассматривая эти результаты, не стоит забывать о том, что только программные платформы способны обеспечить мобильность программного кода и востребованную сегодня возможность его независимой отладки [25, 29].

Не вызывает сомнения, что перспективы применения для программирования ОВС платформ, основанных на гибридных языках, во многом будут определяться совокупностью взаимно исключающих требований удобства программирования и степени абстрагирования от архитектуры вычислительного устройства. Основываясь на изложенном выше (здесь уместно будет вспомнить опасения [5] разработчиков Chapel), можно предположить, что чем больше императивных возможностей разработчик гибридного языка оставит программисту, тем менее приспособленный для параллельного выполнения код получит транслятор. Поэтому значительно более перспективным представляется применение для программирования ОВС решений, которые основаны на чистых декларативных языках программирования. Как будет показано далее, лаконичность программных конструкций в совокупности с развитыми процедурами выявления параллелизма позволяет успешно решать многие проблемы, с которыми могут столкнуться разработчики систем программирования однородных вычислительных сред.

3.2. Платформа, основанная на декларативном языке программирования

Очевидно, что максимально эффективной (по определению) будет программа, которая наиболее точно адаптирована к архитектуре вычислительной системы. Важно при этом определить, на какой именно стадии создания программы и каким образом должна происходить эта адаптация. Наиболее неудачным представляется вариант, когда программа изначально строится на основе типового алгоритма, который не учитывает ни структуру обрабатываемых данных, ни архитектуру вычислительной системы, а потом под них адаптируется программистом или компилятором. Проблема заключается в том, что алгоритм сам по себе уже является адаптацией выполняемой задачи, как правило, для классической архитектуры фон Неймана, поэтому его автоматическая адаптация компилятором сводится к решению описанной ранее проблемы построения «универсального векторизирующего компилятора» [3], а процесс ручной адаптации рано или поздно возвращает программиста к условиям поставленной задачи. Значительно более продуктивным поэтому представляется процесс построения программы непосредственно из условий решаемой задачи с одновременным учетом структур обрабатываемых данных и доступных функций.

Во многом подобный процесс реализован в системе программирования, основанной на языке SequenceL, представляющем собой чистый декларативный язык высокого уровня, который предназначен для программирования распределенных и потоковых вычислений [29]. Отличительными особенностями этого языка программирования является очень лаконичный синтаксис и развитые механизмы неявного распараллеливания вычислений.

По замыслу создателей, эти особенности языка SequenceL позволяют избавить программиста от алгоритмического планирования вычислительного процесса, что особенно важно для программирования реконфигурируемых систем с массивованным параллелизмом вычислений, когда структура (архитектура) вычислительного устройства не определена на момент создания программы. В то же время лаконичная семантика SequenceL поощряет программиста формулировать решаемые задачи в форме, которая способствует выявлению и использованию скрытого параллелизма вычислений [28].

Как следует из названия языка, базовым элементом данных в SequenceL являются последовательности, элементами которых также могут быть последовательности. Скаляр представляется в SequenceL в виде 0-мерной последовательности, которая состоит из одного элемента, вектор представляется одномерной последовательностью, матрица – двумерной (последовательность векторов) и т.д. Размерность (Type – тип) аргументов указывается в определениях функций и операторов SequenceL и играет ключевую роль для распараллеливания их вычислений [30]. В случае, если тип аргумента при обращении к некоторой функции превышает значение типа, которое было задано при ее определении, для этого аргумента фиксируется состояние «превышение типа» (overtyping), которое влечет за собой выполнение для него и для данной функции комплекса согласующих операций – NTD (Normalize-Transpose-Distribute). Назначение операций NTD поясняется приводимым ниже примером умножения (скалярная операция) на вектор, который взят из [30]:

```
10 * [1,2,3]
(normalize) → [[10,10,10],[1,2,3]]
(transpose) → [[10, 1],[10,2],[10,3]]
(distribute) → [[10*1],[10*2],[10*3]]
→ [10,20,30]
```

Применение комплекса NTD в языке SequenceL позволяет на стадии трансляции выражений автоматически выявлять скрытый параллелизм их вычисления и одновременно избавляет программиста от необходимости использования дополнительных программных конструкций для управления вычислительным процессом [30].

Интересно отметить, что хотя описанные выше основные принципы построения языка программирования SequenceL были представлены разработчиками еще в прошлом веке [28], заметное распространение этот язык начал получать только в последние несколько лет. В апреле 2012 г. основанная разработчиками языка компания Texas Multicore Technologies (TMT) представила бета-версию коммерческого компилятора SequenceL, который использовался в качестве пре-процессора к языку C++ для автоматического распараллеливания вычислений. В декабре 2014 г. TMT, уже будучи одним из партнеров AMD в области разработки программного обеспечения, представила новую версию своей платформы, которая обеспечивала совместное использование SequenceL с такими языками программирования, как Java, C#, Python и Fortran, что обеспечило успешное применение для программирования распределенных вычислений. Как показывают испытания, сегодня платформа программирования TMT, которая включает в себя интерпретатор, отладчик, компилятор SequenceL и системные библиотеки для поддерживаемых языков программирования, позволяет автоматически создавать программы, которые превосходят написанные вручную по эффективности использования вычислительных ресурсов [31].

Опыт разработки и внедрения платформы программирования TMT подтверждает представленное выше предположение о растущем интересе к системам, которые основаны на не-императивных языках программирования в связи с наблюдаемыми системными изменениями в организации вычислительного процесса (тенденция ухода от классической схемы фон Неймана).

3.3. Общие требования к платформе программирования ОВС

Представленный выше анализ текущего состояния и тенденций развития технологий программирования систем с массовым параллелизмом вычислений позволяет определить основные требования к составу и функциям компонентов платформы программирования однородных вычислительных сред. Основными компонентами такой платформы должны быть собственно система программирования и система построения загрузочного кода (компоновки).

Система программирования, по нашему мнению, должна быть построена на декларативном языке и обеспечивать синтаксическую и семантическую отладку управления точностью и разрядностью вычислений.

Система трансляции предназначена для формирования объектного кода разработанной программы с учетом особенности организации вычислительного процесса в ОВС и специфики их компоновки, а при необходимости – автоматического эквивалентного преобразования объектного кода.

Система компоновки должна быть построена по блочно-модульному принципу с тем, чтобы обеспечить итеративную форму адаптации объектного кода для оптимизации процесса построения загрузочного кода и возможность дальнейшего развития системы.

Наличие, кроме естественной временной метрики обработки данных (потактные вычисления потоков данных на массивах ПЭ), еще и явно выраженной пространственной метрики (прямоугольная матрица локально связанных ПЭ) обуславливает «геометрический» характер функционирования компилятора. Он должен учитывать возможные траектории распространения потоков данных в двумерном пространстве ПЭ и оптимизировать длины этих траекторий. Это принципиально отличает принципы построения и функционирования компиляторов ОВС от компиляторов современных систем параллельной обработки данных.

Отмеченные особенности ложатся в основу создания технологии программирования ОВС, при этом первым этапом становится создание языка и принципов построения ОВС-компилятора.

Заключение

ОВС представляют собой класс архитектур с массовым параллелизмом, реализуемым на двумерных массивах локально связанных процессорных элементов. По принципу обработки информации это реконфигурируемые машины потоков данных с однородной структурой. Поэтому основными чертами технологии программирования ОВС являются декларативный принцип языка программирования, структурно-функциональный принцип работы компилятора, учитывающий топологию ОВС и возможность программного управления архитектурой ОВС (реконфигурации) при возникновении соответствующих условий и проблем.

ЛИТЕРАТУРА

1. Дмитриенко Н.Н., Каляев И.А., Левин И.И., Семерников Е.А. Семейство многопроцессорных вычислительных систем с динамически перестраиваемой архитектурой // Вестник компьютерных и информационных технологий. 2009. № 6, 7.
2. Лукин Н.А. Реконфигурируемые процессорные массивы для систем реального времени: архитектуры, эффективность, области применения // Известия ТРТУ. 2004. № 9. С. 36–45.
3. Maleki S., Yaoqing Gao, Garzaran M.J., Wong T., Padua D.A. An Evaluation of Vectorizing Compilers Parallel Architectures and Compilation Techniques (PACT) // 2011 International Conference. 2011. 10–14 Oct. P. 372–382.
4. The Landscape of Parallel Computing Research: A View from Berkeley – Technical Report No. UCB/EECS-2006-183.
5. Chamberlain B.L., Callahan D., Zima H.P. Parallel Programmability and the Chapel Language // Journal of High Performance Computing Applications. 2007. August. V. 21, No. 3. P. 291–312.
6. Bal H.E. Matthew Haines Approaches for Integrating Task and Data Parallelism // Journal IEEE Concurrency – July. 1998. V. 6, Issue 3. P. 74–84.
7. Kennedy K., Koelbel C., Zima H. The Rise and Fall of High Performance Fortran: An Historical Object Lesson // Communications of the ACM. November. 2011. V. 54, No. 11. P. 74–82.
8. Dongarra J., Graybill R., Harrod W., Lucas R., Lusk E., Luszczek P., McMahon J., Snively A., Vetter J., Yelick K., Alam S., Campbell R., Carrington L., Tzu-Yi Chen, Khalili O., Meredith J., Tikir M. DARPA's HPCS Program: History, Models, Tools // Languages Advances in Computers. 2008. V. 72. P. 1–100.

9. Richards J., Brezin J. A decade of progress in parallel programming productivity // Communications of the ACM. November. 2014. V. 57, Issue 11. P. 60–66.
10. Weiland M. Chapel, Fortress and X10: Novel languages for HPC. Technical Report from the HPCx Consortium. 2007.
11. Feldman M. Closing the Parallelism Gap with the Chapel Language // HPCwire. 2008. November 19.
12. Voemel C. How to talk new computers into working harder. Technical Reports № 603. ETH Zurich. ICOS–08. 2008.
13. Титов В.Г., Лукин Н.А. Язык макросов для программирования однородной вычислительной среды MiniTera II // Известия Томского политехнического университета. 2008. Т. 313, № 5. С. 93–96.
14. Васильев С.С., Новосельцев В.Б. Об использовании в программировании проблемно-ориентированных языков // Известия Томского политехнического университета. 2008. Т. 313, № 5. С. 68–72.
15. Johnston W.M., Hanna J.R.P. and Millar R.J. Advances in dataflow programming languages // ACM Comput. Surv. 2004. V. 36, No. 1.
16. Solinas M., Badia R.M., Bodin F., Cohen A. The TERAFLUX Project: Exploiting the DataFlow Paradigm in Next Generation Teradevices // 16th Euromicro Conference on Digital System Design. 2013.
17. Najjar W.A., Bohm W., Draper B.A., Hammes J., Rinker R., Beveridge J.R., Chawathe M., Ross C. High-Level Language Abstraction for Reconfigurable Computing // IEEE Computer. 2003. Aug. P. 63–69.
18. Böhm W., Hammes J., Draper B., Chawathe M., Ross C., Rinker R., Najjar W. Mapping a Single Assignment Programming Language to Reconfigurable Systems // Supercomputing 21. 2002. P. 117–130.
19. Hammarlund P., Lisper B. Data Parallel Programming: A Survey and a Proposal for a New Model // Royal Institute of Technology, Department of Teleinformatics. Tech. Rep. TRITA-IT-9308. September. 1993.
20. Wadler P. Why no one uses functional languages // ACM SIGPLAN Notices. 1998. V. 33, No. 8. P. 23–27.
21. Budiu M., Goldstein S.C. Compiling application-specific hardware // In International Conference on Field Programmable Logic and Applications (FPL). Montpellier (La Grande-Motte). France. 2002. September. P. 853–863.
22. Mittrion Users' Guide 2.0.3-001. 2009.
23. Kryjak T., Gorgoń M. Parallel implementation of local thresholding in Mittrion-C // AGH University of Science and Technology, Al. Mickiewicza. Department of Automatics. Laboratory of Biocybernetics Poland. Cracow. 2010. September. V. 20, Issue 3. P. 571–580.
24. El-Araby E., Taher M., Abouellail M., ElGhazawi T. and Newby G.B. Comparative Analysis of High Level Programming for Reconfigurable Computers: Methodology and Empirical Study // III Southern Conference on Programmable Logic (SPL 2007). Argentina. Mar del Plata, 2007.
25. Koo J., Fernandez D., Haddad A. Gross W.J. Evaluation of a High-Level-Language Methodology for High-Performance Reconfigurable Computers // Application-specific Systems. Architectures and Processors ASAP – IEEE International Conf. 2007.
26. Park S.J., Shires D., Henz B. Reconfigurable Computing: Experiences and Methodologies // Army Research Laboratory Aberdeen Proving Ground. MD 21005-5067 ARL-TR-4358. 2008. January.
27. Kindratenko V.V., Brunner R.J., Myers A.D. Mittrion-C Application Development on SGI Altix 350/RC100 // IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'07). 2006. April 23–25.
28. Cooke D., Gelfond M., Urban J. Computer Language Advances // Handbook of Software Engineering and Knowledge Representation. World Scientific. 2001. V. 1. P. 121.
29. Cooke D., Rushton N. SequenceL – An Overview of a Simple Language // Proceedings of The 2005 International Conference on Programming Languages and Compilers PLC. 2005. USA. Nevada. Las Vegas. 2005. June 27–30.
31. Daniel E., Cooke J., Rushton N., Nemanich B., Watson R., Andersen P. Normalize, transpose, and distribute: An automatic approach for handling nonscalars // ACM Trans. Program. Lang. Syst. 2008. V. 30, No. 2.
32. Rushton N., Norton D. SequenceL for C++ programmers saves time and angst // Embedded Computing Design August. 2014.

Лукин Николай Алексеевич, канд. техн. наук. E-mail: n.a.lookin@urfu.ru

Институт машиноведения УрО РАН, Уральский федеральный университет (г. Екатеринбург)

Филимонов Александр Юрьевич. E-mail: af.1015@yahoo.com

Уральский федеральный университет (г. Екатеринбург)

Поступила в редакцию 25 июня 2017 г.

Lookin Nick A. (Institute of Engineering Science, Ural Branch of Russian Academy of Science; Ural Federal University, Yekaterinburg, Russian Federation).

Filimonov Alexander Yu. (Ural Federal University, Yekaterinburg, Russian Federation).

Software technologies for homogenous computing environment.

Keywords: homogenous computing environment; imperative and declarative programming languages for parallel computations; data flow systems.

DOI: 10.17223/19988605/40/7

Homogeneous computing environment (HCE) it's special-purpose computer systems, which are represents as the mesh-connected identical processing elements (PE), each of which is configured to perform arithmetic or logic functions and certain kind connections with the "neighboring" PE. HCE programming is a PE array set-up process for the implementation of data processing and transmission algorithm. The main result of programming is "laying" algorithm graphs on lattice of PE.

To date, numerous projects dedicated to applying of general-purpose vectorizing compilers for adaptation the common programming languages to systems with massively parallel data processing have not been any success with respect to the parallel programs of any kind. At the same time, some of new languages for parallel programming are not getting noticeable proliferation, indicating that the limited capacity of traditional methods for systems programming of massively parallel computing.

The results of numerous researches show that the one of key factor for the efficiency of parallel programming is a radical change of the concept of variable and assignment procedures. That is why the platform of software for parallel data processing based on imperative programming languages, are forced to adopt the basic principles of declarative programming concepts.

As examples of such language implementation the FPGA-coprocessor software based on a hybrid language Mitrion-C programming and parallel computing platform based on a declarative language SquenceL are described in article.

Analysis of the current status and trends of software techniques for massively parallel computing allows define the basic requirements for the structure and functions of the components of the software platform of HCE. The main components of such a platform are programming system and the subsystem of code layout.

The programming system should be based on the declarative language and provide debugging on syntactic and semantic levels and accuracy control. The layout system should be based on the modular principle in order to ensure the iterative adaptation of the object code for optimizes the building boot code and the possibility of further development of the system.

In case of HCE both time (data streams computation on PE arrays) and space (rectangular matrix locally connected PE) metric of computations are taking place. It is causes the "geometric" nature of the operation of the compiler. It must take into account the possible propagation path data streams in two-dimensional space of PE and optimize the lengths of these "trajectories". This fact fundamentally distinguishes the principles of construction and functioning of compilers for HCE and usual parallel computer systems.

REFERENCES

1. Dmitrienko, N.N., Kalyaev, I.A., Levin, I.I. & Semernikov, E.A. (2009) Multiprocessor computer systems with dynamically reconfigurable architecture. *Vestnik komp'yuternykh i informatsionnykh tekhnologiy – Herald of Computer and Information Technologies*. no. 6,7. (In Russian).
2. Lukin, N.A. (2004) Rekonfiguriruemye protsessornye massivyy dlya sistem real'nogo vremeni: arkhitektury, effektivnost', oblasti primeneniya [Reconfigurable Processor Arrays for Real-Time Systems: Architectures, Efficiency, Application]. *Izvestiya TRTU*. 9. pp. 36-45. (In Russian).
3. Maleki, S., Yaoqing Gao, Garzaran, M.J., Wong, T. & Padua, D.A. (2011) An Evaluation of Vectorizing Compilers Parallel Architectures and Compilation Techniques (PACT). *PACT '11 Proceedings of the 2011 International Conference on Parallel Architectures and Compilation Techniques*. pp. 372–382. DOI: 10.1109/PACT.2011.68
4. Asanović, K., Bodik, R., Catanzaro, B.C., Gebis, J.J., Husbands, P., Keutzer, K., Patterson, D.A., Plishker, W.L., Shalf, J., Williams, S.W. & Yelick, K.A. (2006) *The Landscape of Parallel Computing Research: A View from Berkeley*. Technical Report No. UCB/EECS-2006-183.
5. Chamberlain, B.L., Callahan, D. & Zima, H.P. (2007) Parallel Programmability and the Chapel Language. *Journal of High Performance Computing Applications*. 21(3). pp. 291–312. DOI: 10.1177/1094342007078442
6. Bal, H.E. & Haines, M. (1998) Approaches for Integrating Task and Data Parallelism. *Journal IEEE Concurrency*. 6(3). pp. 74–84. DOI: 10.1109/4434.708258
7. Kennedy, K., Koelbel, C. & Zima, H. (2011) The Rise and Fall of High Performance Fortran: An Historical Object Lesson. *Communications of the ACM*. 54(11). pp. 74–82. DOI: 10.1145/29873.29875
8. Dongarra, J., Graybill, R., Harrod, W., Lucas, R., Lusk, E., Luszczek, P., McMahon, J., Snively, A., Vetter, J., Yelick, K., Alam, S., Campbell, R., Carrington, L., Tzu-Yi Chen, Khalili, O., Meredith, J. & Tikir, M. (2008) DARPA's HPCS Program: History, Models, Tools. *Languages Advances in Computers*. 72. pp. 1–100. DOI: 10.1016/S0065-2458(08)00001-6
9. Richards, J. & Brezin, J. (2014) A decade of progress in parallel programming productivity. *Communications of the ACM*. 57(11). pp. 60–66. DOI: 10.1145/2669484
10. Weiland, M. (2007) *Chapel, Fortress and X10: Novel languages for HPC*. Technical Report from the HPCx Consortium.
11. Feldman, M. (2008) Closing the Parallelism Gap with the Chapel Language. *HPCwire*. November 19.
12. Voemel, C. (2008) *How to talk new computers into working harder*. Technical Reports №603. ETH Zurich. ICOS 08.
13. Titov, V.G. & Lukin, N.A. (2008) The Macro's Language for Programming of HCE MiniTera II. *Izvestiya Tomskogo politekhnicheskogo universiteta – Bulletin of Tomsk Polytechnic University*. 313(5). pp. 93–96. (In Russian).
14. Vasiliev, S.S. & Novoseltsev, V.B. (2008) On the use of domain-specific languages for programming. *Izvestiya Tomskogo politekhnicheskogo universiteta – Bulletin of Tomsk Polytechnic University*. 313(5). pp. 68–72. (In Russian).
15. Johnston, W.M., Hanna, J.R.P. & Millar, R.J. (2004) Advances in dataflow programming languages. *ACM Comput. Surv.* 369(1). DOI: 10.1145/1013208.1013209
16. Solinas, M., Badia, R.M., Bodin, F. & Cohen, A. (2013) The TERAFLUX Project: Exploiting the DataFlow Paradigm in Next Generation Teradevices. *16th Euromicro Conference on Digital System Design*. pp. 1–6. DOI: 10.1109/DSD.2013.39. 272
17. Najjar, W.A., Bohm, W., Draper, B.A., Hammes, J., Rinker, R., Beveridge, J.R., Chawathe, M. & Ross, C. (2003) High-Level Language Abstraction for Reconfigurable Computing. *IEEE Computer*. Aug. pp. 63–69. DOI: 10.1109/MC.2003.1220583
18. Böhm, W., Hammes, J., Draper, B., Chawathe, M., Ross, C., Rinker, R. & Najjar, W. (2002) Mapping a Single Assignment Programming Language to Reconfigurable Systems. *Supercomputing 21*. pp. 117–130. DOI: 10.1023/A:1013623303037
19. Hammarlund, P. & Lisper, B. (1993) Data Parallel Programming: A Survey and a Proposal for a New Model. *Royal Institute of Technology, Department of Teleinformatics. Tech. Rep. TRITA-IT-9308*. September.

20. Wadler, P. (1998) Why no one uses functional languages. *ACM SIGPLAN Notices*. 339(8). pp. 23–27. DOI: 10.1145/286385.286387
21. Budi, M. & Goldstein, S.C. (2002) Compiling application-specific hardware. *International Conference on Field Programmable Logic and Applications (FPL)*. Montpellier (La Grande-Motte). France. September. pp. 853–863.
22. *Mitrion Users' Guide* 2.0.3-001. (2009).
23. Kryjak, T. & Gorgoń, M. (2010) Parallel implementation of local thresholding in Mitrion-C. *AGH University of Science and Technology, Al. Mickiewicza*. 20(3). pp. 571–580. DOI: 10.2478/v10006-010-0042-2
24. El-Araby, E., Taher, M., Abouellail, M., ElGhazawi, T. & Newby, G.B. (2007) Comparative Analysis of High Level Programming for Reconfigurable Computers: Methodology and Empirical Study. *III Southern Conference on Programmable Logic (SPL 2007)*. Argentina. Mar del Plata.
25. Koo, J., Fernandez, D., Haddad, A. & Gross, W.J. (2007) Evaluation of a High-Level-Language Methodology for High-Performance Reconfigurable Computers. *Application – specific Systems. Architectures and Processors ASAP – IEEE International Conf.* DOI: 10.1109/ASAP.2007.4429954
26. Park, S.J., Shires, D., Henz, B. (2008) Reconfigurable Computing: Experiences and Methodologies. *Army Research Laboratory Aberdeen Proving Ground*. MD 21005-5067 ARL-TR-4358. January.
27. Kindratenko, V.V., Brunner, R.J. & Myers, A.D. (2006) Mitrion-C Application Development on SGI Altix 350/RC100. *IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'07)*. April 23-25. pp. 1–6. DOI: 10.1109/FCCM.2007.17
28. Cooke, D., Gelfond, M. & Urban, J. (2001) Computer Language Advances. *Handbook of Software Engineering and Knowledge Representation*. 1. pp. 121.
29. Cooke, D. & Rushton, N. (2005) Sequence L – An Overview of a Simple Language. *Proceedings of The 2005 International Conference on Programming Languages and Compilers PLC*. USA. Nevada. Las Vegas. June 27–30, 2005.
31. Daniel, E., Cooke, J., Rushton, N., Nemanich, B., Watson, R. & Andersen, P. (2008) Normalize, transpose, and distribute: An automatic approach for handling nonscalars. *ACM Trans. Program. Lang. Syst.* 30(2). DOI: 10.1145/1330017.1330020
32. Rushton, N. & Norton, D. (2014) SequenceL for C++ programmers saves time and angst. *Embedded Computing Design*. August.

М.А. Сонькин, А.Н. Моисеев, Д.М. Сонькин, Д.А. Буртовая

ОБЪЕКТНАЯ МОДЕЛЬ ПРИЛОЖЕНИЯ ДЛЯ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ЦИКЛИЧЕСКИХ СИСТЕМ МАССОВОГО ОБСЛУЖИВАНИЯ

Представлена разработка объектной модели программной системы, предназначенной для имитационного моделирования циклических систем массового обслуживания. Приложение построено на основе дискретно-событийного подхода и включает реализацию различных механизмов группового подключения и переключения обслуживаемых приборов между входящими потоками заявок.

Ключевые слова: циклическая система массового обслуживания; имитационное моделирование; объектно-ориентированное проектирование.

Системы массового обслуживания (СМО) [1, 2] являются стохастическими моделями реальных систем. Циклические СМО – это класс систем массового обслуживания, в которых обслуживаемые приборы в определенном (циклическом) порядке подключаются к различным накопителям для обслуживания заявок, находящихся в этих накопителях [3, 4]. Заявки попадают в накопители из нескольких входящих в систему потоков заявок. Аналитическим исследованиям циклических систем обслуживания посвящено достаточно много научных трудов (см., например, [5]), но несмотря на это, по вполне понятным причинам во многих случаях не удается получить какие-либо аналитические результаты. Тогда применяют другие методы исследования, среди которых особое место занимает имитационное моделирование [6, 7].

Имитационный подход позволяет воспроизвести поведение системы и таким образом достичь требуемых результатов практически для любой конфигурации исследуемой системы. Конечно, имитационное моделирование имеет и свои недостатки – это в первую очередь числовой, возможно, качественный, но не аналитический характер результатов, что позволяет делать подтвержденные выводы только для модели, определенной с точностью до числовых значений, но не для класса систем. Также выполнение имитационного моделирования требует достаточно долгого времени для проведения комплексного исследования. Однако, несмотря на указанные недостатки, имитационный подход часто используется для решения конкретных задач в тех случаях, когда аналитические исследования невозможны либо возможны при больших упрощениях исходной модели.

Для выполнения имитационного моделирования применяют специализированные программные средства, которые можно разделить на два типа: с полным контролем внутренней структуры со стороны аналитика (типа «белый ящик») и без такового (типа «черный ящик»). Системы первого типа наиболее удобно реализовать с помощью различных готовых программных продуктов, специально предназначенных для имитационного моделирования, например, с помощью систем GPSS [8, 9], MatLab [10], AnyLogic [11] и других программных продуктов. Иногда аналитик использует стандартные языки программирования (например, Fortran или C++) для создания такой программы. В любом из этих случаев от исследователя требуются владение на хорошем уровне навыками программирования и / или достаточно глубокие знания специального языка (GPSS) или устройства соответствующей системы (MatLab).

Системы второго типа представляют собой закрытые приложения, в которых аналитику достаточно только выбрать конфигурацию из предложенного списка и задать параметры, не вдаваясь в подробности устройства программной системы и не написав ни строчки кода на каком-либо языке. В данном случае от пользователя приложения не требуется знания основ программирования и даже глубокого знания внутреннего устройства самого процесса имитационного моделирования.

Каждый тип систем моделирования имеет свои положительные и отрицательные стороны. В настоящей работе представлено решение задачи разработки приложения закрытого типа для имитационного моделирования достаточно широкого класса циклических систем обслуживания.

1. Общее описание модели циклической системы массового обслуживания

Рассматривается класс циклических систем массового обслуживания следующего вида (рис. 1). На вход системы поступает $K \geq 1$ входящих потоков $A_1, \dots, A_K$, каждый из которых описывается заданной для него моделью случайного потока событий с определенными параметрами. Все заявки k -го входящего потока ($k = \overline{1, K}$) поступают в k -й накопитель (буфер), который может иметь заданный максимальный размер или может быть не ограничен в приеме заявок. Также имеется некоторое число $N \geq 1$ обслуживающих приборов (серверов) $B_1, \dots, B_N$, которые подключаются к накопителям, забирают из них заявки в порядке, соответствующем некоторой дисциплине (например, FIFO – first in first out), и обслуживают эти заявки в соответствии с некоторым заданным для каждого прибора законом. После этого заявки покидают систему. Функция распределения времени обслуживания может определяться для каждого прибора независимо от накопителя (входящего потока), из которого взята заявка, но также для каждого прибора может быть задан индивидуальный набор из K функций распределения, определяющих продолжительность обслуживания заявок, поступивших из разных входящих потоков. Таким образом, при K входящих потоках (накопителях) и N обслуживающих приборах можно задать $N \times K$ функций распределения $B_{nk}(x)$, $n = \overline{1, N}$, $k = \overline{1, K}$, определяющих продолжительность обслуживания разными приборами заявок из разных входящих потоков.

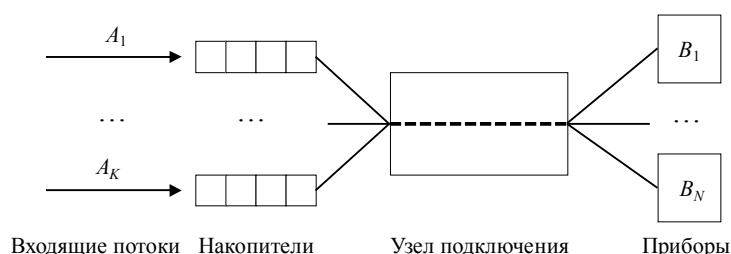


Рис. 1. Модель циклической системы

Порядок, способ и длительность подключения приборов к накопителям определяется заданными параметрами, которые в дальнейшем будем называть параметрами подключения. В случае, когда количество серверов $N > 1$, возможны различные варианты их «группового» поведения. В качестве основных групповых дисциплин выделим три:

1. *Жесткая группа.* Все обслуживающие приборы системы образуют одну группу обслуживания: все приборы в один момент времени работают только с одним накопителем, переключение между накопителями и пребывание в состоянии переключения (см. далее) для всех приборов производится синхронно.

2. *Одиночные приборы.* Обслуживающие приборы подключаются к накопителям независимо друг от друга, причем с одним накопителем в один момент времени может работать только один прибор. В случае, если по окончании сеанса подключения прибора к накопителю следующий накопитель уже обслуживается другим сервером, то прибор ищет следующий по циклическому порядку свободный (не находящийся в сеансе подключения с другим сервером) накопитель и подключается к нему. Эта дисциплина подключения доступна только при условии, что количество серверов не превышает количество накопителей ($N \leq K$).

3. *Гибкие группы.* Обслуживающие приборы подключаются к накопителям независимо друг от друга, в один момент времени к накопителю может быть подключено любое число приборов. По окончании сеанса подключения прибор подключается к следующему накопителю независимо от того, обслуживается он другими серверами или нет.

Кроме дисциплины подключения групп важными параметрами подключения являются параметры, определяющие продолжительность одного сеанса подключения сервера или группы серверов к накопителю. В данной работе будем называть их дисциплиной подключения и рассмотрим два типа:

1. *С разделением времени.* Продолжительность каждого подключения определяется заданной функцией распределения длительности сеанса. По окончании этого периода обслуживание всех заявок, производившееся в рамках данного подключения, прерывается, а сами заявки возвращаются в накопитель (в начало очереди, в порядке времени их поступления в систему). Продолжительность сеанса подключения может определяться одной функцией распределения для всех накопителей либо задаваться индивидуально для каждого накопителя.

2. *До полного исчерпания.* Подключение сохраняется до тех пор, пока не будет закончено обслуживание последней заявки в накопителе. При этом в случае жесткой группы все незадействованные приборы простаивают, пока последний из них не закончит обслуживание, но во время простоя могут принимать и обслуживать заявки, поступающие из текущего входящего потока. В случае гибких групп, если по окончании обслуживания на одном из приборов накопитель оказывается пуст, то сеанс подключения для данного прибора заканчивается и он переключается на другой накопитель.

Когда прибор или группа приборов завершают обслуживание одного накопителя и должны переключиться к следующему, то параметрами задачи может быть задано, что это переключение требует некоторого времени. В случае применения дисциплины подключения «до полного исчерпания» это время должно быть обязательно больше нуля, так как в противном случае, если все накопители оказались временно пустыми, модель перейдет в состояние «зацикливания», когда приборы мгновенно переключаются от очереди к очереди, а модельное время не изменяется. Для дисциплины с разделением времени этот параметр опционален. Если параметрами задачи указано, что переключение требует времени, то задается функция распределения, определяющая длительность состояния переключения, в которое прибор или группа приборов переходят по окончании сеанса подключения к одному накопителю. В этом состоянии приборы не выполняют никаких действий. По окончании времени переключения приборы подключаются к следующему накопителю. Функция распределения длительности времени переключения может определяться одной функцией распределения для всех накопителей либо задаваться индивидуально для каждого накопителя.

2. Объектная модель циклической системы обслуживания

2.1. Основные объекты циклической системы

Для имитационного моделирования стохастических систем наиболее подходящим методом моделирования является дискретно-событийный подход [12, 13], основная идея которого заключается в определении событий, которые должны произойти в системе, и сдвиге модельного времени по моментам наступления этих событий. В настоящей работе в качестве платформы для реализации приложения имитационного моделирования циклических систем будут использованы библиотеки программного комплекса ODIS [14–16], предназначенного для имитационного моделирования сетей массового обслуживания. В этих библиотеках уже имеется базовый класс `SimulationModel`, который реализует основной цикл дискретно-событийного моделирования систем массового обслуживания. Исполнение цикла моделирования базируется на взаимодействии объектов событий, заявок и элементов системы.

Объектная модель разрабатываемого приложения имитационного моделирования циклических СМО представлена на рис. 2, 3. Реализация механизма сеансов подключения и описание основных сценариев имитационного моделирования представлены в следующих подразделах.

Классы `Element`, `Source`, `PassiveBuffer` реализованы каркасом ODIS и предназначены для моделирования элементов СМО общего вида. Классы `InputSource`, `Server` и `ConnectionManager` реализуют объекты, специфичные для циклических СМО.

Класс `Element` представляет собой абстрактный класс, потомками которого будут любые элементы системы (например, входящий поток, накопитель, прибор обслуживания), которые могут принимать заявки с помощью операции `accept(...)`, а также генерировать и / или обрабатывать события. Обработка событий производится с помощью операции `processEvent(...)`.

Класс `Source` представляет входящий поток заявок. В методе `accept(...)` генерируется исключение, так как этот элемент не может принимать заявки. А в методе `processEvent(...)` генерируются новые заявки.

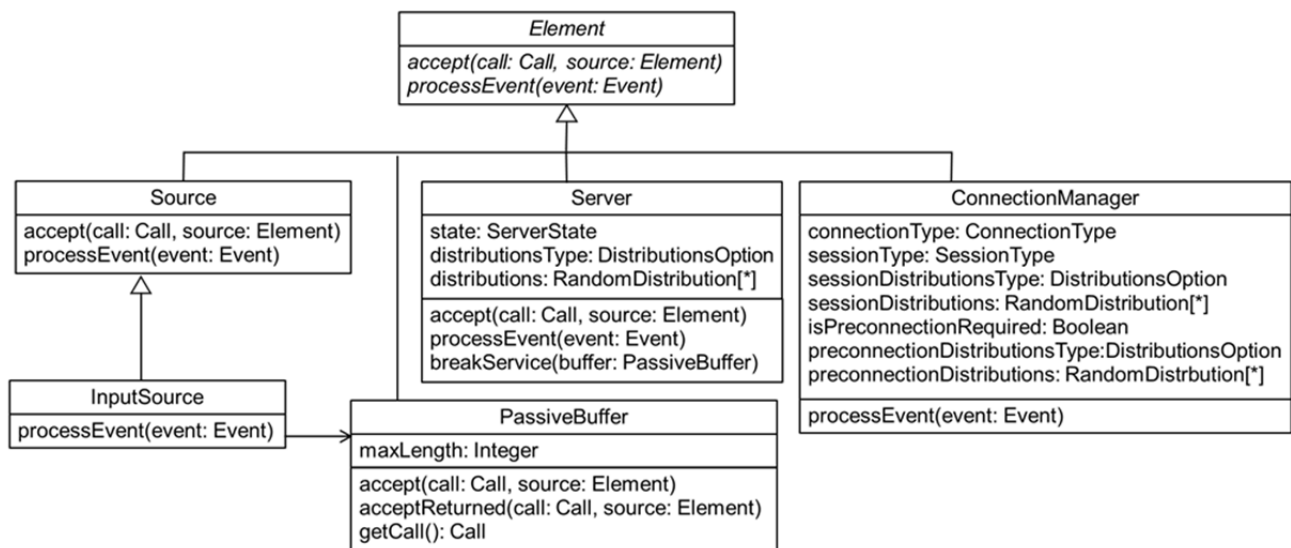


Рис. 2. Объектная модель циклической системы обслуживания

Класс `InputSource`, в отличие от `Source`, включает в себя накопитель заявок (объект стандартного класса `PassiveBuffer`). Метод `processEvent(...)` вызывается автоматически каркасом платформы, если обрабатываемое системой событие сгенерировано данным объектом `InputSource`. В дальнейшем объекты класса `InputSource` будем называть источниками.

Класс `PassiveBuffer` представляет собой реализацию объекта-накопителя заявок. Он имеет атрибут `maxLength`, который определяет максимальный объем накопителя (максимальное число заявок, которые можно в нем разместить). Значение `maxLength = -1` означает, что накопитель не ограничен в размерах. Методы `accept(...)` и `acceptReturned(...)` используются для помещения заявки в накопитель. Если при вызове метода `accept(...)` оказалось, что накопитель заполнен до предела, то заявка получает отказ и немедленно покидает систему. В случае неограниченного объема накопителя или при вызове метода `acceptReturned(...)` заявка будет помещена в накопитель в обязательном порядке. В этом случае считается, что накопитель всегда имеет небольшой резерв для размещения заявок, возвращенных в него в экстренных ситуациях (прерывание обслуживания).

Все заявки в накопителе сортируются в порядке времени их первого поступления в накопитель (времени поступления в систему). Операция `getCall(...)` извлекает самую старую заявку из накопителя и возвращает ее в качестве результата.

Объект `Server` реализует обслуживающий прибор и имеет два существенных отличия по сравнению с соответствующим объектом платформы, а именно:

- 1) может пребывать в трех состояниях (поле `state`): «Свободен», «Занят обслуживанием», «В режиме переключения»;
- 2) реализует возможность задания отдельных функций распределения для длительности обслуживания заявок из разных накопителей (входящих потоков). Для этого используется опция `distributionsType` со значениями `Single` (одна функция распределения для всех накопителей) и `ForEachBuffer` (функции распределения вероятностей заданы для каждого накопителя в отдельности), а также массив `distributions`.

Методы `accept(...)` и `processEvent(...)` используются соответственно для помещения заявки на обслуживание в данном приборе и для обработки события окончания обслуживания. Метод `breakService(...)` немедленно прерывает обслуживание заявки и возвращает ее в соответствующий накопитель с помощью метода `PassiveBuffer.acceptReturned(...)`.

Класс `ConnectionManager` (диспетчер подключений) – искусственно введенный объект, который хранит информацию и реализует логику подключения приборов к очередям. Диспетчер подключений содержит такие атрибуты, как тип соединения `connectionType`, дисциплина обслуживания очередей `sessionType`, способ задания функций распределения длительности подключения `sessionDistributionsType`, функции распределения длительности подключения приборов к накопителям `sessionDistributions`, флаг

переключения приборов `isPreconnectionRequired`, способ задания функций распределения длительности времени переключения `preconnectionDistributionsType`, а также функции распределения длительности переключения приборов `preconnectionDistributions`.

Метод `processEvent(...)` класса `ConnectionManager` обрабатывает событие окончания времени переключения прибора. Другие методы данного класса будут описаны ниже.

2.2. Механизм сеансов подключения

В объектной модели все серверы системы собраны в одну коллекцию – пул серверов (объект класса `ServersPool`). Все входящие потоки с накопителями также образуют коллекцию. У модели имеется единственный объект класса `ConnectionManager`, которому предоставлен полный доступ к обеим коллекциям.

Для организации моделирования подключений приборов к накопителям предлагается использовать механизм сеансов подключения. Сеанс подключения (объект `Session`) – это объект, создаваемый на время подключения пула приборов к одному накопителю (рис. 3). Данный объект отвечает за продолжительность подключения, управляет движением поступающих заявок, а также заявок, находящихся в накопителе и на обслуживании.

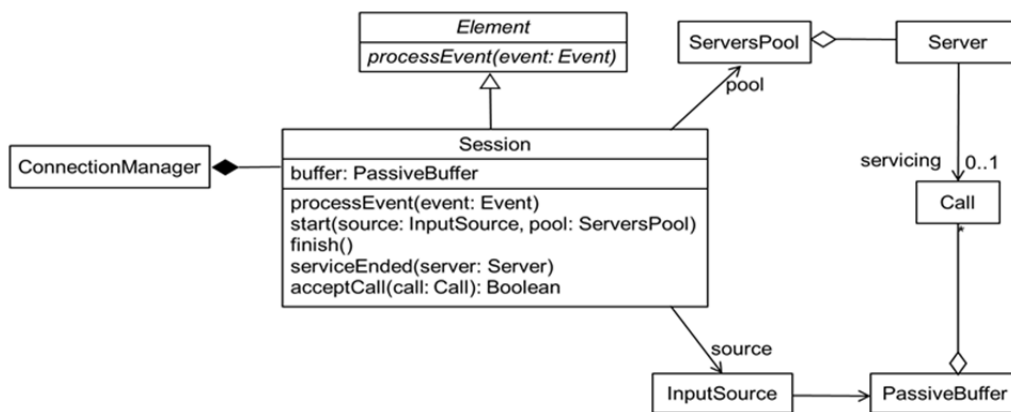


Рис. 3. Структурная схема механизма сеансов подключения

Объект `Session` реализует операцию `processEvent(...)` интерфейса `Element`, в ней обрабатывается событие окончания сеанса. Операции `start(...)` и `finish()` используются соответственно для старта и окончания сеанса. Метод `serviceEnded(...)` вызывается, когда обслуживающий прибор, находящийся в сеансе, закончил обслуживание и готов принять на обслуживание новую заявку. Операция `acceptCall(...)` используется объектами-источниками `InputSource` во время поступления заявки в систему для поиска сеанса со свободным прибором и передачи ему этой заявки. Атрибут `buffer` ссылается на объект-накопитель источника `source`.

Управление сеансами полностью осуществляет объект `ConnectionManager`. Для этого он реализует методы (см. рис. 3) `startNewSession(...)` и `sessionEnded(...)` соответственно для создания сеанса и обработки события его окончания, а также метод `startPreconnection(...)` для создания события отложенного старта сеанса (так называемое время переключения).

2.3. Основные сценарии имитационного моделирования циклической системы

Поскольку вся система построена на дискретно-событийной модели управления, то для выполнения полного цикла имитационного моделирования достаточно выделить типы событий системы, реализовать механизмы их генерации, записи в журнал событий, извлечения из него и процедуры обработки этих событий.

Выделим основные типы системных событий:

1. Начало моделирования. Происходит однократно на старте всего процесса моделирования. Не требует записи в журнал.
2. Поступление заявки в систему в определенном входящем потоке.
3. Окончание времени переключения прибора или группы приборов (если задано, что переключение требует времени).
4. Окончание обслуживания заявки на приборе.
5. Завершение сеанса (для дисциплины подключения «с разделением времени»).

Начало моделирования выполняется в методе `onInitialization` класса `PollingModel` (рис. 4). Здесь создаются по одному событию на каждый источник заявок. Затем в методе инициализации диспетчера подключений производятся необходимые подготовительные действия в зависимости от заданных параметров моделируемой СМО.

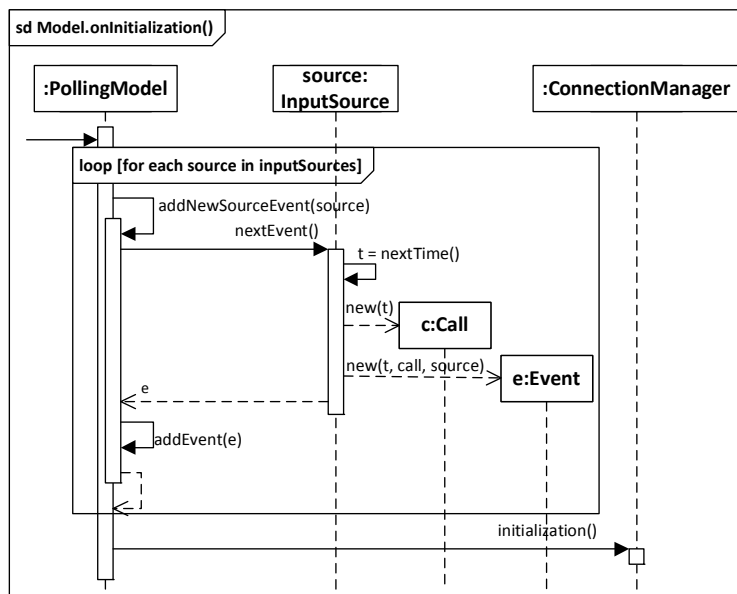


Рис. 4. Сценарий начала моделирования

Событие поступления заявки в систему обрабатывается в методе `processEvent(...)` класса `InputSource` (рис. 5). Если источник заявок подключен хотя бы к одному сеансу и накопитель пуст, находим сеанс со свободным прибором и помещаем заявку в этот прибор для обслуживания. Если хотя бы одно из условий нарушено, заявка просто помещается в накопитель. После этого генерируем следующее событие поступления заявки из входящего потока (на диаграмме этот шаг опущен – он выполняется базовым классом `Source`).

Событие окончания времени переключения группы приборов обрабатывается в методе `processEvent(...)` класса `ConnectionManager`. Сначала нужно создать новый сеанс подключения. Алгоритм создания и старта нового сеанса представлен на рис. 6. Во-первых, находим следующий по порядку источник заявок в соответствии с выбранной групповой политикой. Далее создаем новый сеанс и запускаем его. Если выбрана дисциплина подключения «до полного исчерпания» и в накопителе источника отсутствуют заявки, то сеанс подключения тут же прекращается. Если выбрана дисциплина подключения «с разделением времени», то генерируем и регистрируем событие окончания сеанса. Затем все свободные приборы заполняются имеющимися в накопителе заявками.

Событие окончания обслуживания заявки на приборе обрабатывается в методе `processEvent(...)` класса `Server` (рис. 7). Обработанная заявка удаляется из прибора, сеансу подключения сообщается об окончании обслуживания. Если в накопителе есть заявка, то она помещается на прибор, иначе если выбрана дисциплина подключения «до полного исчерпания» и ни один прибор пула серверов не занят, то сеанс завершается.

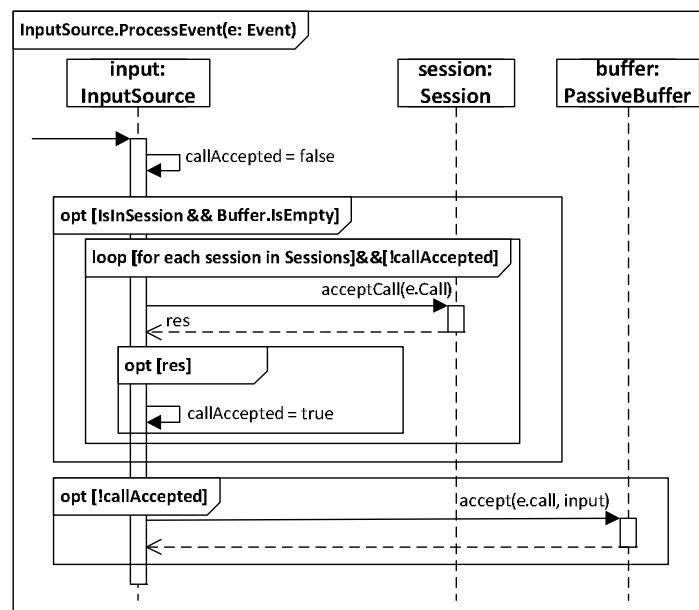


Рис. 5. Обработка события поступления заявки в систему

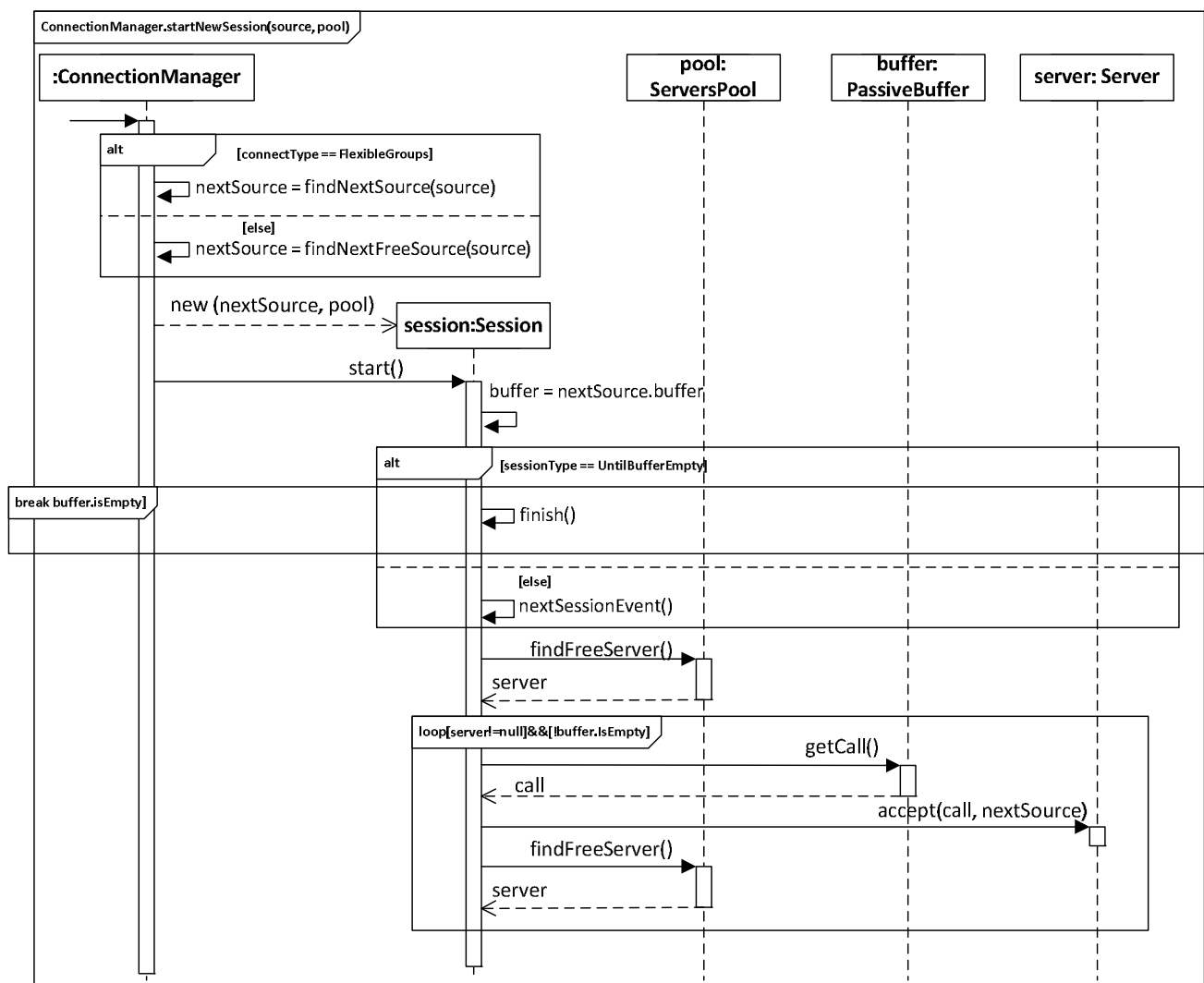


Рис. 6. Алгоритм старта нового сеанса

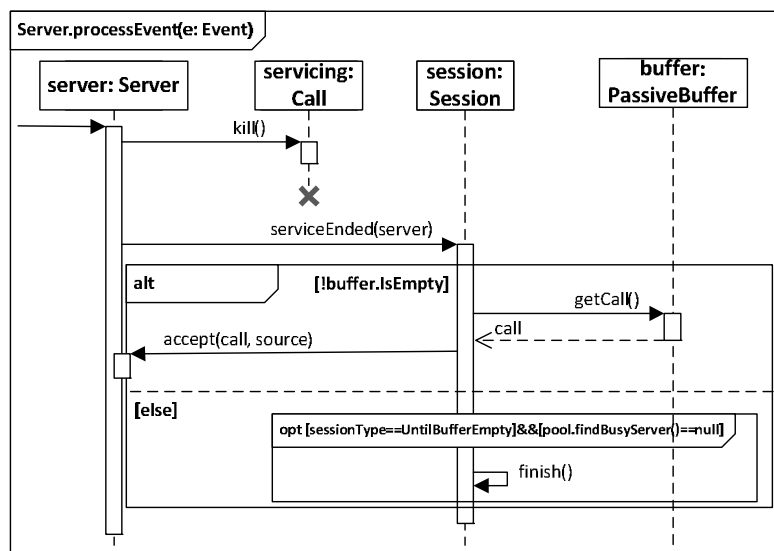


Рис. 7. Обработка события окончания обслуживания заявки

Событие завершения сеанса (для дисциплины подключения «с разделением времени») обрабатывается в методе processEvent(...) класса Session (рис. 8).

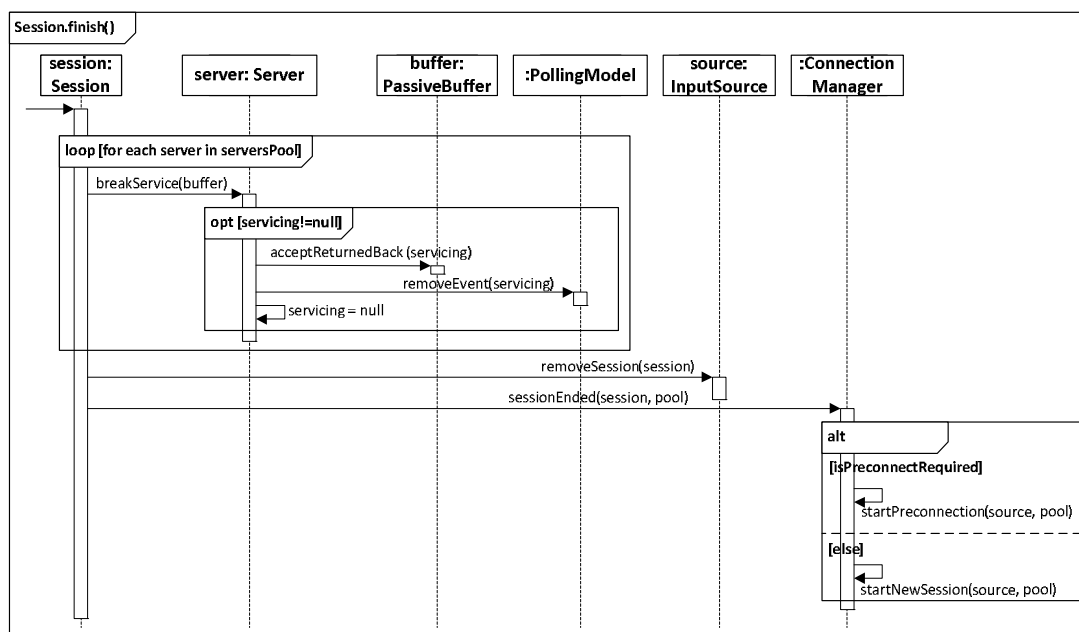


Рис. 8. Обработка события завершения сеанса

Сначала удаляются все признаки сеанса у источника и приборов в пуле. Если какие-то заявки находятся на обслуживании, то их обслуживание прерывается и заявки принудительно возвращаются в накопитель. Затем диспетчеру подключений сообщается об окончании сеанса. Он, в свою очередь, если требуется, переводит пул приборов в режим переключения, регистрируя событие окончания времени переключения. Если времени переключения не требуется, то для текущего пула приборов начинается новый сеанс.

Заключение

Разработанная и представленная выше объектная модель позволяет выполнять имитационное моделирование циклических система массового обслуживания с применением дискретно-событийного

подхода. На основе представленной модели было реализовано приложение, позволяющее выполнять имитационное моделирование для широкого класса циклических систем обслуживания за счет вариативности задаваемых параметров конфигурации: различные типы входящих потоков и их параметры для каждого источника, индивидуальные параметры соответствующих им накопителей, индивидуальные параметры для каждого обслуживающего прибора, в том числе в зависимости от обслуживаемого источника, учет возможных затрат времени на переключение приборов от одного источника к другому, три дисциплины подключения групп приборов, две стратегии, определяющие продолжительность сеансов, в том числе для дисциплины «с разделением времени»; реализована возможность задавать продолжительность сеанса подключения в зависимости от обслуживаемого источника.

Разработанное приложение может использоваться для исследования широкого класса циклических систем обслуживания, в том числе и тогда, когда аналитические исследования не представляются возможными.

ЛИТЕРАТУРА

1. Sztrik J. Basic queueing theory. University of Debrecen, 2011. 193 p.
2. Cooper R.B. Introduction to queueing theory, 2nd ed. New York : Elsevier North Holland, 1981. 347 p.
3. Takagi H. Analysis and applications of polling models // Lecture notes in computer science. 2000. V. 1769. P. 423–442.
4. Borst S.C. Polling systems. Amsterdam : Centrum voor wiskunde en informatica, 1996. 232 p.
5. Вишневский В.М., Семенова О.В. Математические методы исследования систем поллинга // Автоматика и телемеханика. 2006. № 2. С. 3–56.
6. Лой А., Кельтон В. Имитационное моделирование. 3-е изд. СПб. : Питер, 2004. 848 с.
7. Advances in intelligent modelling and simulation: simulation tools and applications / A. Byrski, Z. Oplatková, M. Carvalho, M. Kisiel-Dorohinicki. Springer, 2012. 368 p.
8. GPSS world. URL: <http://www.minutemansoftware.com/simulation.htm> (дата обращения: 27.01.2017).
9. Schriber T.J. Simulation using GPSS. New York : Wiley, 1974. 592 p.
10. Моделирование и симуляция динамических систем для Simulink. URL: <http://matlab.ru/products/simulink> (дата обращения: 27.01.2017).
11. Инструмент многоподходного имитационного моделирования AnyLogic. URL: <http://www.anylogic.ru> (дата обращения: 27.01.2017).
12. Robinson S. Simulation: the practice of model development and use. Hoboken : Wiley, 2004. 336 p.
13. Use cases of discrete event simulation: appliance and research / ed. by S. Bangsow. Springer, 2012. 373 p.
14. Моисеев А.Н. Программная система имитационного моделирования сетей массового обслуживания // Хроники объединенного фонда электронных ресурсов «Наука и образование». 2015. № 11 (78). С. 34.
15. Мещеряков Р.В., Моисеев А.Н., Демин А.Ю., Дорофеев В.А., Матвеев С.А. Применение параллельных вычислений в имитационном моделировании сетей массового обслуживания // Известия Томского политехнического университета. 2014. Т. 325, № 5. С. 99–109.
16. Моисеев А.Н., Сияков М.В. Разработка объектно-ориентированной модели системы имитационного моделирования процессов массового обслуживания // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2010. № 1 (10). С. 89–93.

Сонькин Михаил Аркадьевич, д-р техн. наук, доцент. E-mail: sonkin@tpu.ru

Национальный исследовательский Томский политехнический университет

Моисеев Александр Николаевич, д-р физ.-мат. наук, доцент. E-mail: moiseev.tsu@gmail.com

Национальный исследовательский Томский государственный университет

Сонькин Дмитрий Михайлович, канд. техн. наук, доцент. E-mail: sonkind@tpu.ru

Национальный исследовательский Томский политехнический университет

Буртова Дарья Алексеевна. E-mail: gottok.inbox@gmail.com

Национальный исследовательский Томский государственный университет

Поступила в редакцию 1 февраля 2017 г.

Sonkin Mikhail A., Moiseev Alexander N., Sonkin Dmitriy M., Burtovaya Daria A. (Tomsk State University, Tomsk Polytechnic University, Russian Federation).

Object model of application for simulation of cyclic queueing systems.

Key words: cyclic queueing system; simulation; object-oriented design.

DOI: 10.17223/19988605/40/8

In this paper, we consider the problem of developing the application for simulation of cyclic queueing systems. Cyclic (polling) systems is a class of queueing systems in which servers cyclically connect to various buffers (or inputs) to serve customers from these buff-

ers. In the study of cyclic systems, in most cases, analytical results cannot be obtained, therefore, the development and using software programs for the simulation of such systems is an important problem.

In the paper, the following model of cyclic queuing systems is considered. The system has $K \geq 1$ inputs where incoming customers arrive. All customers from the input number k go to a single buffer connected to this input. Also, the system has $N \geq 1$ servers that periodically connect to the buffers, take customers from them and serve the customers according to the specified service laws. When a service is complete, customers leave the system. In the case of the number of servers $N > 1$, different versions of their group behavior (disciplines) is possible. In the paper, we consider such group disciplines as a hard group, single devices and a flexible group. Also, there are disciplines of two types for connecting of separate or grouped servers to buffers: time-sharing or exhaustive. Moreover, when the server or group of servers have completed a service of one buffer and they need to switch to the next buffer, it maybe requires a certain switching time. In addition, in the case of the exhaustive discipline, this time should be greater than zero.

We use libraries of software package ODIS as a framework for the application implementation. This framework designed for simulation of queuing networks. The main classes of the framework that used in the work are the following: Element which is an abstract parent of all elements of the simulation model. Source which simulates arrivals of customers, Server which implements the service unit, PassiveBuffer which is a buffer with queue organized for customers in it, Call which implements a customer. In addition, the following special classes for simulation of cyclic systems is designed in the work: InputSource combines two objects – a Source instance and a PassiveBuffer one; ConnectionMannager is an object which implements the logic of connecting servers to buffers; Session is an object which is created at the instant when servers pool is connecting to a buffer, it responds for the duration of the connection and controls the movements of incoming customers, customers in the buffer and in servers.

Special technique of system events is designed to realize the simulation based on the discrete event model. This technique implements mechanisms of generation, logging and processing of system events. For the simulation of the cyclic systems, it is identified five types of the system events: start of simulation; customer arrival in a particular input; the end of switching time period (if it set that switching requires some time); customer service completion; the end of the session (for time-sharing connection discipline). Processing procedures for scenarios of all these events was designed and implemented. An application was implemented on the designed basis. The application is used for investigations in the field of queuing theory.

REFERENCES

1. Sztrik, J. (2011) *Basic Queueing Theory*. University of Debrecen.
2. Cooper, R.B. (1981) *Introduction to Queueing Theory*. 2nd ed. New York: Elsevier North Holland.
3. Takagi, H. (2000) Analysis and applications of polling models. *Lecture Notes in Computer Science*. 1769. pp.423–442. DOI: 10.1007/3-540-46506-5_18
4. Borst, S.C. (1996) *Polling systems*. Amsterdam: Centrum voor Wiskunde en Informatica.
5. Vishnevskiy, V.M. & Semenova, O.V. (2006) Mathematical methods to study the polling systems. *Automation and Remote Control*. 67(2). pp. 173–220. DOI: 10.1134/S0005117906020019
6. Law, A.M. & Kelton, W.D. (2000) *Simulation Modeling and Analysis*. McGraw-Hill.
7. Byrski, A., Oplatková, Z., Carvalho, M. & Kisiel-Dorohinicki, M. (eds) (2012) *Advances in intelligent modelling and simulation: simulation tools and applications*. Springer.
8. GPSS world. (n.d.) [Online] Available from: <http://www.minutemansoftware.com/simulation.htm>. (Accessed: 27th January 2017).
9. Schriber, T.J. (1974) *Simulation using GPSS*. New York: Wiley.
10. Matlab.ru (n.d.) *Modeling and Simulation of dynamic systems in Simulink*. [Online] Available from: <http://matlab.ru/products/simulink>. (Accessed: 27th January 2017).
11. Anylogic.ru (n.d.) *The tool of multi-approach simulation AnyLogic*. [Online] Available from: <http://www.anylogic.ru>. (Accessed: 27th January 2017).
12. Robinson, S. (2004) *Simulation: the practice of model development and use*. Hoboken: Wiley.
13. Bangsow, S. (ed.) (2012) *Use cases of discrete event simulation: appliance and research*. Springer.
14. Moiseev, A.N. (2015) Programmnyaya sistema imitatsionnogo modelirovaniya setey massovogo obsluzhivaniya [Software for simulation of queueing networks]. *Khroniki ob"edinen-nogo fonda elektronnykh resursov "Nauka i obrazovanie"*. 11(78). p. 34. (In Russian).
15. Meshcheryakov, R.V., Moiseev, A.N., Demin, A.Yu., Dorofeev, V.A. & Matveev, S.A. (2014) Using parallel computing in queueing network simulation. *Bulletin of Tomsk Polytechnic University*. 325(5). pp. 99–109. (In Russian).
16. Moiseev, A.N. & Sinyakov, M.V. (2010) Development of the object-oriented model of the software for simulation of queueing processes. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science*. 1(10). pp. 89–93. (In Russian).

М.К. Павликов**АЛГОРИТМ РАСПРЕДЕЛЕНИЯ НАГРУЗКИ В ПРОГРАММНОЙ СИСТЕМЕ,
ПОСТРОЕННОЙ НА ОСНОВЕ ПРОТОКОЛА HDP**

Предложен алгоритм эффективного распределения нагрузки в гетерогенной программной системе, построенной на основе протокола HDP. Данный алгоритм позволил получить равномерное распределение нагрузки в произвольный момент времени эксплуатации программной системы за счет использования данных обратной связи от запущенных узлов. Использование алгоритма позволило обеспечить отказоустойчивость системы и повысить толерантность к изменению аппаратного обеспечения. Даны описания алгоритма и его сравнение с другими алгоритмами распределения нагрузки в программной системе.

Ключевые слова: HDP; распределенная программная система; балансировка нагрузки.

На сегодняшний день большинство программных систем подвергаются внушительным функциональным нагрузкам. Порой система развивается настолько молниеносно, что наращивание мощности аппаратного обеспечения не позволяет адекватно справляться с возрастающей нагрузкой. А в случае высоконагруженных программных систем одного физического узла попросту не хватит, чтобы корректно обработать все поступающие запросы. Поэтому такие системы строятся, как правило, из одинаковых копий узлов, каждый из которых выполняет одинаковые функции, что приводит к возникновению задач маршрутизации и равномерного распределения нагрузки между существующими однотипными узлами программной системы. Также остро стоит вопрос о возможности поддержки: ввод нового узла или вывод старого узла из эксплуатации без полного выключения системы. Все эти задачи ложатся на методы и алгоритмы балансировки, но большинство таких методов и алгоритмов при распределении нагрузки не учитывают реальную ситуацию в текущий момент времени: при изменении набора узлов в них вносятся изменения, а дальше система функционирует без стороннего вмешательства.

Важнейшим фактором, приводящим к появлению копий одних и тех же узлов, является и потребность отсутствия единой точки отказа. Система должна быть устойчива к выходу из строя одного или нескольких узлов и продолжать корректно реагировать на поступающие запросы. Такие системы могут быть построены на основе протокола взаимодействия HDP, в котором реализован алгоритм балансировки нагрузки с учетом данных обратной связи, который позволяет в режиме реального времени менять нагрузку на тот или иной узел, чтобы поддерживать оптимальную нагрузку на программную систему в целом.

1. Алгоритмы распределения нагрузки в распределенных системах

Балансировка нагрузки – это метод распределения запросов между несколькими одинаковыми узлами распределенной вычислительной системы с целью оптимизации использования всех узлов совместно. Существуют различные алгоритмы балансировки нагрузки, которые имеют разное применение в различных областях, их целью также является обеспечение отказоустойчивости системы в целом.

Алгоритм round robin представляет из себя циклический перебор всех узлов по кругу, каждый следующий запрос отправляется на следующий в наборе узел. Главное достоинство – это простота алгоритма, что позволяет использовать его во многих системах, и архитектура таких систем остается максимально простой, так как эксплуатационные особенности системы не вносят дополнительного вклада в сложность архитектуры. Из недостатков алгоритма стоит выделить неравномерное распределение нагрузки, если узлы сети имеют разные технические характеристики. Технически более мощный узел будет лучше справляться с запросами, чем остальные, которые будут более нагружены. Также не реше-

на проблема недоступного узла, т.е. узла, который во время эксплуатации системы внезапно вышел из строя.

Алгоритм *weighted round robin* представляет собой модификацию алгоритма *round robin*, в которой каждому узлу вручную назначается некий вес – вероятность, с которой балансировщик нагрузки в следующий раз выберет именно этот узел. По сравнению с *round robin* в данном алгоритме удалось компенсировать возможные технические отличия узлов сети. Из недостатков алгоритма стоит выделить отсутствие решения всех задач отказоустойчивости, ведь если узел сети с высоким приоритетом перестает быть доступным, на него все равно отправляется большинство запросов.

2. Алгоритм распределения нагрузки *feedback node* в программной системе, использующей протокол HDP для взаимодействия узлов

Протокол HDP представляет собой протокол для обмена гетерогенными данными в распределенной вычислительной среде. Протокол использует жестко структурированные данные в формате JSON со специальными служебными метками согласно спецификации протокола. Согласно спецификации протокола все взаимодействующие узлы в HDP делятся на два типа: клиент и поставщик. Клиент – узел, запрашивающий данные. Поставщик – узел, предоставляющий доступ к данным и функциям их обработки.

В протоколе заложена возможность указания нескольких узлов: однотипных копий одних и тех же сервисов для того, чтобы была возможность балансировать нагрузку между копиями, выдающими одинаковые данные и предоставляющими одинаковые функции над данными.

Предлагаемый автором алгоритм *feedback node* является алгоритмом распределения нагрузки в программной системе, построенной на основе протокола HDP. Он учитывает гетерогенные данные о текущей загрузке каждого узла и на основе этих данных распределяет между ними нагрузку. На отслеживаемый узел устанавливается агент, который передает серверу данные об общей загрузке, загрузке процессора и использовании памяти на данном узле. Балансировщик HDP на основе этих данных принимает решение о передаче запроса тому или иному узлу в зависимости от его загруженности в текущий момент времени.

Среди выбранных отслеживаемых метрик используются:

- *LA* (*Load Average*) – среднее значение загрузки системы за период в 1 минуту. Значение *LA* строится на основе процессов, стоящих в очереди ожидания ресурсов, поэтому данное значение трактуется в зависимости от количества ядер процессора: значением по данной метрике является отношение количества ожидающих процессов к общему количеству ядер;

- *CPU* (*CPU utilization*) – загрузка процессора в текущий момент времени, выраженная в процентах;

- *MU* (*Memory usage*) – использование памяти, выраженное в процентах относительно общего количества.

Стоит отметить, что алгоритм может подстраиваться к исключению указанных выше метрик или добавлению новых.

В любой момент времени на сервере поддерживается в актуальном состоянии список приоритетных узлов, в котором перечислены адреса всех узлов в определенном порядке. Один и тот же узел может быть указан в списке многократно. Балансировщик нагрузки обходит этот список по кругу и передает каждый следующий запрос последовательно следующему узлу, указанному в списке.

У этого списка есть предельное время жизни (в секундах), в течение которого список не обновляется и признается актуальным. Предельное время жизни определяется на предположении, что за определенный промежуток времени значения отслеживаемых метрик кардинальным образом не изменятся, поэтому не имеет смысла нагружать узлы лишним получением метрик, тем самым повышая нагрузку на узел.

Алгоритм распределения нагрузки *feedback node*:

1. Присвоить N количество всех доступных однотипных узлов в программной системе.
2. Отправить запросы получения текущих значений *LA*, *CPU*, *MU* всем узлам с установленными агентами.

3.1. Сравнить значения параметра LA у всех узлов.

3.1.1. Если значение параметра LA хотя бы у 2 узлов отличается, то перейти к пункту 3.1.2., иначе перейти к пункту 3.2.

3.1.2. Вычислить SLA – сумму значений всех параметров:

$$SLA = \sum_{i=1}^N LA_i.$$

3.1.3. Сформировать таблицу «Узел-Значение», где «Значение» вычислить по следующей формуле для каждого узла из списка:

$$SLA_i = LA_i \frac{SLA}{100}.$$

3.1.4. Инвертировать столбец «Значение», перейти к пункту 4.

3.2. Сравнить значения параметра CPU у всех узлов.

3.2.1. Если значение параметра CPU хотя бы у 2 узлов отличается, то перейти к пункту 3.2.2., иначе перейти к пункту 3.3.

3.2.2. Вычислить $SCPU$ – сумму значений всех параметров:

$$SCPU = \sum_{i=1}^N CPU_i.$$

3.2.3. Сформировать таблицу «Узел-Значение», где «Значение» вычислить по следующей формуле для каждого узла из списка:

$$ACPU_i = CPU_i \frac{SCPU}{100}.$$

3.2.4. Инвертировать столбец «Значение», перейти к пункту 4.

3.3. Сравнить значения параметра MU у всех узлов.

3.3.1. Если значение параметра MU хотя бы у 2 узлов отличается, то перейти к пункту 3.3.2., иначе составить таблицу «Узел-Значение», в которой каждому узлу поставить 1 в «Значение» и перейти к пункту 4.

3.3.2. Вычислить SMU – сумму значений всех параметров:

$$SMU = \sum_{i=1}^N MU_i.$$

3.3.3. Сформировать таблицу «Узел-Значение», где «Значение» вычислить по следующей формуле для каждого узла из списка:

$$AMU_i = MU_i \frac{SMU}{100}.$$

3.3.4. Инвертировать столбец «Значение», перейти к пункту 4.

4. Сформировать список приоритетных узлов на основе таблицы «Узел-Значение» следующим образом: обойти таблицу и внести в список один и тот же узел столько раз, сколько указано в столбце «Значение».

5. Первый входящий запрос $R_{i=1}$ передать первому узлу из списка.

6. Входящий запрос R_i передать узлу R_i из списка.

7. При получении запроса R_{N+1} проверить, превышено ли предельное время жизни списка приоритетных узлов. Если время жизни списка было превышено, перейти к пункту 1, иначе присвоить $i = 1$ и перейти к пункту 5.

Заключение

В данной работе был проведен анализ проблемы оптимального распределения нагрузки в гетерогенной программной системе, указаны достоинства и недостатки существующих алгоритмов, предложен алгоритм распределения нагрузки, учитывающий обратные данные о текущем состоянии узлов и меняющий вектор нагрузки в режиме реального времени. Данный алгоритм позволяет эффективно ис-

пользовать имеющиеся аппаратные ресурсы в виде распределенных узлов, равномерная загрузка которых, в свою очередь, ведет к повышению отказоустойчивости системы в целом и уменьшению времени реагирования на поступающие входящие запросы. В будущем данный алгоритм достаточно гибко может подстраиваться под новые метрики и толерантен к включению или выключению учета отдельных метрик в произвольных системах. Это приводит к тому, что данный алгоритм можно применить в высоконагруженных распределенных программных системах, где крайне важно избежать единой точки отказа, обеспечить отказоустойчивость и сбор метрик не является критичным фактором, способным значительно увеличить общую нагрузку на программную систему.

ЛИТЕРАТУРА

1. Павликов М.К. Протокол HDP // Вестник компьютерных и информационных технологий. 2016. № 8. С. 52–56.
2. Richardson L., Amundsen M., Ruby S. RESTful Web APIs. O'Reilly Media, 2013. 404 p.
3. Bourke T. Server Load Balancing. O'Reilly Media, 2011. 194 p.
4. Koppurapu C. Load Balancing Servers, Firewalls, and Caches. Wiley, 2012. 224 p.

Павликов Максим Константинович. E-mail: severemax@yandex.ru
Московский авиационный институт

Поступила в редакцию 6 февраля 2017 г.

Pavlikov Maxim K. (Moscow Aviation Institute, Russian Federation).

Load balancing algorithm in a distributed programming system, built on the basis of the HDP protocol.

Keywords: HDP; distributed software system; load balancing.

DOI: 10.17223/19988605/40/9

In this paper, we discuss load balancing in a software system based on the HDP protocol. Load balancing is a method of distributing requests between several identical nodes of a distributed computing system in order to optimize the use of all nodes together. There are various load balancing algorithms that have different applications in different areas and their purpose is also to ensure the fault tolerance of the system as a whole.

The algorithm of round robin is a cyclic search of all nodes in a circle, each subsequent request is sent to the next node in the set. The main advantage is the simplicity of the algorithm, which makes it possible to use it in many systems and the architecture of such systems remains as simple as possible. The operational characteristics of the system do not add to the complexity of the architecture. From the disadvantages of the algorithm is to distinguish the uneven distribution of the load, if the nodes of the network have different technical characteristics. A technically more powerful node will better cope with requests than others, which will be more loaded. Also, the problem of an inaccessible node is not solved, i.e. which during the operation of the system suddenly went out of order.

The algorithm weighted round robin is a modification of the round robin algorithm, in which each node is manually assigned a certain weight - the probability with which the load balancer next time will choose this particular node. Compared with the round robin in this algorithm, it was possible to compensate for possible technical differences between the nodes of the network. From the disadvantages of the algorithm, it is worth highlighting the lack of a solution to all fault-tolerance problems, because if a network node with a high priority ceases to be available, queries are still sent to it.

The main problem with most balancing algorithms is that they do not use the state information of each node that is in the list from which the balancer selects the appropriate node each time based on a certain algorithm. The feedback node algorithm proposed by the author is an algorithm for load distribution in a distributed system based on the HDP protocol. This algorithm takes into account heterogeneous data about the current load of each node and, based on these data, distributes the load between them. An agent is installed on the monitored node, which transmits to the server the data on the overall load, the processor load, and the use of memory on this node. The HDP balancer uses this data to decide whether to send a request to a particular node, depending on its load at the current time.

REFERENCES

1. Pavlikov, M.K. (2016) HDP protocol. *Vestnik komp'yuternykh i informatsionnykh tekhnologiy – Herald of Computer and Information Technologies*. 8. pp. 52–56. (In Russian).
2. Richardson, L., Amundsen, M. & Ruby, S. (2013) *RESTful Web APIs*. O'Reilly Media. [Online] Available from: <https://www.oreilly.com/>
3. Bourke, T. (2011) *Server Load Balancing*. O'Reilly Media.
4. Koppurapu, C. (2012) *Load Balancing Servers, Firewalls, and Caches*. Wiley.

СВЕДЕНИЯ ОБ АВТОРАХ

БУРТОВАЯ Дарья Алексеевна – магистрант факультета информатики Национального исследовательского Томского государственного университета. E-mail: gottok.inbox@gmail.com

ГОРЦЕВ Александр Михайлович – профессор, доктор технических наук, заведующий кафедрой исследования операций факультета прикладной математики и кибернетики Национального исследовательского Томского государственного университета. E-mail: gam@mail.fpmk.tsu.ru

ДОМБРОВСКИЙ Владимир Валентинович – профессор, доктор технических наук, заведующий кафедрой информационных технологий и бизнес-аналитики Института экономики и менеджмента Национального исследовательского Томского государственного университета. E-mail: dombrovs@ef.tsu.ru

ЗАВГОРОДНЯЯ Мария Евгеньевна – кандидат технических наук, доцент кафедры исследования операций факультета прикладной математики и кибернетики Национального исследовательского Томского государственного университета. E-mail: mari.zavgor@mail.ru

КОСТИН Кирилл Константинович – аспирант кафедры прикладной математики факультета математики, механики и компьютерных наук Южно-Уральского государственного университета (национального исследовательского университета) (г. Челябинск). E-mail: lemwwwar@gmail.com

ЛУКИН Николай Алексеевич – кандидат технических наук, научный руководитель группы микропроцессорных архитектур Института машиноведения УрО РАН, заведующий лабораторией «Нанокomпьютер» ИРИТ-РТФ Уральского федерального университета (г. Екатеринбург). E-mail: n.a.lookin@urfu.ru

МАНСИМОВ Камил Байрамали оглы – доктор физико-математических наук, профессор, заведующий кафедрой математической кибернетики Бакинского государственного университета, руководитель лаборатории «Управление в сложных динамических системах» Института систем управления НАН Азербайджана (г. Баку, Азербайджан). E-mail: kmansimov@mail.ru

МЕДВЕДЕВ Геннадий Алексеевич – профессор, доктор физико-математических наук, профессор факультета прикладной математики и информатики Белорусского государственного университета (г. Минск, Беларусь). E-mail: MedvedevGA@bsu.by

МОИСЕЕВ Александр Николаевич – доцент, доктор физико-математических наук, доцент кафедры программной инженерии факультета прикладной математики и кибернетики Национального исследовательского Томского государственного университета. E-mail: moiseev.tsu@gmail.com

ОБЪЕДКО Татьяна Юрьевна – кандидат физико-математических наук, доцент кафедры информационных технологий и бизнес-аналитики Института экономики и менеджмента Национального исследовательского Томского государственного университета. E-mail: tatyana.obedko@mail.ru

ПАВЛИКОВ Максим Константинович – аспирант кафедры вычислительной математики и программирования Московского авиационного института. E-mail: severemax@yandex.ru

РАСУЛОВА Шахла Меджид кызы – аспирантка Института систем управления НАН Азербайджана (г. Баку, Азербайджан). E-mail: rasulzade_sh@yahoo.com

СМАГИН Валерий Иванович – профессор, доктор технических наук, профессор кафедры исследования операций факультета прикладной математики и кибернетики Национального исследовательского Томского государственного университета. E-mail: vsm@mail.tsu.ru

СОНЬКИН Дмитрий Михайлович – кандидат технических наук, заместитель директора по развитию Института кибернетики Национального исследовательского Томского политехнического университета. E-mail: sonkind@tpu.ru

СОНЬКИН Михаил Аркадьевич – доцент, доктор технических наук, профессор кафедры информационных систем и технологий Национального исследовательского Томского политехнического университета. E-mail: sonkin@tpu.ru

ТЫРСИН Александр Николаевич – доцент, доктор технических наук, заведующий кафедрой прикладной математики Уральского энергетического института Уральского федерального университета имени первого Президента России Б.Н. Ельцина (г. Екатеринбург). E-mail: at2001@yandex.ru

ФИЛИМОНОВ Александр Юрьевич – доцент кафедры информационных технологий Уральского федерального университета имени первого Президента России Б.Н. Ельцина (г. Екатеринбург). E-mail: af.1015@yahoo.com

Научный журнал

**ВЕСТНИК
ТОМСКОГО
ГОСУДАРСТВЕННОГО
УНИВЕРСИТЕТА**

**УПРАВЛЕНИЕ,
ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА
И ИНФОРМАТИКА**

**TOMSK STATE UNIVERSITY
JOURNAL OF CONTROL AND COMPUTER SCIENCE**

2017. № 40

Редактор Н.А. Афанасьева
Корректор Е.Г. Шумская
Оригинал-макет А.И. Лелоюр
Редакторы-переводчики: Г.М. Кошкин; В.Н. Горенинцева
Дизайн обложки Л.Д. Кривцовой

Подписано к печати 11.09.2017 г. Формат 60x84¹/₈.
Гарнитура Times. Печ. л. 10,7; усл. печ. л. 9,9.
Тираж 250 экз. Заказ № 2747. Цена свободная.

Дата выхода в свет 4.10.2017 г.

Журнал отпечатан на полиграфическом оборудовании
Издательского Дома Томского государственного университета
634050, г. Томск, Ленина, 36
Тел. 8(382-2)–52-98-49; 8(382-2)–53-15-28; 8(382-2)–52-96-75
Сайт: <http://publish.tsu.ru>; E-mail: rio.tsu@mail.ru