

**ВЕСТНИК
ТОМСКОГО
ГОСУДАРСТВЕННОГО
УНИВЕРСИТЕТА**

**УПРАВЛЕНИЕ,
ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА
И ИНФОРМАТИКА**

**TOMSK STATE UNIVERSITY
JOURNAL OF CONTROL AND COMPUTER SCIENCE**

Научный журнал

2019

№ 48

**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 Applied Mathematics Department Tomsk State University. Tel: +73822529599

Valery Smagin – Deputy Editor-in-Chief, Doctor of Sc., Prof. of the Applied Mathematics Department Tomsk State University. Tel: +73822529599

Lyudmila Nezhelskaya – Executive Editor, Doctor. of Sc., Prof. of the Applied Mathematics Department Tomsk State University. E-mail: vestnik_uvti@mail.tsu.ru

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

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

Anjela Matrosova – Doctor of Sc., Prof. 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 Semenkina – Doctor of Sc., Prof. System Analysis and Operations Research Department Reshetnev Siberian State University of Science and Technology (Krasnoyarsk, Russia)

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

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

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

Editorial address:

Institute of Applied Mathematics and Computer Science, unit of Applied Mathematics

National Research Tomsk State University

36 Lenina Avenue, Tomsk, 634050

Telephone / fax: +73822529599

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.
Ivannikov V.P. ISP RAS
Moscow, Russia

Yervant Zorian

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

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).

Учредитель – Томский государственный университет

РЕДАКЦИОННАЯ КОЛЛЕГИЯ

Горцев Александр Михайлович – гл. редактор, проф., д-р техн. наук, зав. кафедрой прикладной математики ТГУ. Тел. +73822529599

Смагин Валерий Иванович – зам. гл. редактора, проф., д-р техн. наук, проф. кафедры прикладной математики ТГУ. Тел. +73822529599

Нежелская Людмила Алексеевна – ответственный секретарь, доц., д-р физ.-мат. наук, проф. кафедры прикладной математики ТГУ
E-mail: vestnik_uvti@mail.tsu.ru

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

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

Коскин Геннадий Михайлович – проф., д-р физ.-мат. наук, проф. кафедры системного анализа и математического моделирования ТГУ

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

Матросова Анжела Юрьевна – проф., д-р техн. наук, проф. кафедры 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)

Семенин Евгений Станиславович – проф., д-р техн. наук, проф. каф. системного анализа и исследования операций, СибГУ им. акад. М.Ф. Решетнева (г. Красноярск)

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

Фархадов Маис 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, Институт прикладной математики и компьютерных наук,

отделение прикладной математики

Телефон / факс: +73822529599

E-mail: vestnik_uvti@mail.tsu.ru

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

д-р, проф.
Технологический университет,
Таллин, Эстония

Рейндерт Нобель

д-р, доцент
Свободный университет,
Амстердам, Нидерланды

Нина Евтушенко

д-р техн. наук, проф.
ИСП РАН им. Иваницова В.П.,
Москва, Россия

Ервант Зорян

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

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., Пашинская Т.Ю.
Оптимальные стратегии прогнозирующего управления системами со случайными параметрами, описываемыми многомерной регрессионной моделью с марковским переключением режимов 4

ОБРАБОТКА ИНФОРМАЦИИ

Назаров А.А., Пауль С.В., Лизюра О.Д.
Асимптотический анализ RQ-системы с N типами вызываемых заявок в предельном условии большой задержки заявок на орбите 13

Нежелская Л.А., Сидорова Е.Ф.
Оценка длительности непродлевающегося мертвого времени в коррелированном обобщенном синхронном потоке второго порядка 21

Rouban A.I.
The sensitivity functionals in the Bolts's problem for multivariate dynamic systems described by integro-differential equations with delay time 31

Рудько И.М.
Обнаружение подвижного объекта с использованием порядковых статистик 42

Цициашвили Г.Ш., Осипова М.А.
Исследование процесса сборки пуассоновских потоков 51

ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ

Приступа А.В., Лапатин И.Л., Замятин А.В.
Применение глобальных дескрипторов формы для идентификации 3D объектов 57

Filimonov A.Y., Trishin V.N.
Model-oriented co-design of homogeneous computing systems 67

Черемисин Д.И., Черемисина Л.Д.
Поиск подсхем в КМОП схеме из транзисторов методом Graph Mining 74

ДИСКРЕТНЫЕ ФУНКЦИИ И АВТОМАТЫ

Широкова Е.В., Прокопенко С.А., Шабалдина Н.В.
К построению параллельной композиции расширенных автоматов 83

**ПРОЕКТИРОВАНИЕ И ДИАГНОСТИКА
ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ**

Буторина Н.Б., Пахомова Е.Г.
О длинах путей самотестируемых детекторов, построенных в базе ПЛБ 92

Мосин С.Г.
Оптимизация построения нейроморфного справочника неисправностей для тестирования и диагностики аналоговых интегральных схем 103

**МЕМУАРЫ, ПАМЯТНЫЕ ДАТЫ,
ПЕРСОНАЛИИ**

75 лет профессору
Анатолию Григорьевичу Дмитренко 112

Сведения об авторах 114

CONTENTS

**CONTROL
OF DYNAMICAL SYSTEMS**

Dombrovskii V.V., Pashinskaya T.Yu.
Optimal predictive control strategies for systems with random parameters described by multidimensional Markov switching regression model 4

DATA PROCESSING

Nazarov A.A., Paul S.V., Lizyura O.D.
Asymptotic analysis of retrieval queue with N types of outgoing calls under low rate of retrials condition 13

Nezhel'skaya L.A., Sidorova E.F.
Estimation of the unextendable dead time duration in correlated synchronous generalized flow of the second order 21

Rouban A.I.
The sensitivity functionals in the Bolts's problem for multivariate dynamic systems described by integro-differential equations with delay time 31

Rudko I.M.
Detection of moving object using order statistics 42

Tsitsiashvili G.Sh., Osipova M.A.
The study of the assembly of poisson flows 51

INFORMATICS AND PROGRAMMING

Pristupa A.V., Lapatin I.L., Zamyatin A.V.
3D object identification based on global shape descriptors 57

Filimonov A.Y., Trishin V.N.
Model-oriented co-design of homogeneous computing systems 67

Cheremisinov D.I., Cheremisinova L.D.
Subcircuits discovery in transistor level CMOS circuits using Graph Mining 74

DISCRETE FUNCTION AND AUTOMATION

Shirokova E.V., Prokopenko S.A., Shabaldina N.V.
On deriving the parallel composition of extended finite state machines 83

**DESIGNING AND DIAGNOSTICS
OF COMPUTER SYSTEMS**

Butorina N.B., Pakhomova E.G.
On the lengths of the paths in self-testing checkers based on CLB 92

Mosin S.G.
Optimization of constructing the neuromorphic fault dictionary for testing and diagnostics of analog ICs 103

**MEMOIRS, MEMORABLE DATE,
PERSONALITIS**

75 years to professor
Anatoly Grigorievich Dmitrenko 112

Brief information about the authors 114

УПРАВЛЕНИЕ ДИНАМИЧЕСКИМИ СИСТЕМАМИ

УДК 519.2

DOI: 10.17223/19988605/48/1

В.В. Домбровский, Т.Ю. Пашинская

ОПТИМАЛЬНЫЕ СТРАТЕГИИ ПРОГНОЗИРУЮЩЕГО УПРАВЛЕНИЯ СИСТЕМАМИ СО СЛУЧАЙНЫМИ ПАРАМЕТРАМИ, ОПИСЫВАЕМЫМИ МНОГОМЕРНОЙ РЕГРЕССИОННОЙ МОДЕЛЬЮ С МАРКОВСКИМ ПЕРЕКЛЮЧЕНИЕМ РЕЖИМОВ

Рассматривается класс дискретных стохастических систем с параметрами, эволюция которых описывается уравнением многомерной регрессии с марковскими скачками. Динамика экзогенных факторов описывается векторной авторегрессионной моделью с марковским переключением режимов порядка p (MS-VAR(p) модель). Синтезированы оптимальные стратегии прогнозирующего управления с учетом явных ограничений на управляющие переменные по обобщенному критерию, представляющему собой линейную комбинацию; а) ожидаемых значений квадратичных форм по состоянию и управлению; б) квадратичной формы ожидаемых значений состояний системы; с) линейной части – ожидаемого значения состояния системы.

Ключевые слова: стохастические системы; марковские скачки; многомерная модель регрессии; прогнозирующее управление; ограничения.

Моделями со случайными параметрами описывается широкий класс реальных динамических систем [1]. Одной из важных областей применения является финансовая инженерия, где такие модели используются для описания эволюции инвестиционного портфеля (см.: [2] и данный там обзор). Эффективным подходом к синтезу стратегий управления такими системами при ограничениях на состояния и / или управления является метод управления с прогнозированием, (прогнозирующее управление, управление с прогнозирующей моделью) [3, 4].

Прогнозирующему управлению дискретными системами, параметры которых изменяются в соответствии с эволюцией марковской цепи, посвящены работы [5–13].

В настоящей работе рассматривается класс дискретных стохастических систем с параметрами, эволюция которых описывается уравнением многомерной регрессии с марковскими скачками. Динамика экзогенных факторов описывается векторной авторегрессионной моделью с марковским переключением режимов порядка p (MS-VAR(p) модель [14]). Данный класс систем ранее в литературе не рассматривался. Синтезированы оптимальные стратегии прогнозирующего управления с учетом явных ограничений на управляющие переменные по обобщенному критерию, представляющему собой линейную комбинацию: а) ожидаемых значений квадратичных форм по состоянию и управлению; б) квадратичной формы ожидаемых значений состояний системы; с) линейной части – ожидаемого значения состояния системы. Изменяя весовые матрицы в обобщенном критерии, можно получать различные критерии управления: квадратичный критерий; критерий «mean-variance».

1. Постановка задачи

Пусть объект управления описывается уравнениями:

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

$$\eta(k+1) = \beta[\theta(k+1)]Y(k) + \lambda[\theta(k+1)]\omega(k+1), \quad (2)$$

$$Y(k+1) = \alpha[\theta(k+1)]Y(k) + \sigma[\theta(k+1)]W(k+1), \quad (3)$$

$$\alpha[\theta(k)] = \sum_{i=1}^v \theta_i(k)\alpha^{(i)}, \quad \lambda[\theta(k)] = \sum_{i=1}^v \theta_i(k)\lambda^{(i)}, \quad (4)$$

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

где $x(k) \in \mathbb{R}^{n_x}$ – вектор состояния, $u(k) \in \mathbb{R}^{n_u}$ – вектор управления, $\eta(k) \in \mathbb{R}^{n_\eta}$ – вектор случайных параметров, $Y(k) = [y^T(k), y^T(k-1), \dots, y^T(k-p+1)]^T_{n_y \times p}$, $y(k) \in \mathbb{R}^{n_y}$, $W(k) = [w^T(k), 0, 0, \dots, 0]^T_{n_y \times p}$,

$w(k) \in \mathbb{R}^{n_y}$, $\omega(k) \in \mathbb{R}^{n_\omega}$ – векторы белых шумов с нулевым средним и матрицами ковариаций $M\{w(k)w^T(k)\} = I_{n_y}$, $M\{\omega(k)\omega^T(k)\} = I_{n_\omega}$; I_{n_y}, I_{n_ω} – единичные матрицы размерностей n_y, n_ω соответственно; $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$, $\beta^{(i)} \in \mathbb{R}^{n_\eta \times n_y}$, $\lambda^{(i)} \in \mathbb{R}^{n_\eta \times n_\omega}$,

$$\alpha^{(i)} = \begin{bmatrix} \alpha_1^{(i)} & \alpha_2^{(i)} & \dots & \alpha_{p-1}^{(i)} & \alpha_p^{(i)} \\ I_{n_y} & 0 & \dots & 0 & 0 \\ 0 & I_{n_y} & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & I_{n_y} & 0 \end{bmatrix}_{n_y \times n_y \times p}, \quad \alpha_j^{(i)} \in \mathbb{R}^{n_y \times n_y},$$

$\sigma^{(i)} = \text{diag}\{\varphi^{(i)}, 0, \dots, 0\}$, $\varphi^{(i)} \in \mathbb{R}^{n_y \times n_y}$, $\theta_i(k+1)$ ($i = \overline{1, v}$) – компоненты вектора $\theta(k+1)$, $\theta(k) = [\delta(\tau(k), 1), \dots, \delta(\tau(k), v)]^T$, $\delta(\tau(k), j)$ – функция Кронекера; $\{\tau(k); k = 0, 1, 2, \dots\}$ – однородная дискретная марковская цепь с конечным множеством состояний $\{1, 2, \dots, v\}$, известной матрицей переходных вероятностей $P = [P_{ij}]$ и известным начальным распределением. Последовательности $w(k)$, $\omega(k)$ и $\theta(k)$ независимы. Матрица $B[\eta(k)]$ ($i = \overline{1, v}$) зависит от $\eta(k)$ линейно. Предполагается, что состояние марковской цепи в момент времени k доступно наблюдению. Уравнение (2) представляет собой уравнение множественной регрессии с экзогенными факторами $Y(k)$ и параметрами, зависящими от состояния цепи Маркова. Динамика экзогенных факторов $Y(k)$ описывается векторной авторегрессионной моделью с переключающимися режимами порядка p MS-VAR(p), представленной в виде процесса первого порядка вида (3) [14].

Вектор $\theta(k)$ допускает следующее представление в пространстве состояний [15]:

$$\theta(k+1) = P\theta(k) + v(k+1), \quad (5)$$

где $\{v(k)\}$ – последовательность мартингалов приращений.

На управляющие воздействия наложены ограничения:

$$u_{\min}(k) \leq S(k)u(k) \leq u_{\max}(k), \quad (6)$$

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

Для управления системой (1)–(4) используем методологию управления с прогнозирующей моделью. На каждом шаге k будем определять закон управления системой (1)–(4) при ограничениях (6) из условия минимума критерия со скользящим горизонтом управления:

$$J(k+m/k) = \sum_{i=1}^m M\{x^T(k+i)R_1(k+i)x(k+i) / x(k), \eta(k), Y(k), \theta(k)\} - \quad (7)$$

$$- \sum_{i=1}^m M\{x^T(k+i) / x(k), \eta(k), Y(k), \theta(k)\} R_2(k+i) M\{x(k+i) / x(k), \eta(k), Y(k), \theta(k)\} -$$

$$-\sum_{i=1}^m R_3(k+i)M\{x(k+i) / x(k), \eta(k), Y(k), \theta(k)\} + \\ + \sum_{i=0}^{m-1} M\{u^T(k+i/k)R(k+i)u(k+i/k) / x(k), \eta(k), Y(k), \theta(k)\},$$

по последовательности прогнозирующих управлений $U(k) = [u^T(k/k), \dots, u^T(k+m-1/k)]^T$, зависящих от состояния системы в момент времени k , m – горизонт прогноза, $R_1(k+i) \geq 0$, $R_2(k+i) \geq 0$, $R(k+i-1) > 0$ – весовые матрицы соответствующих размерностей, $R_3(k+i)$ – весовой вектор соответствующей размерности.

В качестве управления в момент времени k берем $u(k) = u(k/k)$. Тем самым получаем управление $u(k)$ как функцию состояний $\theta(k)$, $x(k)$, $\eta(k)$ и $Y(k)$, 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.

Изменяя весовые матрицы $R_1(k+i)$, $R_2(k+i)$, $R_3(k+i)$ в выражении (7), можно получать различные критерии управления системой (1)–(4).

Задача 2.1. Полагая $R_2(k+i) = 0$, имеем задачу прогнозирующего управления по квадратичному критерию:

$$J(k+m/k) = \sum_{i=1}^m M\left\{x^T(k+i)R_1(k+i)x(k+i) - \right. \\ \left. - R_3(k+i)x(k+i) + u^T(k+i-1/k)R(k+i-1)u(k+i-1/k) / x(k), \eta(k), Y(k), \theta(k)\right\}.$$

Данный критерий представляет собой линейную комбинацию квадратичной и линейной частей. При $R_3(k+i) = 0$ имеем классический квадратичный критерий.

Задача 2.2. Пусть скалярный выход системы (1)–(4)

$$z(k) = c(k)x(k),$$

где $c(k)$ – вектор соответствующей размерности. Полагая

$$R_1(k+i) = R_2(k+i) = \rho_1(k+i)c^T(k+i)c(k+i), \quad R_3(k+i) = \rho_2(k+i)c(k+i), \quad i = \overline{1, m},$$

где $\rho_1(k+i) \geq 0$, $\rho_2(k+i) \geq 0$ – скалярные величины, имеем задачу управления по критерию «mean-variance»:

$$J(k+m/k) = \sum_{i=1}^m \rho_1(k+i)M\left\{\left(z(k+i) - M\{z(k+i) / x(k), \eta(k), Y(k), \theta(k)\}\right)^2 / x(k), \eta(k), Y(k), \theta(k)\right\} - \\ - \sum_{i=1}^m \rho_2(k+i)M\{z(k+i) / x(k), \eta(k), Y(k), \theta(k)\} + \\ + \sum_{i=0}^{m-1} M\left\{u^T(k+i/k)R(k+i)u(k+i/k) / x(k), \eta(k), Y(k), \theta(k)\right\}.$$

Весовые коэффициенты $\rho_1(k+i)$, $\rho_2(k+i)$ характеризуют склонность к риску (risk-aversion) и задают соотношение между ожидаемым значением и вариацией выхода системы в момент времени $k+i$.

2. Синтез стратегий прогнозирующего управления

Рассмотрим следующие выражения:

$$J^{(1)}(k+m/k) = M\left\{\sum_{i=1}^m x^T(k+i)R_1(k+i)x(k+i) - R_3(k+i)x(k+i) + \right. \\ \left. + u^T(k+i-1/k)R(k+i-1)u(k+i-1/k) / x(k), \eta(k), Y(k), \theta(k)\right\}, \quad (8)$$

$$J^{(2)}(k+m/k) = \sum_{i=1}^m M\left\{x^T(k+i) / x(k), \eta(k), Y(k), \theta(k)\right\} R_2(k+i)M\{x(k+i) / x(k), \eta(k), Y(k), \theta(k)\}. \quad (9)$$

Очевидно, что

$$J(k+m/k) = J^{(1)}(k+m/k) - J^{(2)}(k+m/k). \quad (10)$$

Лемма 1. Выражение (8) для $J^{(1)}(k+m/k)$ может быть представлено в виде

$$J^{(1)}(k+m/k) = C^{(1)}[x(k), k] + [2x^T(k)G^{(1)}(k) - F(k)]U(k) + U^T(k)H^{(1)}(k)U(k), \quad (11)$$

где

$$C^{(1)}[x(k), k] = x^T(k)A^T Q_1(m-1)Ax(k) - Q_2(m-1)Ax(k), \quad (12)$$

блоки матриц $H^{(1)}(k)$, $G^{(1)}(k)$, $F(k)$ удовлетворяют уравнениям:

$$\begin{aligned} H_{tt}^{(1)}(k) &= R(k+t-1) + \\ &+ \sum_{i_1=1}^v \dots \sum_{i_t=1}^v B^T [\beta^{(i_1)} \alpha^{(i_1-1)} \dots \alpha^{(i_t)} Y(k)] Q_1(m-t) \Theta^{(i_1, \dots, i_t)}(k) B [\beta^{(i_1)} \alpha^{(i_1-1)} \dots \alpha^{(i_t)} Y(k)] + \\ &+ \sum_{i_1=1}^v M \left\{ B^T [\lambda^{(i_1)} \omega(k+t)] Q_1(m-t) \Theta^{(i_1)}(k) B [\lambda^{(i_1)} \omega(k+t)] \right\} + \\ &+ \sum_{j=1}^{t-1} \sum_{i_j=1}^v \dots \sum_{i_1=1}^v M \left\{ B^T [\beta^{(i_1)} \alpha^{(i_1-1)} \dots \alpha^{(i_{j+1})} \sigma^{(i_j)} W(k+j)] Q_1(m-t) \Theta^{(i_1, \dots, i_j)}(k) B [\beta^{(i_1)} \alpha^{(i_1-1)} \dots \alpha^{(i_{j+1})} \sigma^{(i_j)} W(k+j)] \right\}, \end{aligned} \quad (13)$$

$$\begin{aligned} H_{ff}^{(1)}(k) &= \sum_{i_1=1}^v \dots \sum_{i_f=1}^v B^T [\beta^{(i_1)} \alpha^{(i_1-1)} \dots \alpha^{(i_f)} Y(k)] (A^T)^{f-t} Q_1(m-f) \Theta^{(i_1, \dots, i_f)}(k) B [\beta^{(i_f)} \alpha^{(i_f-1)} \dots \alpha^{(i_1)} Y(k)] + \\ &+ \sum_{j=1}^{t-1} \sum_{i_j=1}^v \dots \sum_{i_f=1}^v M \left\{ B^T [\beta^{(i_1)} \alpha^{(i_1-1)} \dots \alpha^{(i_{j+1})} \sigma^{(i_j)} W(k+j)] (A^T)^{f-t} \times \right. \\ &\times \left. Q_1(m-f) \Theta^{(i_1, \dots, i_f)}(k) B [\beta^{(i_f)} \alpha^{(i_f-1)} \dots \alpha^{(i_{j+1})} \sigma^{(i_j)} W(k+j)] \right\}, f > t, \\ &\sum_{j=1}^0 (\cdot) = 0, \end{aligned} \quad (14)$$

$$H_{ff}^{(1)}(k) = \left(H_{ft}^{(1)}(k) \right)^T, f < t, \quad (15)$$

$$G_t^{(1)}(k) = (A^t)^T Q_1(m-t) \sum_{i_1=1}^v \dots \sum_{i_t=1}^v B [\beta^{(i_1)} \alpha^{(i_1-1)} \dots \alpha^{(i_t)} Y(k)] \Theta^{(i_1, \dots, i_t)}(k), \quad (16)$$

$$F_t(k) = Q_2(m-t) \sum_{i_1=1}^v \dots \sum_{i_t=1}^v B [\beta^{(i_1)} \alpha^{(i_1-1)} \dots \alpha^{(i_t)} Y(k)] \Theta^{(i_1, \dots, i_t)}(k). \quad (17)$$

Последовательности матриц $Q_1(t)$, $Q_2(t)$ ($t = \overline{1, m}$), $\Theta^{(i)}(k)$, $\Theta^{(i_1, \dots, i_f)}(k)$ ($t, f = \overline{1, m}$, $f > t$) определяются уравнениями:

$$Q_1(t) = R_1(k+m-t) + A^T Q_1(t-1)A, t = \overline{1, m}, Q_1(0) = R_1(k+m), \quad (18)$$

$$Q_2(t) = R_3(k+m-t) + Q_2(t-1)A, t = \overline{1, m}, Q_2(0) = R_3(k+m), \quad (19)$$

$$\Theta^{(i_1, \dots, i_f)}(k) = P_{i_f, i_{f-1}} P_{i_{f-1}, i_{f-2}} \dots P_{i_{t+1}, i_t} \theta_{i_t}(k+t/k), t = \overline{1, m-1}, f > t, \quad (20)$$

$$\Theta^{(i)}(k) = \theta_{i_t}(k+t/k), t = \overline{1, m}, i_t = \overline{1, v}, \quad (21)$$

где $\theta_{i_t}(k+t/k)$ – компонента вектора $\theta(k+t/k) = P^t \theta(k)$.

Доказательство. Выражая последовательно все $x(k+i)$ через $x(k)$ из (1), $\eta(k+i)$ через $\eta(k)$ из (2), $Y(k+i)$ через $Y(k)$ из (3), $\theta(k+i)$ через $\theta(k)$ из (5) и подставляя результат в (8), получим

$$\begin{aligned} J_1(k+m/k) &= x^T(k)A^T Q(m-1)Ax(k) + \\ &+ 2x^T(k) \sum_{t=1}^m (A^t)^T Q_1(m-t) \sum_{i_1=1}^v \dots \sum_{i_t=1}^v B [\beta^{(i_1)} \alpha^{(i_1-1)} \dots \alpha^{(i_t)} Y(k)] \Theta^{(i_1, \dots, i_t)}(k) + \\ &+ \sum_{t=1}^m u^T(k+t-1/k) \sum_{i_1=1}^v \dots \sum_{i_t=1}^v B^T [\beta^{(i_1)} \alpha^{(i_1-1)} \dots \alpha^{(i_t)} Y(k)] Q_1(m-t) \Theta^{(i_1, \dots, i_t)}(k) B [\beta^{(i_1)} \alpha^{(i_1-1)} \dots \alpha^{(i_t)} Y(k)] u(k+t-1/k) + \end{aligned} \quad (22)$$

$$\begin{aligned}
& + \sum_{t=1}^m u^T(k+t-1/k) \sum_{i_1=1}^v M \left\{ B^T [\lambda^{(i_1)} \omega(k+t)] Q_1(m-t) \Theta^{(i_1)}(k) B [\lambda^{(i_1)} \omega(k+t)] \right\} u(k+t-1/k) + \\
& + \sum_{t=1}^m u^T(k+t-1/k) \sum_{j=1}^{t-1} \sum_{i_1=1}^v \dots \sum_{i_j=1}^v M \left\{ B^T [\beta^{(i_1)} \alpha^{(i_1)} \dots \alpha^{(i_{j+1})} \sigma^{(i_j)} W(k+j)] \times \right. \\
& \times Q_1(m-t) \Theta^{(i_1, \dots, i_t)}(k) B [\beta^{(i_1)} \alpha^{(i_1)} \dots \alpha^{(i_{j+1})} \sigma^{(i_j)} W(k+j)] \left. \right\} u(k+t-1/k) + \\
& + 2 \sum_{t=1}^{m-1} \sum_{f=t+1}^m u^T(k+t-1/k) \sum_{i_1=1}^v \dots \sum_{i_f=1}^v B^T [\beta^{(i_1)} \alpha^{(i_1)} \dots \alpha^{(i_f)} Y(k)] (A^T)^{f-t} \times \\
& \times Q_1(m-f) \Theta^{(i_1, \dots, i_f)}(k) B [\beta^{(i_f)} \alpha^{(i_{f-1})} \dots \alpha^{(i_1)} Y(k)] u(k+f-1/k) + \\
& + 2 \sum_{t=1}^{m-1} \sum_{f=t+1}^m u^T(k+t-1/k) \sum_{j=1}^{t-1} \sum_{i_1=1}^v \dots \sum_{i_j=1}^v M \left\{ B^T [\beta^{(i_1)} \alpha^{(i_1)} \dots \alpha^{(i_{j+1})} \sigma^{(i_j)} W(k+j)] (A^T)^{f-t} \times \right. \\
& \times Q_1(m-f) \Theta^{(i_1, \dots, i_f)}(k) B [\beta^{(i_f)} \alpha^{(i_{f-1})} \dots \alpha^{(i_{j+1})} \sigma^{(i_j)} W(k+j)] \left. \right\} u(k+f-1/k) - \\
& - Q_2(m-1) A x(k) - \sum_{t=1}^m Q_2(m-t) \sum_{i_1=1}^v \dots \sum_{i_t=1}^v B [\beta^{(i_1)} \alpha^{(i_1)} \dots \alpha^{(i_t)} Y(k)] \Theta^{(i_1, \dots, i_t)}(k) u(k+t-1/k) + \\
& + \sum_{t=1}^m u^T(k+t-1/k) R(k+t-1) u(k+t-1/k),
\end{aligned}$$

где $\sum_{j=1}^0 (\cdot) = 0$, последовательности матриц $Q_1(t)$, $Q_2(t)$ ($t = \overline{0, m-1}$) определяются уравнениями (18)–

(19), матрицы $\Theta^{(i)}(k)$, $\Theta^{(i_1, \dots, i_f)}(k)$ ($f, t = \overline{1, m}$, $f > t$) определяются уравнениями (20)–(21).

Выражение (22) можно записать в матричной форме (11), где матрицы $G^{(1)}(k)$, $H^{(1)}(k)$, $F(k)$ имеют вид (13)–(17), $C^{(1)}[x(k), k]$ имеет вид (12). Лемма доказана.

Лемма 2. Выражение (9) для $J^{(2)}(k+m/k)$ может быть представлено в виде:

$$J^{(2)}(k+m/k) = C^{(2)}[x(k), k] + 2x^T(k)G^{(2)}(k)U(k) + U^T(k)H^{(2)}(k)U(k), \quad (23)$$

где

$$C^{(2)}[x(k), k] = x^T(k)\Psi^T \Delta(k+1)\Psi x(k), \quad (24)$$

$$G^{(2)}(k) = \Psi^T \Delta(k+1)\Phi(k), \quad (25)$$

$$H^{(2)}(k) = \Phi^T(k)\Delta(k+1)\Phi(k), \quad (26)$$

$$\Delta(k+1) = \text{diag}\{R_2(k+1), \dots, R_2(k+m)\},$$

блоки матриц $\Phi(k)$, Ψ имеют вид:

$$\Phi_{ft}(k) = A^{f-t} \sum_{i_1=1}^v \dots \sum_{i_t=1}^v B [\beta^{(i_1)} \alpha^{(i_1)} \dots \alpha^{(i_t)} Y(k)] \Theta^{(i_1, \dots, i_t)}(k), \quad t, f = \overline{1, m}, f \geq t, \quad (27)$$

$$\Phi_{ft}(k) = 0, \quad f < t, \quad (28)$$

$$\Psi_t = A^t, \quad t = \overline{1, m}, \quad (29)$$

где $\Theta^{(i_1, \dots, i_t)}(k)$ определяется выражениями (20)–(21).

Доказательство. Используя уравнения (1)–(4), получим

$$\begin{aligned}
& M\{x(k+t) / x(k), \eta(k), Y(k), \theta(k)\} = A^t x(k) + \\
& + \sum_{j=1}^t A^{t-j} \sum_{i_1=1}^v \dots \sum_{i_j=1}^v B [\beta^{(i_1)} \alpha^{(i_1)} \dots \alpha^{(i_j)} Y(k)] \Theta^{(i_1, \dots, i_j)}(k) u(k+j-1/k), \quad t = \overline{1, m},
\end{aligned} \quad (30)$$

где матрицы $\Theta^{(i_1, \dots, i_j)}(k)$ ($j = \overline{1, m}$) определяются уравнениями (20)–(21).

Введем вектор

$$X(k+1) = \begin{bmatrix} M\{x(k+1)/x(k), \eta(k), Y(k), \theta(k)\} \\ \dots \\ M\{x(k+m)/x(k), \eta(k), Y(k), \theta(k)\} \end{bmatrix}.$$

С учетом (30) динамика вектора $X(k+1)$ может быть записана в матричном виде:

$$X(k+1) = \Psi x(k) + \Phi(k)U(k), \quad (31)$$

где матрицы $\Phi(k)$, Ψ имеют вид (27)–(29).

Выражение (9) для $J^{(2)}(k+m/k)$ может быть записано в виде:

$$J^{(2)}(k+m/k) = X^T(k+1)\Delta(k+1)X(k+1), \quad (32)$$

где $\Delta(k+1) = \text{diag}\{R_2(k+1), \dots, R_2(k+m)\}$.

Подставляя (31) в (32), получим

$$J^{(2)}(k+m/k) = x^T(k)\Psi^T\Delta(k+1)\Psi x(k) + \\ + 2x^T(k)\Psi^T\Delta(k+1)\Phi(k)U(k) + U^T(k)\Phi^T(k)\Delta(k+1)\Phi(k)U(k). \quad (33)$$

Выражение (33) можно записать в виде (23), где матрицы $C^{(2)}[x(k), k]$, $G^{(2)}(k)$, $H^{(2)}(k)$ имеют вид (24)–(26) соответственно. Лемма доказана.

На основе лемм 1 и 2 можно показать, что задача прогнозирующего управления системой (1)–(4) по критерию (7) при ограничениях (6) сводится к задаче квадратичного программирования.

Теорема. Вектор прогнозирующих управлений $U(k)$ системой (1)–(4), минимизирующий критерий (7) при ограничениях (6), на каждом шаге k определяется из решения задачи квадратичного программирования с критерием вида:

$$Z(k+m/k) = [2x^T(k)G(k) - F(k)]U(k) + U^T(k)H(k)U(k), \quad (34)$$

при ограничениях

$$U_{\min}(k) \leq \bar{S}(k)U(k) \leq U_{\max}(k), \quad (35)$$

где

$$\bar{S}(k) = \text{diag}\{S(k), \dots, S(k+m-1)\}, \quad U_{\min}(k) = [u_{\min}^T(k), \dots, u_{\min}^T(k)]^T, \quad U_{\max}(k) = [u_{\max}^T(k), \dots, u_{\max}^T(k)]^T, \\ G(k) = G^{(1)}(k) - G^{(2)}(k), \quad H(k) = H^{(1)}(k) - H^{(2)}(k).$$

Оптимальное управление со скользящим горизонтом m в каждый момент времени k равно

$$u(k) = \begin{bmatrix} I_{n_u} & 0_{n_u} & \dots & 0_{n_u} \end{bmatrix} U(k), \quad (36)$$

где I_{n_u} – единичная матрица размерности n_u , 0_{n_u} – квадратная нулевая матрица размерности n_u .

Доказательство. Из (10), (11) и (23) следует, что критерий (7) может быть представлен в виде:

$$J(k+m/k) = C^{(1)}[x(k), k] - C^{(2)}[x(k), k] + \\ + 2x^T(k)[G^{(1)}(k) - G^{(2)}(k)]U(k) - F(k)U(k) + U^T(k)[H^{(1)}(k) - H^{(2)}(k)]U(k). \quad (37)$$

Очевидно, что задача минимизации критерия (37) эквивалентна задаче минимизации критерия (34), где удалены слагаемые, не зависящие от управлений. Таким образом, получаем, что задача минимизации критерия (7) по последовательности прогнозирующих управлений $U(k)$ эквивалентна задаче квадратичного программирования с критерием (34). Теорема доказана.

Заключение

В данной работе предложен метод синтеза стратегий прогнозирующего управления по обобщенному критерию для дискретных стохастических систем, динамика которых зависит от случайного процесса, описываемого уравнением многомерной регрессии с экзогенными факторами и параметрами.

рами, зависящими от состояния цепи Маркова. Динамика экзогенных факторов описывается MS-VAR(p) моделью. Изменяя весовые матрицы в обобщенном критерии, можно получать различные критерии управления: квадратичный критерий, критерий «mean-variance». Синтезированы оптимальные стратегии управления с учетом явных ограничений на управляющие воздействия.

ЛИТЕРАТУРА

1. Costa O.L.V., Fragoso M.D., Marques R.P. Discrete-time Markov jump linear systems. New York : Springer, 2005. 286 p.
2. Dombrovskii V., Obedko T. Feedback predictive control strategies for investment in the financial market with serially correlated returns subject to constraints and trading costs // Optimal control applications and methods. 2017. V. 38, No. 6. P. 908–921.
3. Mayne D.Q. Model predictive control: Recent developments and future promise // Automatica. 2014. V. 50, No.12. P. 2967–2986.
4. Farina M., Giulioni L., Scattolini R. Stochastic model predictive control with chance constraints : a review // Journal of Process Control. 2016. V. 44, No. 8. P. 53–67.
5. 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.
6. Henandez-Medjias M.A., Sala A., Querol A., Arino C. Multiple-Horizon predictive control for Markov/switched linear systems // IFAC-PapersOnLine. 2015. V. 48 (23). P. 230–235.
7. Tonne J., Jilg M., Stursberg O. Constrained Model Predictive Control of High Dimensional Jump Markov Linear Systems // American Control Conference. Palmer House Hilton. July 1–3, Chicago, IL. 2015. P. 2993–2998.
8. Chitraganti S., Aberkane S., Aubrun C., Valencia-Palomo G., Dragan V. On control of discrete-time state-dependent jump linear systems with probabilistic constraints: a receding horizon approach // Systems & Control Letters. 2014. V. 74. P. 81–89.
9. Lu J., Xi Y., Li D. Stochastic model predictive control for probabilistically constrained Markovian jump linear systems with additive disturbance // Int. J. Robust Nonlinear Control. 2017. P. 1–15.
10. Sala A., Henandez-Medjias M.A., Arino C. Stable receding-horizon scenario predictive control for Markov-jump linear systems // Automatica. 2017. V. 86. P. 121–128.
11. Patrinos P., Sotiras P., Sarimveis H., Bemporad A. Stochastic model predictive control for constrained discrete-time Markovian switching systems // Automatica. 2014. V. 50, No. 10. P. 2504–2514.
12. Dombrovskii V.V., Obedko T.Yu., Samorodova M. Model predictive control of constrained Markovian jump nonlinear stochastic systems and portfolio optimization under market frictions // Automatica. 2018. V. 87, No. 1. P. 61–68.
13. Домбровский В.В., Пашинская Т.Ю. Прогнозирующее управление системами с марковскими скачками и авторегрессионным мультипликативным шумом с марковским переключением режимов // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2018. № 44. С. 4–9.
14. Krolzig H.-M. Markov Switching Vector Autoregressions. Modelling, Statistical Inference and Application to Business Cycle Analysis. Berlin : Springer, 1997. 357 p.
15. Elliott G.J., Aggoun L., Moore J.B. Hidden Markov models: Estimation and control. Berlin, Heidelberg, New York : Springer, 1995. 382 p.

Поступила в редакцию 22 ноября 2018 г.

Dombrovskii V.V., Pashinskaya T.Yu. (2019) OPTIMAL PREDICTIVE CONTROL STRATEGIES FOR SYSTEMS WITH RANDOM PARAMETERS DESCRIBED BY MULTIDIMENSIONAL MARKOV SWITCHING REGRESSION MODEL. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie vychislitel'naja tehnika i informatika* [Tomsk State University Journal of Control and Computer Science]. 48. pp. 4–12

DOI: 10.17223/19988605/48/1

Let the control object is described by the equations:

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

$$\eta(k+1) = \beta[\theta(k+1)]Y(k) + \lambda[\theta(k+1)]\omega(k+1), \quad (2)$$

$$Y(k+1) = \alpha[\theta(k+1)]Y(k) + \sigma[\theta(k+1)]W(k+1), \quad (3)$$

$$\alpha[\theta(k)] = \sum_{i=1}^{\nu} \theta_i(k) \alpha^{(i)}, \quad \lambda[\theta(k)] = \sum_{i=1}^{\nu} \theta_i(k) \lambda^{(i)}, \quad \beta[\theta(k)] = \sum_{i=1}^{\nu} \theta_i(k) \beta^{(i)}, \quad \sigma[\theta(k)] = \sum_{i=1}^{\nu} \theta_i(k) \sigma^{(i)}, \quad (4)$$

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, $\eta(k) \in \mathbb{R}^{n_\eta}$ is a sequence of stochastic vectors, $Y(k) = [y^T(k), y^T(k-1), \dots, y^T(k-p+1)]^T$, $y(k) \in \mathbb{R}^{n_y}$, $W(k) = [w^T(k), 0, 0, \dots, 0]^T_{n_y, p \times 1}$, $w(k) \in \mathbb{R}^{n_y}$, $\omega(k) \in \mathbb{R}^{n_\omega}$ are white noise vectors with zero mean and unique covariance matrices; $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$, $\beta^{(i)} \in \mathbb{R}^{n_\eta \times n_y}$, $\lambda^{(i)} \in \mathbb{R}^{n_\eta \times n_\omega}$,

$$\alpha^{(i)} = \begin{bmatrix} \alpha_1^{(i)} & \alpha_2^{(i)} & \dots & \alpha_{p-1}^{(i)} & \alpha_p^{(i)} \\ I_{n_y} & 0 & \dots & 0 & 0 \\ 0 & I_{n_y} & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & I_{n_y} & 0 \end{bmatrix}_{n_y p \times n_y p}, \quad \alpha_j^{(i)} \in \mathbb{R}^{n_y \times n_y},$$

$\sigma^{(i)} = \text{diag}\{\varphi^{(i)}, 0, \dots, 0\}$, $\varphi^{(i)} \in \mathbb{R}^{n_y \times n_y}$, $\theta_i(k+1)$ ($i = \overline{1, v}$) are the components of the vector $\theta(k+1)$, $\theta(k) = [\delta(\tau(k), 1), \dots, \delta(\tau(k), v)]^T$, $\delta(\tau(k), j)$ is the Kronecker function; $\{\tau(k); k = 0, 1, 2, \dots\}$ is a finite-state discrete-time homogeneous Markov chain taking values in $\{1, 2, \dots, v\}$ with transition probability matrix $P = [P_{ij}]$. Sequences $w(k)$, $\omega(k)$, and $\theta(k)$ are independent. It is assumed that the state of Markov chain is observable at time k . All of the elements $B[\eta(k)]$ ($i = \overline{1, v}$) are assumed to be linear functions of $\eta(k)$. Equation (2) is a multivariate regression model with exogenous factors $Y(k)$ and regime-switching parameters. The dynamics of factors $Y(k)$ follows vector autoregression model with regime switching (MS VAR (p)) of order p represented in the form of MS VAR (1) model (Equation (3)).

We impose the following inequality constraints on the control inputs (element-wise inequality):

$$u_{\min}(k) \leq S(k)u(k) \leq u_{\max}(k); \quad S(k) \in \mathbb{R}^{q \times n_u}; \quad u_{\min}(k), \quad u_{\max}(k) \in \mathbb{R}^q. \quad (5)$$

For control of system (1)–(4), we synthesize the strategies with a predictive control model. At each step k , we minimize the following criterion with a receding horizon

$$\begin{aligned} J(k+m/k) = & \sum_{i=1}^m E\{x^T(k+i)R_1(k+i)x(k+i) / x(k), \eta(k), Y(k), \theta(k)\} - \\ & - \sum_{i=1}^m E\{x^T(k+i) / x(k), \eta(k), Y(k), \theta(k)\}R_2(k+i)E\{x(k+i) / x(k), \eta(k), Y(k), \theta(k)\} - \\ & - \sum_{i=1}^m R_3(k+i)E\{x(k+i) / x(k), \eta(k), Y(k), \theta(k)\} + \sum_{i=0}^{m-1} E\{u^T(k+i/k)R(k+i)u(k+i/k) / x(k), \eta(k), Y(k), \theta(k)\}, \end{aligned} \quad (6)$$

on trajectories of system (1)–(4) over the sequence of predictive controls $u(k/k), \dots, u(k+m-1/k)$ dependent on the system state at the moment k , under constraints (5); where $R_1(k+i) \geq 0$, $R_2(k+i) \geq 0$, $R(k+i) > 0$ are given symmetric weight matrices of corresponding dimensions; $R_3(k+i)$ is a given vector of corresponding dimension; m is the prediction horizon. Different cost functions can be obtained from criterion (6) after setting the coefficients $R_1(k+i)$, $R_2(k+i)$, and $R_3(k+i)$ to some appropriate values.

Problem 1. Taking $R_2(k+i)$ ($i = \overline{1, m}$), we have the MPC problem with quadratic criterion.

Problem 2. Let system (1)–(4) have a scalar output $z(k) = c(k)x(k)$, where $c(k)$ is a vector of appropriate dimension. Taking $R_1(k+i) = R_2(k+i) = \rho_1(k+i)c^T(k+i)c(k+i)$, $R_3(k+i) = \rho_2(k+i)c(k+i)$ ($i = \overline{1, m}$), where $\rho_1(k+i) \geq 0$, $\rho_2(k+i) \geq 0$ are scalar values, we have a mean-variance optimization problem.

Keywords: stochastic systems; Markov jumps; multidimensional regression; model predictive control; constrains.

DOMBROVSKII Vladimir Valentinovich (Doctor of Technical Sciences, Professor, National Research Tomsk State University, Tomsk, Russian Federation).

E-mail: dombrovs@ef.tsu.ru

PASHINSKAYA Tatiana Yurievna (Candidate of Physical and Mathematical Sciences, Associate professor, National Research Tomsk State University, Tomsk, Russian Federation).

E-mail: tatyana.obedko@mail.ru

REFERENCES

1. Costa, O.L.V., Fragoso, M.D. & Marques, R.P. (2005) *Discrete-time Markov jump linear systems*. Springer: New York.
2. Dombrovskii, V.V. & Obedko, T. (2017) Feedback predictive control strategies for investment in the financial market with serially correlated returns subject to constraints and trading costs. *Optimal Control Applications and Methods*. 38(6). pp. 908–921. DOI: 10.1002/oca.2296
3. Mayne, D.Q. (2014) Model predictive control: Recent developments and future promise. *Automatica*. 50(12). pp. 2967–2986. DOI: 10.1016/j.automatica.2014.10.128
4. Farina, M., Giullioni, L. & Scatoloni, R. (2016) Stochastic model predictive control with chance constraints. A review. *Journal of Process Control*. 44(8). pp. 53–67.
5. 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: 10.1134/S0005117911050079
6. Henandez-Medjias, M.A., Sala, A., Querol, A. & Arino, C. (2015) Multiple-Horizon predictive control for Markov/switched linear systems. *IFAC-PapersOnLine*. vol. 48. no 23. pp. 230–235. DOI: 10.1016/j.ifacol.2015.11.288

7. Tonne, J., Jilg, M. & Stursberg, O. (2015) *Constrained Model Predictive Control of High Dimensional Jump Markov Linear Systems*. Chicago, IL, USA, Palmer House Hilton. pp. 2993–2998. DOI: 10.1109/ACC.2015.7171190
8. Chitraganti, S., Aberkane, S., Aubrun, C., Valencia-Palomo, G. & Dragan, V. (2014) On control of discrete-time state-dependent jump linear systems with probabilistic constraints: A receding horizon approach. *Systems & Control Letters*. 74. pp. 81–89. DOI: 10.1016/j.sysconle.2014.10.008
9. Lu, J., Xi, Y. & Li, D. (2017) Stochastic model predictive control for probabilistically constrained Markovian jump linear systems with additive disturbance. *International Journal of Robust and Nonlinear Control*. pp. 1–15. DOI: 10.1002/rnc.3971
10. Sala, A., Henandez-Medjias, M.A. & Arino, C. (2017) Stable receding-horizon scenario predictive control for Markov-jump linear systems. *Automatica*. 86. pp. 121–128. DOI: 10.1016/j.automatica.2017.07.032
11. Patrinos, P., Sotiras, P., Sarimveis, H. & Bemporad, A. (2014) Stochastic model predictive control for constrained discrete-time Markovian switching systems. *Automatica*. 50(10). pp. 2504–2514. DOI: 10.1016/j.automatica.2014.08.031
12. Dombrovskii, V.V., Obyedko, T.Yu. & Samorodova, M. (2018) Model predictive control of constrained Markovian jump nonlinear stochastic systems and portfolio optimization under market frictions. *Automatica*. 87(1). pp. 61–68. DOI: 10.1016/j.automatica.2017.09.018
13. Dombrovskii, V.V. & Pashinskaya, T.Y. (2018) Predictive Control for Markov Jump Systems with Markov switching autoregressive multiplicative noise. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science*. 44. pp. 4–9. (In Russian). DOI: 10.17223/19988605/44/1
14. Krolzig, H.-M. (1997) *Markov Switching Vector Autoregressions. Modelling, Statistical Inference and Application to Business Cycle Analysis*. Berlin: Springer.
15. Elliott, R.J., Aggoun, L. & Moore, J.B. (1995) *Hidden Markov Models: Estimation and Control*. Berlin, Heidelberg, New York: Springer.

ОБРАБОТКА ИНФОРМАЦИИ

УДК 519.872

DOI: 10.17223/19988605/48/2

А.А. Назаров, С.В. Пауль, О.Д. Лизюра

АСИМПТОТИЧЕСКИЙ АНАЛИЗ RQ-СИСТЕМЫ С N ТИПАМИ ВЫЗЫВАЕМЫХ ЗАЯВОК В ПРЕДЕЛЬНОМ УСЛОВИИ БОЛЬШОЙ ЗАДЕРЖКИ ЗАЯВОК НА ОРБИТЕ

Работа выполнена при финансовой поддержке гранта РФФИ № 18-01-00277.

Рассматривается RQ-система с несколькими типами вызываемых заявок. Основным методом исследования является метод асимптотического анализа, который позволяет в предложенной RQ-системе найти вид предельного распределения числа заявок, поступивших в систему в условии большой задержки заявок на орбите. На основе найденного распределения построено дискретное распределение (гауссовская аппроксимация). Определены условия применимости полученной аппроксимации в зависимости от значений параметров, определяющих систему, на основе численных экспериментов.

Ключевые слова: RQ-система; вызываемые заявки; метод асимптотического анализа; предельное условие большой задержки; гауссовская аппроксимация.

RQ-системы характеризуются тем, что заявка, поступившая в систему, в случае занятости сервера остается в ней и пытается вновь занять обслуживающий прибор после некоторой случайной задержки на орбите. RQ-системы являются математическими моделями телекоммуникационных сетей связи, компьютерных сетей, систем в экономике и систем call-центров [1, 2]. В таких системах время простоя сервера должно быть уменьшено для повышения эффективности системы.

Мы рассматриваем системы, в которых оператор не только принимает вызовы извне, но и выполняет исходящие вызовы в режиме простоя. Например, в call-центрах операторы могут получать поступающие вызовы, но как только они имеют свободное время и находятся в режиме ожидания, они могут выполнять исходящие вызовы [3, 4]. Такие системы будем называть RQ-системами с вызываемыми заявками, или системами с двумя классами заявок.

В работах [5–7] рассматриваются марковские RQ-системы с вызываемыми заявками. Модель RQ-системы с двумя классами заявок и несколькими типами вызываемых заявок рассмотрена Сакураи и Фунг-Дуком [8]. Для этой модели получен численный алгоритм расчета стационарного распределения состояний системы.

В предложенной работе основным методом исследования является метод асимптотического анализа [9, 10], который позволяет в RQ-системе $M/M/1/N$ с N типами вызываемых заявок найти вид предельного распределения числа поступивших заявок в системе в условии большой задержки заявок на орбите. На основе найденного распределения построено дискретное распределение (гауссовская аппроксимация), которое аппроксимирует дискретное распределение числа поступивших заявок в системе.

1. Описание математической модели и постановка задачи

Рассмотрим однолинейную RQ-систему с несколькими типами вызываемых заявок, на вход которой поступает простейший поток заявок с параметром λ . Время обслуживания каждой поступившей заявки распределено по экспоненциальному закону с параметром μ_1 . Если поступившая заявка

застает прибор свободным, она занимает его для обслуживания. Если прибор занят, то заявка переходит на орбиту, где осуществляет случайную задержку, продолжительность которой имеет экспоненциальное распределение с параметром σ . С орбиты после случайной задержки заявка вновь обращается к прибору с повторной попыткой его захвата.

Когда прибор свободен, он вызывает заявки извне. Рассматривается система с N типами вызываемых заявок. Прибор вызывает заявки типа n с интенсивностью α_n . Время обслуживания вызванной заявки типа n распределено по экспоненциальному закону с параметром μ_n .

Обозначим $i(t)$ – число поступивших заявок в системе в момент времени t , без учета вызванной заявки, если она обслуживается на приборе. Процесс $k(t)$ определяет состояние прибора в момент времени t следующим образом: 0, если прибор свободен; 1, если прибор занят обслуживанием поступившей заявки; n , если прибор занят обслуживанием вызванной заявки типа n , где $n = \overline{2, N+1}$. Двумерный процесс $\{i(t), k(t)\}$ является цепью Маркова с непрерывным временем.

Введем обозначение $P\{i(t) = i, k(t) = k\} = P_k(i)$ – стационарная вероятность того, что в момент времени t прибор находится в состоянии k и в системе находится i поступивших заявок. Для распределения вероятностей $P_k(i)$ рассматриваемой RQ-системы составим систему уравнений Колмогорова:

$$\begin{aligned} -\left(\lambda + i\sigma + \sum_{n=2}^{N+1} \alpha_n\right)P_0(i) + \mu_1 P_1(i+1) + \sum_{n=2}^{N+1} \mu_n P_n(i) &= 0, \\ -(\lambda + \mu_1)P_1(i) + \lambda P_1(i-1) + \lambda P_0(i-1) + i\sigma P_0(i) &= 0, \\ -(\lambda + \mu_n)P_n(i) + \lambda P_n(i-1) + \alpha_n P_0(i) &= 0, \quad n = \overline{2, N+1}. \end{aligned} \quad (1)$$

Введем частичные характеристические функции $H_k(u) = \sum_{i=0}^{\infty} e^{ju} P_k(i)$, где $j = \sqrt{-1}$, $k = \overline{0, N+1}$.

Тогда систему (1) перепишем в виде:

$$\begin{aligned} -\left(\lambda + \sum_{n=2}^{N+1} \alpha_n\right)H_0(u) + j\sigma H_0'(u) + \mu_1 e^{-ju} H_1(u) + \sum_{n=2}^{N+1} \mu_n H_n(u) &= 0, \\ -(\lambda + \mu_1)H_1(u) + \lambda e^{ju} H_1(u) + \lambda e^{ju} H_0(u) - j\sigma H_0'(u) &= 0, \\ -(\lambda + \mu_n)H_n(u) + \lambda e^{ju} H_n(u) + \alpha_n H_0(u) &= 0, \quad n = \overline{2, N+1}. \end{aligned} \quad (2)$$

Суммируя уравнения системы (2), получим уравнение

$$\lambda H_0(u) + (\lambda - \mu_1 e^{-ju}) H_1(u) + \lambda \sum_{n=2}^{N+1} H_n(u) = 0. \quad (3)$$

Характеристическая функция $H(u)$ числа заявок для системы (2), (3) выражается через частичные характеристические функции $H_k(u)$ следующим равенством: $H(u) = \sum_{k=0}^{N+1} H_k(u)$. Аналитическое выражение для $H(u)$ представлено в следующей теореме.

Теорема 1. Характеристическая функция $H(u)$ числа поступивших заявок в RQ-системе M/M/1/N с вызываемыми заявками имеет следующий вид:

$$H(u) = \frac{1}{1 + v_1} \left(1 + \sum_{n=2}^{N+1} \frac{\alpha_n}{\mu_n + \lambda(1 - e^{ju})} \right) \left[\frac{1 - \rho}{1 - \rho e^{ju}} \right]^{\lambda(1+v_2)+1} \prod_{n=2}^{N+1} \left[\frac{1 - p_n}{1 - p_n e^{ju}} \right]^{\frac{\alpha_n(\theta_n - \lambda)}{\sigma \theta_n}}, \quad (4)$$

где $\rho = \frac{\lambda}{\mu_1}$, $v_1 = \sum_{n=2}^{N+1} \frac{\alpha_n}{\mu_n}$, $v_2 = \sum_{n=2}^{N+1} \frac{\alpha_n}{\theta_n}$, $p_n = \frac{\lambda}{\lambda + \mu_n}$, $\theta_n = \lambda + \mu_n - \mu_1$, $n = \overline{2, N+1}$.

Доказательство. Для доказательства теоремы 1 необходимо второе и третье уравнения системы (2) разрешить относительно функций $H_1(u)$ и $H_n(u)$, $n = \overline{2, N+1}$ соответственно. Подставляя полученные выражения в первое уравнение системы (2), мы получим обыкновенное дифференциальное уравнение с разделяющимися переменными относительно функции $H_0(u)$. Решая дифференциальное уравнение, получаем функцию $H_0(u)$ в явном виде с точностью до мультипликативной константы.

Подставляем $H_0(u)$ в выражения для функций $H_1(u)$ и $H_n(u)$, $n = \overline{2, N+1}$ и суммируем все полученные функции согласно равенству $H(u) = \sum_{k=0}^{N+1} H_k(u)$. Константу интегрирования определяем из условия нормировки $H(0) = 1$. Теорема доказана.

Применяя обратное преобразование Фурье к найденной характеристической функции (4), можно записать распределение вероятностей $P(i)$ в виде $P(i) = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{-j\omega i} H(u) du$, $i = \overline{0, \infty}$. Однако нахождение аналитического выражения этих интегралов вряд ли возможно, следовательно, целесообразно использовать методы численного интегрирования. Численные расчеты, в свою очередь, требуют больших затрат вычислительных ресурсов.

В настоящей работе на основе применения метода асимптотического анализа, ставится задача построения аналитической аппроксимации распределения $P(i)$ и с помощью численных экспериментов проведение анализа точности.

2. Асимптотика первого порядка

Решение системы (2), (3) найдем с помощью метода асимптотического анализа при условии большой задержки заявок на орбите ($\sigma \rightarrow 0$).

Результат сформулируем в виде следующей теоремы.

Теорема 2. Пусть $i(t)$ – число поступивших заявок в RQ-системе с N типами вызываемых заявок, тогда для последовательности характеристических функций выполняется равенство

$$\lim_{\sigma \rightarrow 0} M e^{j\omega i(t)\sigma} = e^{j\omega \kappa_1}, \quad (5)$$

где

$$\kappa_1 = \frac{\lambda \mu_1 v + \lambda^2}{\mu_1 - \lambda}, \quad v = \sum_{n=2}^{N+1} \frac{\alpha_n}{\mu_n}. \quad (6)$$

Доказательство. Обозначим $\sigma = \varepsilon$ и сделаем в системе (4) следующие замены:

$$u = \varepsilon w, \quad H_k(u) = F_k(w, \varepsilon), \quad k = \overline{0, N+1},$$

получим следующую систему уравнений:

$$\begin{aligned} -\left(\lambda + \sum_{n=2}^{N+1} \alpha_n\right) F_0(w, \varepsilon) + j \frac{\partial F_0(w, \varepsilon)}{\partial w} + \mu_1 e^{-j\omega \varepsilon} F_1(w, \varepsilon) + \sum_{n=2}^{N+1} \mu_n F_n(w, \varepsilon) &= 0, \\ -(\lambda + \mu_1) F_1(w, \varepsilon) + \lambda e^{j\omega \varepsilon} F_1(w, \varepsilon) + \lambda e^{j\omega \varepsilon} F_0(w, \varepsilon) - j \frac{\partial F_0(w, \varepsilon)}{\partial w} &= 0, \\ -(\lambda + \mu_n) F_n(w, \varepsilon) + \lambda F_n(w, \varepsilon) + \alpha_n F_0(w, \varepsilon) &= 0, \quad n = \overline{2, N+1}, \\ \lambda F_0(w, \varepsilon) + (\lambda - \mu_1 e^{-j\omega \varepsilon}) F_1(w, \varepsilon) + \lambda \sum_{n=2}^{N+1} F_n(w, \varepsilon) &= 0. \end{aligned} \quad (7)$$

В системе (7) сделаем предельный переход при $\varepsilon \rightarrow 0$. Обозначив $F_k(w) = \lim_{\varepsilon \rightarrow 0} F_k(w, \varepsilon)$, $k = \overline{0, N+1}$, получим

$$\begin{aligned} -\left(\lambda + \sum_{n=2}^{N+1} \alpha_n\right) F_0(w) + j F_0'(w) + \sum_{k=1}^{N+1} \mu_k F_k(w) &= 0, \\ -\mu_1 F_1(w) - j F_0'(w) + \lambda F_0(w) &= 0, \\ -\mu_n F_n(w) + \alpha_n F_0(w) &= 0, \quad n = \overline{2, N+1}, \\ \lambda F_0(w) - (\mu_1 - \lambda) F_1(w) + \lambda \sum_{n=2}^{N+1} F_n(w) &= 0. \end{aligned} \quad (8)$$

Будем искать решение системы уравнений (8) в следующем виде:

$$F_k(w) = r_k \Phi(w), \quad k = \overline{0, N+1}, \quad (9)$$

где r_k – это вероятность того, что прибор находится в состоянии k .

$$\begin{aligned} -\left(\lambda + \sum_{n=2}^{N+1} \alpha_n\right)r_0 + j \frac{\Phi'(w)}{\Phi(w)} r_0 + \sum_{k=1}^{N+1} \mu_k r_k &= 0, \\ -\mu_1 r_1 - j \frac{\Phi'(w)}{\Phi(w)} r_0 + \lambda r_0 &= 0, \\ -\mu_n r_n + \alpha_n r_0 &= 0, \quad n = \overline{2, N+1}, \\ \lambda r_0 - (\mu_1 - \lambda) r_1 + \lambda \sum_{n=2}^{N+1} r_n &= 0. \end{aligned} \quad (10)$$

Так как отношение $j \frac{\Phi'(w)}{\Phi(w)}$ не зависит от w , функция $\Phi(w)$ имеет следующий вид: $\Phi(w) = \exp\{jw\kappa_1\}$, что соотносится с (5), где параметр κ_1 будет найден ниже. Систему (10) перепишем в виде:

$$\begin{aligned} -\left(\lambda + \sum_{n=2}^{N+1} \alpha_n\right)r_0 - \kappa_1 r_0 + \sum_{k=1}^{N+1} \mu_k r_k &= 0, \\ -\mu_1 r_1 + \kappa_1 r_0 + \lambda r_0 &= 0, \\ -\mu_n r_n + \alpha_n r_0 &= 0, \quad n = \overline{2, N+1}, \\ \lambda r_0 - (\mu_1 - \lambda) r_1 + \lambda \sum_{n=2}^{N+1} r_n &= 0. \end{aligned} \quad (11)$$

Запишем условие нормировки для распределения вероятностей состояний прибора: $\sum_{k=0}^{N+1} r_k = 1$.

Выписывая 3 и 4 уравнения системы (11) совместно с условием нормировки, получим систему

$$\begin{aligned} -\mu_n r_n + \alpha_n r_0 &= 0, \quad n = \overline{2, N+1}, \\ \lambda r_0 - (\mu_1 - \lambda) r_1 + \lambda \sum_{n=2}^{N+1} r_n &= 0, \\ \sum_{k=0}^{N+1} r_k &= 1, \end{aligned} \quad (12)$$

решение которой имеет вид: $r_0 = \frac{\mu_1 - \lambda}{\mu_1(1 + \nu)}$, $r_1 = \frac{\lambda}{\mu_1}$, $r_n = \frac{\alpha_n(\mu_1 - \lambda)}{\mu_1 \mu_n(1 + \nu)}$, $n = \overline{2, N+1}$, где $\nu = \sum_{n=2}^{N+1} \frac{\alpha_n}{\mu_n}$.

Подставляя полученное решение в систему (11) получаем значение параметра: $\kappa_1 = \frac{\lambda \nu \mu_1 + \lambda^2}{\mu_1 - \lambda}$. Теорема доказана.

Асимптотика первого порядка определяет среднее значение числа поступивших заявок в системе. Для более детального исследования процесса $i(t)$ следует рассмотреть асимптотику второго порядка.

3. Асимптотика второго порядка

Основной результат анализа асимптотики второго порядка представим в виде теоремы.

Теорема 3. Пусть $i(t)$ – число поступивших заявок в RQ-системе с N типами вызываемых заявок, тогда имеет место предельное равенство:

$$\lim_{\sigma \rightarrow 0} M \exp\left\{jw\sqrt{\sigma}\left(i(t) - \frac{\kappa_1}{\sigma}\right)\right\} = \exp\left\{\frac{(jw)^2}{2} \kappa_2\right\}, \quad (13)$$

где

$$\kappa_2 = \frac{\lambda\mu_1(\mu_1 - \lambda)(\nu_1 - \lambda\nu_2) + \lambda^2(\mu_1 + \lambda\nu_1)}{(\mu_1 - \lambda)^2}, \quad \nu_1 = \sum_{n=2}^{N+1} \frac{\alpha_n}{\mu_n}, \quad \nu_2 = \sum_{n=2}^{N+1} \frac{\alpha_n}{\mu_n^2}. \quad (14)$$

Доказательство. В системе (2), (3) сделаем следующие замены: $H_k(u) = \exp\left\{ju \frac{\kappa_1}{\sigma}\right\} H_k^{(2)}(u)$,

перепишем систему (14) в виде:

$$\begin{aligned} & -\left(\lambda + \sum_{n=2}^{N+1} \alpha_n + \kappa_1\right) H_0^{(2)}(u) + j\sigma \frac{dH_0^{(2)}(u)}{du} + \mu_1 e^{-ju} H_1^{(2)}(u) + \sum_{n=2}^{N+1} H_n^{(2)}(u) = 0, \\ & -(\lambda + \mu_1) H_1^{(2)}(u) + \lambda e^{ju} H_1^{(2)}(u) + (\lambda e^{ju} + \kappa_1) H_0^{(2)}(u) - j\sigma \frac{dH_0^{(2)}(u)}{du} = 0, \\ & -(\lambda + \mu_n) H_n^{(2)}(u) + \lambda e^{ju} H_n^{(2)}(u) + \alpha_n H_0^{(2)}(u) = 0, \quad n = \overline{2, N+1}, \\ & \lambda H_0^{(2)}(u) + (\lambda - \mu_1 e^{-ju}) H_1^{(2)}(u) + \lambda \sum_{n=2}^{N+1} H_n^{(2)}(u) = 0. \end{aligned} \quad (15)$$

В предельном условии большой задержки на орбите обозначим $\sigma = \varepsilon^2$, введем следующие замены:

$$u = w\varepsilon, \quad H_k^{(2)}(u) = F_k^{(2)}(w, \varepsilon), \quad k = \overline{0, N+1},$$

в результате получим систему:

$$\begin{aligned} & -\left(\lambda + \sum_{n=2}^{N+1} \alpha_n + \kappa_1\right) F_0^{(2)}(w, \varepsilon) + j\varepsilon \frac{\partial F_0^{(2)}(w, \varepsilon)}{\partial w} + \mu_1 e^{-jw\varepsilon} F_1^{(2)}(w, \varepsilon) + \sum_{n=2}^{N+1} F_n^{(2)}(w, \varepsilon) = 0, \\ & -(\lambda + \mu_1) F_1^{(2)}(w, \varepsilon) + \lambda e^{jw\varepsilon} F_1^{(2)}(w, \varepsilon) + (\lambda e^{jw\varepsilon} + \kappa_1) F_0^{(2)}(w, \varepsilon) - j\varepsilon \frac{\partial F_0^{(2)}(w, \varepsilon)}{\partial w} = 0, \\ & -(\lambda + \mu_n) F_n^{(2)}(w, \varepsilon) + \lambda e^{jw\varepsilon} F_n^{(2)}(w, \varepsilon) + \alpha_n F_0^{(2)}(w, \varepsilon) = 0, \quad n = \overline{2, N+1}, \\ & \lambda F_0^{(2)}(w, \varepsilon) + (\lambda - \mu_1 e^{-jw\varepsilon}) F_1^{(2)}(w, \varepsilon) + \lambda \sum_{n=2}^{N+1} F_n^{(2)}(w, \varepsilon) = 0. \end{aligned} \quad (16)$$

Найдем решение системы (16) в следующем виде:

$$F_k^{(2)}(w, \varepsilon) = \Phi_2(w) \{r_k + jw\varepsilon f_k\} + o(\varepsilon^2), \quad k = \overline{0, N+1}. \quad (17)$$

Подставляя разложение (17) в систему (16), учитывая (11) и выполняя несложные преобразования, получим

$$\begin{aligned} & -\left(\kappa_1 + \lambda + \sum_{n=2}^{N+1} \alpha_n\right) f_0 + \sum_{k=1}^{N+1} \mu_k f_k = \mu_1 r_1 - \frac{\Phi_2'(w)}{w\Phi_2(w)} r_0, \\ & (\kappa_1 + \lambda) f_0 - \mu_1 f_1 = -\lambda r_0 - \lambda r_1 + \frac{\Phi_2'(w)}{w\Phi_2(w)} r_0, \\ & -\mu_n f_n + \alpha_n f_0 = -\lambda r_n, \quad n = \overline{2, N+1}, \\ & \lambda f_0 - (\mu_1 - \lambda) f_1 + \lambda \sum_{n=2}^{N+1} f_n = -\mu_1 r_1. \end{aligned}$$

Из полученной системы можем сделать вывод, что выражение $\frac{\Phi_2'(w)}{w\Phi_2(w)}$ не зависит от w , следовательно,

функцию $\Phi_2(w)$ можно представить в виде: $\Phi_2(w) = \exp\left\{\frac{(jw)^2}{2} \kappa_2\right\}$, что соотносится с (13). Пере-

пишем последнюю систему в виде:

$$-\left(\kappa_1 + \lambda + \sum_{n=2}^{N+1} \alpha_n\right) f_0 + \sum_{k=1}^{N+1} \mu_k f_k = \mu_1 r_1 + \kappa_2 r_0,$$

$$\begin{aligned}
 (\kappa_1 + \lambda)f_0 - \mu_1 f_1 &= -\lambda r_0 - \lambda r_1 - \kappa_2 r_0, \\
 -\mu_n f_n + \alpha_n f_0 &= -\lambda r_n, \quad n = \overline{2, N+1}, \\
 \lambda f_0 - (\mu_1 - \lambda)f_1 + \lambda \sum_{n=2}^{N+1} f_n &= -\mu_1 r_1.
 \end{aligned} \tag{18}$$

Подставляя значения вероятностей r_k в систему (18), получим следующие выражения для функций f_n :

$$f_1 = \frac{\lambda(1 + v_1)}{\mu_1 - \lambda} f_0 + \frac{\lambda^2 v_2}{\mu_1(1 + v_1)} + \frac{\lambda}{\mu_1 - \lambda}, \quad f_n = \frac{\alpha_n}{\mu_n} f_0 + \frac{\lambda(\mu_1 - \lambda)\alpha_n}{\mu_1 \mu_n^2 (1 + v_1)}, \quad n = \overline{2, N+1},$$

где $v_1 = \sum_{n=2}^{N+1} \frac{\alpha_n}{\mu_n}$, $v_2 = \sum_{n=2}^{N+1} \frac{\alpha_n}{\mu_n^2}$. Подставляя эти выражения во второе уравнение системы (18), мы получим искомый параметр κ_2 . Нужно отметить, что неизвестная величина f_0 уже не входит в выражение для κ_2 , поэтому он однозначно определяется равенством

$$\kappa_2 = \frac{\lambda \mu_1 (\mu_1 - \lambda) (v_1 - \lambda v_2) + \lambda^2 (\mu_1 + \lambda v_1)}{(\mu_1 - \lambda)^2}.$$

Теорема доказана.

Теорема 3 показывает, что асимптотическое распределение вероятностей числа поступивших заявок в RQ-системе с несколькими типами вызываемых заявок является гауссовским с параметрами κ_1/σ и κ_2/σ , что позволяет для допредельного распределения $P(i)$ построить аппроксимацию, в частности аппроксимацию $P^{(2)}(i)$ вида

$$P^{(2)}(i) = (L(i + 0,5) - L(i - 0,5))(1 - L(-0,5))^{-1}, \tag{19}$$

где $L(x)$ – функция нормального распределения с параметрами κ_1/σ и κ_2/σ .

4. Точность аппроксимации

Точность аппроксимации $P^{(2)}(i)$ определим с помощью расстояния Колмогорова

$$\Delta_2 = \max_{0 \leq i \leq N} \left| \sum_{v=0}^i (P^{(2)}(v) - P(v)) \right|,$$

которое показывает разницу между распределением $P(i)$ и $P^{(2)}(i)$, где $P(i)$

получено с использованием обратного преобразования Фурье допредельной характеристической функции числа поступивших заявок в системе, найденного численно, а аппроксимация $P^{(2)}(i)$ построена на основе гауссовской аппроксимации.

Расстояние Колмогорова

	$\sigma = 0,2$	$\sigma = 0,1$	$\sigma = 0,05$	$\sigma = 0,035$	$\sigma = 0,02$	$\sigma = 0,01$
Δ_2	0,063	0,036	0,027	0,025	0,022	0,02

Положим $N = 3$, $\lambda = 0,2$, $\mu_1 = 1$, $\mu_2 = 2$, $\mu_3 = 3$, $\mu_4 = 4$, $\alpha_2 = 1$, $\alpha_3 = 2$, $\alpha_4 = 3$. В табл. 1 приведены значения расстояния Колмогорова при заданном наборе параметров и различных значениях параметра σ .

Заключение

В предложенной работе рассмотрена RQ-система с N типами вызываемых заявок. Были найдены асимптотики первого и второго порядков числа поступивших заявок в системе в асимптотическом условии большой задержки на орбите. На основе полученных асимптотик построена гауссовская аппроксимация распределения вероятностей числа поступивших заявок в системе. Численные результаты показывают, что точность гауссовской аппроксимации растет с уменьшением параметра σ .

ЛИТЕРАТУРА

1. Artalejo J.R., Gómez-Corral A. Retrial queueing systems: a computational approach. Berlin : Springer, 2008. 320 p.
2. Falin G., Templeton J.G.C. Retrial queues. CRC Press, 1997. 320 p.
3. Bhulai S., Koole G. A queueing model for call blending in call centers // IEEE Transactions on Automatic Control. 2003. V. 48 (8). P. 1434–1438.
4. Aguir S., Karaesmen F., Akşin O.Z., Chauvet F. The impact of retrials on call center performance // OR Spectrum. 2004. V. 26 (3). P. 353–376.
5. Artalejo J.R., Phung-Duc T. Markovian retrial queues with two-way communication // Journal of industrial and management optimization. 2012. V. 8 (4). P. 781–806.
6. Artalejo J.R., Phung-Duc T. Single server retrial queues with two-way communication // Applied Mathematical Modelling. 2013. V. 37 (4). P. 1811–1822.
7. Phung-Duc T., Rogiest W. Two-way communication retrial queues with balanced call blending // Lecture Notes in Computer Science. 2012. V. 7314. P. 16–31.
8. Sakurai H., Phung-Duc T. Two-way communication retrial queues with multiple types of outgoing calls // TOP: An Official Journal of the Spanish Society of Statistics and Operations Research. 2015. V. 23 (2). P. 466–492.
9. Nazarov A.A., Paul S.V., Gudkova I. Asymptotic analysis of Markovian retrial queue with two-way communication under low rate of retrials condition. // 31th European Conference on Modelling and Simulation. 2017. P. 687–693.
10. Nazarov A., Phung-Duc T., Paul S. Heavy Outgoing Call Asymptotics for MMPP/M/1/1 Retrial Queue with Two-Way Communication // Communications in Computer and Information Science. 2017. V. 800. P. 28–41.

Поступила в редакцию 9 декабря 2018 г.

Nazarov A.A., Paul S.V., Lizyura O.D. (2019) ASYMPTOTIC ANALYSIS OF RETRIAL QUEUE WITH N TYPES OF OUTGOING CALLS UNDER LOW RATE OF RETRIALS CONDITION. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naja tehnika i informatika* [Tomsk State University Journal of Control and Computer Science]. 48. pp. 13–20

DOI: 10.17223/19988605/48/2

In this paper, we consider Markovian retrial queue with two-way communication and multiple types of outgoing calls, which could be used as a mathematical model of a call center operator. Incoming calls arrive at system according to a Poisson process with rate λ . Service times of incoming calls follow the exponential distribution with rate μ_1 . Upon arrival, an incoming call either occupies the server if it is idle or joins an orbit if the server is busy. Incoming calls stay in orbit for exponentially distributed time with rate σ . From the orbit, an incoming call retries to occupy the server and behaves the same as a fresh incoming call.

On the other hand, the server makes outgoing calls after some exponentially distributed idle time. We assume that there are N types of outgoing calls whole durations follow N distinct distributions.

We consider a random process of the number of incoming calls at the system. The aim of the research is to derive an asymptotic stationary characteristic function of this process under the low rate of retrials condition and to find the parameters of the stationary distribution of this process. To use the asymptotic analysis method we have obtained the Kolmogorov equation system for probability distribution of a 2-dimensional random process of the number of incoming calls in the system and the state of the server. We have also converted the Kolmogorov equation system for probabilities to the Kolmogorov equation system for the partial characteristic functions.

We derived the explicit expression for the characteristic function of the number of incoming calls in the system and discovered that it is difficult to apply this result. We then extend the study to use the asymptotic analysis method under the low rate of retrials limit condition to research the model.

The first order asymptotic only defines the distribution of probabilities of the server state r_k and the mean value κ_1 of the random process of the number of incoming calls in the system. The second order asymptotic shows that the asymptotic probability distribution of the number of incoming calls in the system is Gaussian with the mean κ_1/σ and variance κ_2/σ .

Based on the obtained asymptotic, we have built the Gaussian approximation of the probability distribution of the number of incoming calls in the system. Our numerical results have revealed that the accuracy of Gaussian approximation increases while decreasing σ .

Keywords: retrial queue; outgoing calls; asymptotic analysis method; low rate of retrials condition; Gaussian approximation.

NAZAROV Anatoly Andreevich (Doctor of Technical Sciences, Professor, National Research Tomsk State University, Tomsk, Russian Federation).

E-mail: nazarov.tsu@gmail.com

PAUL Svetlana Vladimirovna (Candidate of Physics and Mathematics, Associate Professor, National Research Tomsk State University, Tomsk, Russian Federation).

E-mail: paulsv82@mail.ru

LIZYURA Olga Dmitrievna (National Research Tomsk State University, Tomsk, Russian Federation).
E-mail: oliztsu@mail.ru

REFERENCES

1. Artalejo, J.R. & Gómez-Corral, A. (2008) *Retrial queueing systems: a computational approach*. Berlin: Springer.
2. Falin, G. & Templeton, J.G.C. (1997) *Retrial Queues*. CRC Press.
3. Bhulai, S. & Koole, G. (2003) A queueing model for call blending in call centers. *IEEE Transactions on Automatic Control*. 48(8). pp. 1434–1438. DOI: 10.1109/TAC.2003.815038
4. Aguir, S., Karaesmen, F., Akşin, O.Z., & Chauvet, F. (2004) The impact of retrials on call center performance. *OR Spectrum*. 26(3). pp. 353–376. DOI: 10.1007/s00291-004-0165-7
5. Artalejo, J. R. & Phung-Duc, T. (2012) Markovian retrial queues with two-way communication. *Journal of Industrial and Management Optimization*. 8(4). pp. 781–806. DOI: 10.3934/jimo.2012.8.781
6. Artalejo, J.R. & Phung-Duc, T. (2013) Single server retrial queues with two-way communication. *Applied Mathematical Modelling*. 37(4). pp. 1811–1822. DOI: 10.1016/j.apm.2012.04.022
7. Phung-Duc, T. & Rogiest, W. (2012) Two-way communication retrial queues with balanced call blending. *Lecture Notes in Computer Science*. 7314. pp. 16–31.
8. Sakurai, H. & Phung-Duc, T. (2015) Two-way communication retrial queues with multiple types of outgoing calls. *TOP: An Official Journal of the Spanish Society of Statistics and Operations Research*. 23(2). pp. 466–492. DOI: 10.1007/s11750-014-0349-5
9. Nazarov, A.A., Paul, S.V. & Gudkova, I. (2017) Asymptotic analysis of Markovian retrial queue with two-way communication under low rate of retrials condition. *31th European Conference on Modelling and Simulation*. pp. 687–693. DOI: 10.7148/2017-0687
10. Nazarov, A., Phung-Duc, T. & Paul, S. (2017) Heavy Outgoing Call Asymptotics for MMPP/M/1/1 Retrial Queue with Two-Way Communication. *Communications in Computer and Information Science*. 800. pp. 28–41. DOI: 10.1007/978-3-319-68069-9_3

УДК 519.21

DOI: 10.17223/19988605/48/3

Л.А. Нежелская, Е.Ф. Сидорова

**ОЦЕНКА ДЛИТЕЛЬНОСТИ НЕПРОДЛЕВАЮЩЕГОСЯ МЕРТВОГО ВРЕМЕНИ
В КОРРЕЛИРОВАННОМ ОБОБЩЕННОМ СИНХРОННОМ ПОТОКЕ
ВТОРОГО ПОРЯДКА**

Рассматривается задача оценивания длительности непродлевающегося мертвого времени в обобщенном синхронном потоке событий второго порядка с использованием метода моментов. Оценивание выполнено с учетом того, что мертвое время является параметром плотности вероятности значений длительности интервала между соседними событиями в наблюдаемом потоке. С помощью модели потока, имитирующей его функционирование при искажающем факторе, исследуется качество оценок в рамках выбранных критериев.

Ключевые слова: обобщенный синхронный дважды стохастический поток второго порядка; непродлевающееся мертвое время; плотность вероятности; оценка параметра; метод моментов.

В настоящей статье проводится дальнейшее исследование относящегося к классу дважды стохастических потоков [1–8] и являющегося математической моделью поступающих в системы и сети массового обслуживания (СМО и СеМО) информационных потоков заявок [9] обобщенного синхронного потока событий второго порядка (далее потока), начатое в работах [10–12].

Режим функционирования как СМО, так и СеМО непосредственно зависит от параметров потока и состояния сопровождающего его процесса. Зачастую в реальных ситуациях эти характеристики либо лишь частично известны, либо неизвестны, либо изменяются со временем случайным образом, вследствие чего возникают задачи оценивания в произвольный момент времени состояний входящего потока (фильтрации его интенсивности) [10, 11, 13] и его параметров [12, 14, 15] по наблюдениям за ним. Математическая модель может также содержать искажающие факторы, к числу которых, в частности, относится мертвое время регистрирующих приборов [16, 17], порождаемое каждым зарегистрированным событием так, что последующие события исходного потока, наступившие в течение обозначенного периода времени, не вызывают его продления (непродлевающееся мертвое время) [11, 14] и теряются для наблюдения.

На предмет установления количества потерянных событий потока, возникающих ввиду эффекта мертвого времени, необходимо оценить его длительность; с этой целью в данной работе применяется метод моментов, обеспечивающий построение оценок, обладающих достаточно хорошими свойствами при больших объемах выборок наблюдаемых моментов наступления событий.

1. Математическая модель потока

Рассматривается обобщенный синхронный поток событий второго порядка, сопровождающий случайный процесс которого $\lambda(t)$ является кусочно-постоянным с двумя состояниями S_1 и S_2 ; здесь и далее S_i понимается как i -е состояние $\lambda(t)$ и имеет место при $\lambda(t) = \lambda_i$, $i = 1, 2$, $\lambda_1 > \lambda_2 \geq 0$.

Длительность интервала между событиями потока в i -м состоянии определяется случайной величиной $\eta_i = \min(\xi_i^{(1)}, \xi_i^{(2)})$, где случайные величины $\xi_i^{(1)}$ и $\xi_i^{(2)}$ независимы и распределены по

законам $F_i^{(1)}(t) = 1 - e^{-\lambda_i t}$ и $F_i^{(2)}(t) = 1 - e^{-\alpha_i t}$ соответственно. В момент наступления события потока процесс $\lambda(t)$ либо переходит из i -го состояния в j -е, $i \neq j$, либо остается в i -м состоянии, $i = j$, с вероятностью $P_1^{(1)}(\lambda_j | \lambda_i)$ или $P_1^{(2)}(\lambda_j | \lambda_i)$, $i, j = 1, 2$, в зависимости от значения η_i , $i = 1, 2$. При этом $P_1^{(1)}(\lambda_j | \lambda_i) + P_1^{(1)}(\lambda_i | \lambda_i) = 1$, $P_1^{(2)}(\lambda_j | \lambda_i) + P_1^{(2)}(\lambda_i | \lambda_i) = 1$, $i, j = 1, 2$, $i \neq j$. Процесс $\lambda(t)$ является скрытым марковским [10] с матрицами инфинитезимальных характеристик

$$\mathbf{D}_0 = \begin{vmatrix} -(\lambda_1 + \alpha_1) & 0 \\ 0 & -(\lambda_2 + \alpha_2) \end{vmatrix}, \mathbf{D}_1 = \begin{vmatrix} \lambda_1 P_1^{(1)}(\lambda_1 | \lambda_1) + \alpha_1 P_1^{(2)}(\lambda_1 | \lambda_1) & \lambda_1 P_1^{(1)}(\lambda_2 | \lambda_1) + \alpha_1 P_1^{(2)}(\lambda_2 | \lambda_1) \\ \lambda_2 P_1^{(1)}(\lambda_1 | \lambda_2) + \alpha_2 P_1^{(2)}(\lambda_1 | \lambda_2) & \lambda_2 P_1^{(1)}(\lambda_2 | \lambda_2) + \alpha_2 P_1^{(2)}(\lambda_2 | \lambda_2) \end{vmatrix}.$$

Элементами матрицы \mathbf{D}_1 выступают интенсивности сопряженных с наступлением события потока переходов процесса $\lambda(t)$ из состояния в состояние. Недиagonальные элементы \mathbf{D}_0 имеют смысл интенсивностей переходов из состояния в состояние без наступления события; диагональные – интенсивностей выхода $\lambda(t)$ из своих состояний, взятые с противоположным знаком [5].

Имеет место ситуация недоступности наблюдению всех событий потока – каждое зарегистрированное в момент времени t_k событие порождает непродлевающееся мертвое время фиксированной длительности T , в течение которого другие события исходного потока теряются, а по его окончании первое наступившее событие вновь вызывает период ненаблюдаемости длительности T и т.д. Вариант возникающей ситуации приведен на рис. 1, где S_1, S_2 – состояния случайного процесса $\lambda(t)$, $t_1, t_2, \dots, t_k, \dots$ – моменты наступления событий в наблюдаемом потоке, штриховкой показаны периоды мертвого времени; черными кружками обозначены потерянные события обобщенного синхронного потока второго порядка, белыми – наблюдаемые события.

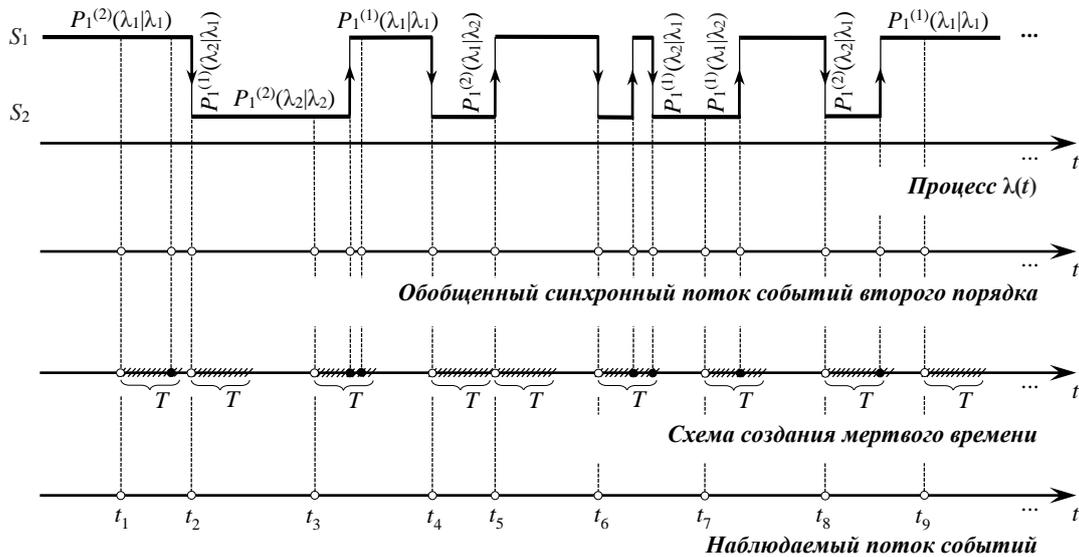


Рис. 1. Формирование наблюдаемого потока событий
 Fig. 1. Formation of the observed event flow

Поскольку рассматривается установившийся (стационарный) режим функционирования исследуемого дважды стохастического потока событий (наблюдение за ним начинается тогда, когда он функционирует бесконечно долго), в силу сформулированных предпосылок последовательность заключающих в себе всю доступную информацию о потоке наблюдаемых моментов $t_1, t_2, \dots, t_k, \dots$ образует вложенную цепь Маркова $\{\lambda(t_k)\}$, т.е. наблюдаемый поток событий обладает марковским свойством, если его эволюцию рассматривать с момента t_k , $k = 1, 2, \dots$

2. Вывод плотности вероятности значений длительности интервала между событиями в потоке

Определим значение длительности k -го интервала между событиями t_k и t_{k+1} , $k=1,2,\dots$, как $\tau_k = t_{k+1} - t_k$, $\tau_k \geq 0$. Для плотности вероятности значения τ_k , ввиду рассмотрения стационарного режима, справедливо $p_T(\tau_k) = p_T(\tau)$, $\tau \geq 0$, $k \geq 1$, что позволяет без ограничения общности положить t_k равным нулю или, что то же самое, момент наступления события в наблюдаемом потоке есть $\tau = 0$; индекс T подчеркивает зависимость плотности от длительности мертвого времени.

Рассмотрим интервал $(0, \tau)$ длительности $\tau = T + t$ между соседними событиями наблюдаемого потока; t – значение длительности интервала между моментом окончания мертвого времени фиксированной длительности T и моментом наступления следующего события, $t > 0$. Введем $q_{ij}(T)$ – вероятность того, что за мертвое время длительности T процесс $\lambda(\tau)$ перейдет из i -го состояния S_i (имеет место в момент $\tau = 0$) в j -е S_j (имеет место в момент $\tau = T$), $i, j = 1, 2$; $\pi_i(0|T)$ – условную стационарную вероятность того, что процесс $\lambda(\tau)$ в момент времени $\tau = 0$ пребывает в состоянии S_i , $i = 1, 2$, при условии, что $\tau = 0$ есть момент наступления события наблюдаемого потока и порождения мертвого времени длительности T .

На интервале $(0, t)$, когда обобщенный синхронный поток событий второго порядка вновь наблюдаем, его поведение описывается условной вероятностью $p_{jk}(t)$ того, что на $(0, t)$ нет событий потока и $\lambda(t) = \lambda_k$ при условии, что $\lambda(0) = \lambda_j$, $j, k = 1, 2$; соответствующую $p_{jk}(t)$ плотность запишем как $\tilde{p}_{jk}(t)$, $j, k = 1, 2$. Тогда искомая плотность вероятности $p_T(\tau)$ запишется в виде:

$$p_T(\tau) = \begin{cases} 0, & 0 \leq \tau < T, \\ \sum_{i=1}^2 \pi_i(0|T) \sum_{j=1}^2 q_{ij}(T) \sum_{k=1}^2 \tilde{p}_{jk}(\tau - T), & \tau \geq T. \end{cases} \quad (1)$$

Для исследуемого потока справедливы следующие леммы.

Лемма 1. Переходные вероятности $q_{ij}(T)$, $i, j = 1, 2$, в коррелированном обобщенном синхронном потоке событий второго порядка с мертвым временем имеют вид:

$$q_{11}(T) = \pi_1 + \pi_2 e^{-\varphi T}, \quad q_{12}(T) = \pi_2 - \pi_2 e^{-\varphi T}, \quad q_{21}(T) = \pi_1 - \pi_1 e^{-\varphi T}, \quad q_{22}(T) = \pi_2 + \pi_1 e^{-\varphi T}, \quad (2)$$

где $\varphi_i = \lambda_i P_1^{(1)}(\lambda_j | \lambda_i) + \alpha_i P_1^{(2)}(\lambda_j | \lambda_i)$, $i, j = 1, 2$, $i \neq j$, $\varphi = \varphi_1 + \varphi_2$, $\pi_1 = \varphi_2 \varphi^{-1}$, $\pi_2 = \varphi_1 \varphi^{-1}$.

Доказательство. Для введенных вероятностей имеют место дифференциальные уравнения

$$q'_{i1}(\tau) = -\varphi_1 q_{i1}(\tau) + \varphi_2 q_{i2}(\tau), \quad q'_{i2}(\tau) = \varphi_1 q_{i1}(\tau) - \varphi_2 q_{i2}(\tau), \quad i = 1, 2,$$

с начальными условиями $q_{11}(0) = q_{22}(0) = 1$, $q_{12}(0) = q_{21}(0) = 0$. Интегрируя полученные системы [19] и заменяя в решении момент времени τ на T , приходим к (2). Лемма 1 доказана.

Найдем p_{jk} – вероятность перехода процесса $\lambda(\tau)$ из состояния S_j в состояние S_k , $j, k = 1, 2$, за время, которое пройдет от момента $t = 0$ до момента наступления следующего события потока.

Лемма 2. В потоке с мертвым временем вероятности p_{jk} , $j, k = 1, 2$, определяются формулами

$$p_{11} = 1 - \varphi_1 z_1^{-1}, \quad p_{12} = \varphi_1 z_1^{-1}, \quad p_{21} = \varphi_2 z_2^{-1}, \quad p_{22} = 1 - \varphi_2 z_2^{-1}, \quad (3)$$

где $z_1 = \lambda_1 + \alpha_1$, $z_2 = \lambda_2 + \alpha_2$; φ_1, φ_2 определены в (2).

Доказательство. Условные вероятности $p_{jk}(t)$, $j, k = 1, 2$, идентичны по своему смысловому значению вероятностям $p_{ij}(\tau)$, $i, j = 1, 2$, для исследуемого потока в случае отсутствия мертвого времени, т.е. определяются теми же формулами [12] при $\tau = t$. Согласно определению потока установ-

лена величина $p_{jj}(t)(\lambda_j P_1^{(1)}(\lambda_k | \lambda_j) + \alpha_j P_1^{(2)}(\lambda_k | \lambda_j))\Delta t + o(\Delta t)$, которая является совместной вероятностью того, что без наступления события наблюдаемого потока на интервале $(0, t)$ процесс $\lambda(t)$ переходит на этом интервале из состояния S_j в состояние S_k , $j, k = 1, 2$, на полуинтервале $[t, t + \Delta t)$ происходит окончание состояния S_k процесса $\lambda(t)$, и в результате розыгрыша пар в момент наступления события $\lambda(t)$ переходит из S_k в S_j , $j, k = 1, 2$. Эта величина представима в виде:

$$p_{jj}(t)(\lambda_j P_1^{(1)}(\lambda_k | \lambda_j) + \alpha_j P_1^{(2)}(\lambda_k | \lambda_j))\Delta t + o(\Delta t) = \int_t^{t+\Delta t} \tilde{p}_{jk}(u) du = \tilde{p}_{jk}(t)\Delta t + o(\Delta t),$$

где $\tilde{p}_{jk}(t)$ – соответствующая вероятности $p_{jk}(t)$ плотность вероятности, $j, k = 1, 2$.

Тогда плотности $\tilde{p}_{jk}(t)$ того, что без наступления события на $(0, t)$ и наступления события в момент t процесс $\lambda(t)$ перейдет из состояния S_j в состояние S_k , $j, k = 1, 2$, запишутся в виде [12]:

$$\tilde{p}_{jk}(t) = (\lambda_j P_1^{(1)}(\lambda_k | \lambda_j) + \alpha_j P_1^{(2)}(\lambda_k | \lambda_j))e^{-z_j t}. \quad (4)$$

В силу произвольности момента t вероятности перехода p_{jk} , $j, k = 1, 2$, определяются как

$$p_{jk} = \int_0^{\infty} \tilde{p}_{jk}(t) dt. \quad (5)$$

Подставляя (4) в (5) и выполняя интегрирование, получим (3). Лемма 2 доказана.

Лемма 3. Условные стационарные вероятности $\pi_i(0|T)$, $i = 1, 2$, задаются выражениями

$$\pi_1(0|T) = \frac{z_1 \phi_2 + \pi_1(z_1 z_2 - z_1 \phi_2 - z_2 \phi_1)(1 - e^{-\phi T})}{z_1 \phi_2 + z_2 \phi_1 + (z_1 z_2 - z_1 \phi_2 - z_2 \phi_1)(1 - e^{-\phi T})}, \quad \pi_2(0|T) = 1 - \pi_1(0|T), \quad (6)$$

где $\phi_1, \phi_2, \phi, \pi_1, \pi_2$ определены в (2), z_1, z_2 – в (3).

Доказательство. Поскольку в моменты $t_1, t_2, \dots, t_k, \dots$ последовательность $\{\lambda(t_k)\}$ представляет собой вложенную цепь Маркова, для вероятностей $\pi_i(0|T)$, $i = 1, 2$, справедливы уравнения

$$\pi_1(0|T) = \pi_1(0|T)\pi_{11}(T) + \pi_2(0|T)\pi_{21}(T), \quad \pi_2(0|T) = \pi_1(0|T)\pi_{12}(T) + \pi_2(0|T)\pi_{22}(T), \quad (7)$$

где $\pi_{ik}(T)$ – переходная вероятность того, что за время, которое пройдет от момента $\tau = 0$ до наступления следующего события потока, $\lambda(\tau)$ перейдет из состояния S_i в состояние S_k , $i, k = 1, 2$.

В силу марковости $\lambda(t)$ вероятности $q_{ij}(T)$ и p_{jk} , определенные в (2) и (3) соответственно, $i, j, k = 1, 2$, позволяют записать для $\pi_{ik}(T)$, $i, k = 1, 2$, следующие выражения:

$$\begin{aligned} \pi_{11}(T) &= q_{11}(T)p_{11} + q_{12}(T)p_{21}, \quad \pi_{12}(T) = q_{11}(T)p_{12} + q_{12}(T)p_{22}, \quad \pi_{11}(T) + \pi_{12}(T) = 1, \\ \pi_{21}(T) &= q_{21}(T)p_{11} + q_{22}(T)p_{21}, \quad \pi_{22}(T) = q_{21}(T)p_{12} + q_{22}(T)p_{22}, \quad \pi_{21}(T) + \pi_{22}(T) = 1. \end{aligned} \quad (8)$$

Подставляя в (8) переходные вероятности $q_{ij}(T)$, $i, j = 1, 2$, найденные в виде (2), получим

$$\pi_{1k}(T) = p_{1k} - \pi_2(p_{1k} - p_{2k})(1 - e^{-\phi T}), \quad \pi_{2k}(T) = p_{2k} + \pi_1(p_{1k} - p_{2k})(1 - e^{-\phi T}). \quad (9)$$

Далее, подставляя (9) и (3) в уравнения (7), приходим к (6). Лемма 3 доказана.

На основании лемм 1–3 сформулируем теорему.

Теорема. В коррелированном потоке, функционирующем в условиях непродлевающегося мертвого времени, плотность вероятности длительности интервала между событиями имеет вид:

$$P_T(\tau) = \begin{cases} 0, & 0 \leq \tau < T, \\ \gamma(T)z_1 e^{-z_1(\tau-T)} + (1 - \gamma(T))z_2 e^{-z_2(\tau-T)}, & \tau \geq T, \end{cases} \quad (10)$$

где $\gamma(T) = \pi_1 - \pi_1 \phi_1 (z_1 - z_2)(z_1 z_2 - z_1 \phi_2 - z_2 \phi_1 - z_1 z_2 e^{\phi T})^{-1}$; $\phi_1, \phi_2, \phi, \pi_1$ определены в (2), z_1, z_2 – в (3).

Доказательство. Подставляя в (1) сначала (4), затем (2) и, наконец, (6), проделывая достаточно трудоемкие преобразования с учетом $t = \tau - T$, приходим к (10). Теорема доказана.

Замечание 1. Равенство $z_1 = z_2 = z$, которое вполне может выполняться в силу того, что λ_1, λ_2 удовлетворяют условию $\lambda_1 > \lambda_2 \geq 0$, а α_1, α_2 , вообще говоря, произвольные неотрицательные числа, приводит к $p_T(\tau) = ze^{-z(\tau-T)}$; поток вырождается в простейший. В дальнейшем принимается $z_1 \neq z_2$.

Отметим, что приведенная в [12] формула для плотности – частный случай (10) при $T = 0$. Нетрудно по $p(\tau)$ [12] и $p_T(\tau)$ вида (10) получить выражения для среднего числа событий в единицу времени в обобщенном синхронном потоке второго порядка в условиях полной наблюдаемости Λ и при непродлевающемся мертвом времени Λ_T соответственно. На их основе при $\phi_1, \phi_2, \phi, \pi_1, \pi_2$ и z_1, z_2 , определенных в (2) и (3), запишем среднее число потерянных событий в единицу времени:

$$\Delta = \Lambda - \Lambda_T = \frac{\pi_1 \pi_2 (z_1 - z_2)^2 (1 - e^{\phi T}) + T(z_1 \pi_1 + z_2 \pi_2)(z_1 z_2 - z_1 \phi_2 - z_2 \phi_1 - z_1 z_2 e^{\phi T})}{(z_1 \pi_2 + z_2 \pi_1)(1 - e^{\phi T}) + T(z_1 z_2 - z_1 \phi_2 - z_2 \phi_1 - z_1 z_2 e^{\phi T}) - \phi}, \quad (11)$$

где $\Lambda = z_1 \pi_1 + z_2 \pi_2$, $\Lambda_T = \frac{z_1 z_2 - z_1 \phi_2 - z_2 \phi_1 - z_1 z_2 e^{\phi T}}{(z_1 \pi_2 + z_2 \pi_1)(1 - e^{\phi T}) + T(z_1 z_2 - z_1 \phi_2 - z_2 \phi_1 - z_1 z_2 e^{\phi T}) - \phi}$.

3. Оценка длительности мертвого времени в наблюдаемом потоке

Для оценки периода ненаблюдаемости обратимся к методу моментов [18]: рассмотрим выборку $\tau_1, \tau_2, \dots, \tau_n$ из распределения $p_T(\tau) = \gamma(T)z_1 e^{-z_1(\tau-T)} + (1 - \gamma(T))z_2 e^{-z_2(\tau-T)}$, $\tau \geq T$, зависящего от параметра T . В силу близости теоретической и эмпирической функций распределения (неограниченного сближения их значений при достаточно большом n) следует ожидать близости теоретического начального момента первого порядка $M\tau = \int_T^\infty \tau p_T(\tau) d\tau$ и статистики $C_1 = \frac{1}{n} \sum_{k=1}^n \tau_k$, т.е. чтобы найти \hat{T} , необходимо решить уравнение моментов $M\tau = C_1$, принимающее вид:

$$T + (z_2 - z_1)\gamma(T)z_1^{-1}z_2^{-1} + z_2^{-1} = C_1. \quad (12)$$

Решение (12) возможно только с применением численных методов.

Замечание 2. Для $f(T) = M\tau$: $f'(T) = 1 + \phi_1 \phi_2 (z_1 - z_2)^2 e^{\phi T} (z_1 z_2 - z_1 \phi_2 - z_2 \phi_1 - z_1 z_2 e^{\phi T})^{-2} > 0$, $f(0) = (z_1 \pi_1 + z_2 \pi_2)^{-1} > 0$. Тогда $f(T)$ – возрастающая функция переменной T , $T \geq 0$.

Согласно замечанию 2 оценка длительности мертвого времени $\hat{T}_{\text{ЧМ}}$ (численное решение (12)) на полуинтервале $(0, \tau_{\min}]$, где $\tau_{\min} = \min \tau_k$, $k=1, 2, \dots, n$, определяется единственным образом: $\hat{T}_{\text{ЧМ}} = \hat{T}$, если $f(0) < C_1 < f(\tau_{\min})$; $\hat{T}_{\text{ЧМ}} = \tau_{\min}$, если $f(0) < f(\tau_{\min}) \leq C_1$; $\hat{T}_{\text{ЧМ}} = 0$, если $f(0) \geq C_1$.

Рассмотрим подход, обеспечивающий аналитическое решение \hat{T} : воспользуемся замечанием, сформулированным в [18] и позволяющим использовать одноименные центральные моменты – дисперсию $D\tau = \int_T^\infty \tau^2 p_T(\tau) d\tau - (M\tau)^2$ и выборочную дисперсию $C_2 - C_1^2$, где $C_2 = \frac{1}{n} \sum_{k=1}^n \tau_k^2$. В данном случае преобразованное уравнение моментов $D\tau = C_2 - C_1^2$ для оценивания T запишется в виде:

$$(z_2 - z_1)^2 \gamma^2(T) - 2z_2(z_2 - z_1)\gamma(T) - z_1^2(1 + C_1^2 z_2^2 - C_2 z_2^2) = 0. \quad (13)$$

При решении (13), с учетом замечания 1, возможны следующие варианты.

1. $z_2^2 + z_1^2(1 + C_1^2 z_2^2 - C_2 z_2^2) > 0$:

1.1) $z_1 > z_2$, $1 + C_1^2 z_2^2 - C_2 z_2^2 > 0$, тогда $\hat{\gamma}(T) = \left(z_2 - \sqrt{z_2^2 + z_1^2(1 + C_1^2 z_2^2 - C_2 z_2^2)} \right) (z_2 - z_1)^{-1}$;

1.2) $z_2 > z_1$, тогда $\hat{\gamma}(T) = \left(z_2 + \sqrt{z_2^2 + z_1^2(1 + C_1^2 z_2^2 - C_2 z_2^2)} \right) (z_2 - z_1)^{-1}$;

1.3) $z_2 > z_1$, $1 + C_1^2 z_2^2 - C_2 z_2^2 < 0$, тогда $\hat{\gamma}_{1,2}(T) = (z_2 \mp \sqrt{z_2^2 + z_1^2(1 + C_1^2 z_2^2 - C_2 z_2^2)}) (z_2 - z_1)^{-1}$.

2. $z_2^2 + z_1^2(1 + C_1^2 z_2^2 - C_2 z_2^2) = 0$:

2.1) $z_1 > z_2$, тогда $\hat{\gamma}(T) = z_2(z_2 - z_1)^{-1} < 0$ и корень исключается из рассмотрения;

2.2) $z_2 > z_1$, тогда $\hat{\gamma}(T) = z_2(z_2 - z_1)^{-1}$.

3. $z_2^2 + z_1^2(1 + C_1^2 z_2^2 - C_2 z_2^2) < 0$: действительных корней нет, увеличивается выборка.

Так, $\hat{T} = \varphi^{-1} \ln |1 - \varphi_1 z_1^{-1} - \varphi_2 z_2^{-1} - \pi_1 \varphi_1 (z_1 - z_2) z_1^{-1} z_2^{-1} (\pi_1 - \hat{\gamma}(T))^{-1}|$ при единственном корне (13), и полученная аналитически оценка мертвого времени \hat{T}_A есть: $\hat{T}_A = \hat{T}$, если $0 < \hat{T} < \tau_{\min}$; $\hat{T}_A = \tau_{\min}$, если $\hat{T} > \tau_{\min}$; $\hat{T}_A = 0$, если $\hat{T} < 0$. При двух действительных корнях уравнения моментов $\hat{T}_{1,2} = \varphi^{-1} \ln |1 - \varphi_1 z_1^{-1} - \varphi_2 z_2^{-1} - \pi_1 \varphi_1 (z_1 - z_2) z_1^{-1} z_2^{-1} (\pi_1 - \hat{\gamma}_{1,2}(T))^{-1}|$, и оценка \hat{T}_A устанавливается как $\hat{T}_A = 0,5(\hat{T}_1 + \hat{T}_2)$, если $0 < \hat{T}_2 < \hat{T}_1 < \tau_{\min}$; $\hat{T}_A = \hat{T}_1$, если $\hat{T}_2 < 0 < \hat{T}_1 < \tau_{\min}$; $\hat{T}_A = \hat{T}_2$, если $0 < \hat{T}_2 < \tau_{\min} < \hat{T}_1$; $\hat{T}_A = \tau_{\min}$, если $\hat{T}_2 < 0 < \tau_{\min} < \hat{T}_1$ или $\tau_{\min} < \hat{T}_2 < \hat{T}_1$; $\hat{T}_A = 0$, если $\hat{T}_2 < \hat{T}_1 < 0$.

Замечание 3. Применение τ_{\min} при оценивании T дает улучшенную оценку метода моментов.

4. Результаты численных расчетов

В разделе исследуются численные значения оценок $\hat{T}_{\text{чм}}$ и \hat{T}_A , полученные с использованием имитационной модели. Для проверки их качества по $N = 100$ независимым реализациям при $\lambda_1 = 5,3$, $\lambda_2 = 3,8$, $\alpha_1 = 3,7$, $\alpha_2 = 1,2$, $P_1^{(1)}(\lambda_1 | \lambda_1) = P_1^{(2)}(\lambda_2 | \lambda_1) = 0,3$, $P_1^{(2)}(\lambda_1 | \lambda_1) = P_1^{(1)}(\lambda_2 | \lambda_1) = 0,7$, $P_1^{(1)}(\lambda_2 | \lambda_2) = P_1^{(2)}(\lambda_1 | \lambda_2) = 0,8$, $P_1^{(2)}(\lambda_2 | \lambda_2) = P_1^{(1)}(\lambda_1 | \lambda_2) = 0,2$ вычислялись выборочные средние значения $\hat{M}\{\hat{T}\} = N^{-1} \sum \hat{T}^{(k)}$ и оценки смещения $|\hat{M}\{\hat{T}\} - T|$, $\hat{T} \in \{\hat{T}_{\text{чм}}, \hat{T}_A\}$, $k = \overline{1, N}$, для $T_m = 100, \dots, 1000$ и $T = 0,3, 0,5, 0,7, 0,9$. Результаты соответствующих расчетов приведены в табл. 1–4.

Таблица 1

Результаты статистического эксперимента ($T = 0,3$)

T_m	100	200	300	400	500	600	700	800	900	1000
$\hat{M}\{\hat{T}_{\text{чм}}\}$	0,2977	0,2976	0,2990	0,2991	0,2993	0,2990	0,2994	0,2994	0,2993	0,2994
$ \hat{M}\{\hat{T}_{\text{чм}}\} - T $	0,0023	0,0024	0,0010	0,0009	0,0007	0,0010	0,0006	0,0006	0,0007	0,0006
$\hat{M}\{\hat{T}_A\}$	0,1229	0,2080	0,2027	0,1921	0,2385	0,2018	0,2321	0,2355	0,2263	0,2332
$ \hat{M}\{\hat{T}_A\} - T $	0,1771	0,0920	0,0973	0,1079	0,0615	0,0982	0,0679	0,0645	0,0737	0,0668

Таблица 2

Результаты статистического эксперимента ($T = 0,5$)

T_m	100	200	300	400	500	600	700	800	900	1000
$\hat{M}\{\hat{T}_{\text{чм}}\}$	0,4964	0,4988	0,4985	0,4987	0,4982	0,4992	0,4993	0,4991	0,4995	0,4996
$ \hat{M}\{\hat{T}_{\text{чм}}\} - T $	0,0036	0,0012	0,0015	0,0013	0,0018	0,0008	0,0007	0,0009	0,0005	0,0004
$\hat{M}\{\hat{T}_A\}$	0,3662	0,4581	0,3140	0,3270	0,3471	0,4417	0,3652	0,3415	0,3715	0,3647
$ \hat{M}\{\hat{T}_A\} - T $	0,1338	0,0419	0,1860	0,1730	0,1529	0,0583	0,1348	0,1585	0,1285	0,1353

Таблица 3

Результаты статистического эксперимента ($T = 0,7$)

T_m	100	200	300	400	500	600	700	800	900	1000
$\hat{M}\{\hat{T}_{\text{чм}}\}$	0,6963	0,6971	0,6988	0,6984	0,6985	0,6983	0,6990	0,6989	0,6989	0,6992
$ \hat{M}\{\hat{T}_{\text{чм}}\} - T $	0,0037	0,0029	0,0012	0,0016	0,0015	0,0017	0,0010	0,0011	0,0011	0,0008
$\hat{M}\{\hat{T}_A\}$	0,4012	0,3641	0,5263	0,3880	0,5203	0,5014	0,5832	0,3992	0,3472	0,5448
$ \hat{M}\{\hat{T}_A\} - T $	0,2988	0,3359	0,1737	0,3120	0,1797	0,1986	0,1168	0,3008	0,3528	0,1552

Таблица 4

Результаты статистического эксперимента ($T = 0,9$)

T_m	100	200	300	400	500	600	700	800	900	1000
$\hat{M}\{\hat{T}_{\text{чм}}\}$	0,8967	0,8970	0,8978	0,8986	0,8982	0,8984	0,8987	0,8993	0,8990	0,8990
$ \hat{M}\{\hat{T}_{\text{чм}}\} - T $	0,0033	0,0030	0,0022	0,0014	0,0018	0,0016	0,0013	0,0007	0,0010	0,0010
$\hat{M}\{\hat{T}_A\}$	0,5369	0,5410	0,5122	0,4141	0,5748	0,5450	0,4706	0,5443	0,5872	0,6882
$ \hat{M}\{\hat{T}_A\} - T $	0,3631	0,3590	0,3878	0,4859	0,3252	0,3550	0,4294	0,3557	0,3128	0,2118

При анализе результатов эксперимента, приведенных в табл. 1–4, выявлена сильная зависимость оценок длительности мертвого времени от времени моделирования: с увеличением T_m выборочные средние $\hat{M}\{\hat{T}_{\text{чм}}\}$ стабилизируются, а качество $\hat{T}_{\text{чм}}$ улучшается в смысле уменьшения оценки смещения $|\hat{M}\{\hat{T}_{\text{чм}}\} - T|$, что обусловлено концепцией метода моментов, основывающегося на значениях статистики C_1 , заключающей в себе всю доступную информацию об интервалах между наблюдаемыми моментами; в то же время для $\hat{M}\{\hat{T}_A\}$ отмечается колебательный характер с тенденцией к уменьшению, что объясняется использованием центрального момента для составления (13) и, следовательно, накоплением меньшего объема информации в $C_2 - C_1^2$.

Таблица 5

Результаты статистического эксперимента

T	Λ	$\hat{\Lambda}$	$ \Lambda - \hat{\Lambda} /\Lambda$	Λ_T	$\hat{\Lambda}_T$	$ \Lambda_T - \hat{\Lambda}_T /\Lambda_T$	Δ	$\hat{\Delta}$	$ \Delta - \hat{\Delta} /\Delta$
0,3	6,0520	6,0505	0,000248	2,1044	2,1081	0,001758	3,9476	3,9424	0,001317
0,5		6,0561	0,000677	1,4788	1,4767	0,001420	4,5732	4,5795	0,001378
0,7		6,0447	0,001206	1,1409	1,1459	0,004383	4,9111	4,8999	0,002281
0,9		6,0604	0,001388	0,9286	0,9293	0,000754	5,1231	5,1311	0,001562

Увеличение периода T приводит в среднем к снижению качества оценивания $\hat{T}_{\text{чм}}$ и \hat{T}_A , причиной этого является недоступность наблюдений всех событий: вычисленные для отдельной реализации по формуле (11) при $T_m = 1000$ на основе Λ и Λ_T средние значения потерянных в единицу времени событий Δ отражены в табл. 5, в которой также содержатся статистические оценки $\hat{\Lambda}$, $\hat{\Lambda}_T$, $\hat{\Delta}$ для демонстрации обоснованности использования аппарата имитационного моделирования для проведения расчетов.

Заключение

В данной работе для коррелированного обобщенного синхронного дважды стохастического потока второго порядка при неполной наблюдаемости найден вид плотности вероятности $p_T(\tau)$ длитель-

ности интервала между событиями в наблюдаемом потоке, на основании которой методом моментов получены численная $\hat{T}_{\text{чМ}}$ и аналитическая \hat{T}_A оценки длительности непродлевающегося мертвого времени, выступающего неизвестным параметром плотности вероятности $p_T(\tau)$.

С использованием имитационной модели для конкретного статистического эксперимента исследовано качество построенных оценок: результаты сравнения $\hat{T}_{\text{чМ}}$ и \hat{T}_A определяют предпочтение в использовании $\hat{T}_{\text{чМ}}$ при произвольном задании времени наблюдения за потоком T_m .

ЛИТЕРАТУРА

1. Cox D.R. The analysis of non-Markovian stochastic processes by the inclusion of supplementary variables // Proc. of the Cambridge Philosophical Society. 1955. V. 51, No. 3. P. 433–441.
2. Kingman J.F.C. On doubly stochastic Poisson process // Proc. of the Cambridge Philosophical Society. 1964. V. 60, is. 4. P. 923–930.
3. Cox D.R., Miller H.D. The theory of stochastic processes. New York : Wiley, 1965. 398 p.
4. Башарин Г.П., Кокотушкин В.А., Наумов В.А. О методе эквивалентных замен расчета фрагментов сетей связи. Ч. 1 // Известия АН СССР. Техническая кибернетика. 1979. № 6. С. 92–99.
5. Башарин Г.П., Кокотушкин В.А., Наумов В.А. О методе эквивалентных замен расчета фрагментов сетей связи. Ч. 2 // Известия АН СССР. Техническая кибернетика. 1980. № 1. С. 55–61.
6. Neuts M.F. A versatile Markovian point process // Journal of Applied Probability. 1979. V. 16. P. 764–779.
7. Snyder D.L., Miller M.I. Random point processes in time and space. Heidelberg : Springer-Verlag, 1991. 481 p.
8. Lucantoni D.M. New results on the single server queue with a batch markovian arrival process // Communications in Statistics Stochastic Models. 1991. V. 7. P. 1–46.
9. Дудин А.Н., Клименок В.И. Системы массового обслуживания с коррелированными потоками. Минск : Изд-во Белорус. гос. ун-та, 2000. 175 с.
10. Nezhelskaya L., Sidorova E. Optimal estimation of the states of synchronous generalized flow of events of the second order under its complete observability // Communications in Computer and Information Science. 2018. V. 912. P. 157–171.
11. Нежелская Л.А., Сидорова Е.Ф. Оптимальная оценка состояний обобщенного синхронного потока событий второго порядка в условиях неполной наблюдаемости // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2018. № 45. С. 30–41.
12. Нежелская Л.А., Сидорова Е.Ф. Оценивание параметров плотности вероятности значений длительности интервала между событиями в коррелированном обобщенном синхронном потоке второго порядка // Информационные технологии и математическое моделирование (ИТММ-2019) : материалы XVIII Междунар. конф. им. А.Ф. Терпугова. Томск : Изд-во НТЛ, 2019. Ч. 2. С. 358–363.
13. 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.
14. 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.
15. Okamura H., Dohi T., Trivedi K.S. Markovian arrival process parameter estimation with group data // IEEE/ACM Transactions on Networking (TON). 2009. V. 17, No. 4. P. 1326–1339.
16. Апанасович В.В., Коляда А.А., Чернявский А.Ф. Статистический анализ случайных потоков в физическом эксперименте. Минск : Университетское, 1988. 256 с.
17. Normey-Rico J.E. Control of dead-time process. London : Springer-Verlag, 2007. 462 p.
18. Малинковский Ю.В. Теория вероятностей и математическая статистика. Гомель : УО «ГТУ им. Ф. Скорины», 2004. Ч. 2: Математическая статистика. 146 с.
19. Камке Э. Справочник по обыкновенным дифференциальным уравнениям. М. : Наука, 1976. 576 с.

Поступила в редакцию 11 февраля 2019 г.

Nezhel'skaya L.A., Sidorova E.F. (2019) ESTIMATION OF THE UNEXTENDABLE DEAD TIME DURATION IN CORRELATED SYNCHRONOUS GENERALIZED FLOW OF THE SECOND ORDER. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie vychislitel'naja tehnika i informatika* [Tomsk State University Journal of Control and Computer Science]. 48. pp. 21–30

DOI: 10.17223/19988605/48/3

The paper deals with the estimation of the unobservability period duration in a synchronous generalized doubly stochastic flow of the second order, whose accompanying process $\lambda(t)$ represented by a piecewise constant random process with two states S_1, S_2 ; the state S_i of the process $\lambda(t)$ takes place when $\lambda(t) = \lambda_i, i = 1, 2, \lambda_1 > \lambda_2 \geq 0$.

The flow inter-event interval duration at the i -th state is determined by the random variable $\eta_i = \min(\xi_i^{(1)}, \xi_i^{(2)})$, where random variables $\xi_i^{(1)}$ and $\xi_i^{(2)}$ are both independent and distributed according to the laws $F_i^{(1)}(t) = 1 - e^{-\lambda_1 t}$ and $F_i^{(2)}(t) = 1 - e^{-\lambda_2 t}$, respec-

tively, $i = 1, 2$. At the moment when a flow event occurs, the process $\lambda(t)$ either transits from the i -th state to the j -th state, $i \neq j$, or remains at the i -th state, $i = j$, with probability $P_1^{(1)}(\lambda_j | \lambda_i)$ or $P_1^{(2)}(\lambda_j | \lambda_i)$, $i, j = 1, 2$, depending on the value taken by the random variable η_i , $i = 1, 2$. Here $P_1^{(1)}(\lambda_j | \lambda_i) + P_1^{(1)}(\lambda_i | \lambda_i) = 1$, $P_1^{(2)}(\lambda_j | \lambda_i) + P_1^{(2)}(\lambda_i | \lambda_i) = 1$, $i, j = 1, 2$, $i \neq j$.

The probability density of the values of interval duration between events of the flow operating in a steady-state mode under conditions of incomplete observability has an explicit form

$$P_T(\tau) = \begin{cases} 0, & 0 \leq \tau < T, \\ \gamma(T)z_1 e^{-z_1(\tau-T)} + (1-\gamma(T))z_2 e^{-z_2(\tau-T)}, & \tau \geq T, \end{cases}$$

$$\gamma(T) = \pi_1 - \pi_1 \varphi_1 (z_1 - z_2) (z_1 z_2 - z_1 \varphi_2 - z_2 \varphi_1 - z_1 z_2 e^{\varphi T})^{-1},$$

$$\varphi_1 = \lambda_1 P_1^{(1)}(\lambda_2 | \lambda_1) + \alpha_1 P_1^{(2)}(\lambda_2 | \lambda_1), \quad \varphi_2 = \lambda_2 P_1^{(1)}(\lambda_1 | \lambda_2) + \alpha_2 P_1^{(2)}(\lambda_1 | \lambda_2), \quad \varphi = \varphi_1 + \varphi_2, \quad \pi_1 = \varphi_2 \varphi^{-1}, \quad z_1 = \lambda_1 + \alpha_1, \quad z_2 = \lambda_2 + \alpha_2.$$

Then the average number of lost events in the flow (due to the dead time effect) per unit time is

$$\Delta = \Lambda - \Lambda_T = \frac{\pi_1 \pi_2 (z_1 - z_2)^2 (1 - e^{\varphi T}) + T(z_1 \pi_1 + z_2 \pi_2)(z_1 z_2 - z_1 \varphi_2 - z_2 \varphi_1 - z_1 z_2 e^{\varphi T})}{(z_1 \pi_2 + z_2 \pi_1)(1 - e^{\varphi T}) + T(z_1 z_2 - z_1 \varphi_2 - z_2 \varphi_1 - z_1 z_2 e^{\varphi T}) - \varphi},$$

$$\Lambda = z_1 \pi_1 + z_2 \pi_2, \quad \Lambda_T = \frac{z_1 z_2 - z_1 \varphi_2 - z_2 \varphi_1 - z_1 z_2 e^{\varphi T}}{(z_1 \pi_2 + z_2 \pi_1)(1 - e^{\varphi T}) + T(z_1 z_2 - z_1 \varphi_2 - z_2 \varphi_1 - z_1 z_2 e^{\varphi T}) - \varphi}.$$

The estimation \hat{T}_{QM} of the dead time duration T is determined numerically as a solution of the moment equation

$$T + (z_2 - z_1)\gamma(T)z_1^{-1}z_2^{-1} + z_2^{-1} = C_1,$$

however, as it was revealed during the simulation model testing, it is preferable in sense of smallness of the displacement estimate in comparison with the estimate \hat{T}_A , analytically obtained from the moment equation

$$(z_2 - z_1)^2 \gamma^2(T) - 2z_2(z_2 - z_1)\gamma(T) - z_1^2(1 + C_1^2 z_2^2 - C_2 z_2^2) = 0.$$

Keywords: synchronous generalized doubly stochastic flow of the second order; unextendable dead time; probability density; parameter estimation; method of moments.

NEZHEL'SKAYA Lyudmila Alekseevna (Doctor of Physical and Mathematical Sciences, Associate Professor, National Research Tomsk State University, Tomsk, Russian Federation).

E-mail: ludne@mail.ru

SIDOROVA Ekaterina Filippovna (National Research Tomsk State University, Tomsk, Russian Federation).

E-mail: katusha_sidorova@mail.ru

REFERENCES

1. Cox, D.R. (1955) The analysis of non-Markovian stochastic processes by the inclusion of supplementary variables. *Proceedings of the Cambridge Philosophical Society*. 51(3). pp. 433–441. DOI: 10.1017/S0305004100030437
2. Kingman, J.F.C. (1964) On doubly stochastic Poisson process. *Proceedings of the Cambridge Philosophical Society*. 60(4). pp. 923–930. DOI: 10.1017/S030500410003838X
3. Cox, D.R. & Miller, H.D. (1965) *The Theory of Stochastic Processes*. New York: Wiley.
4. Basharin, G.P., Kokotushkin, V.A. & Naumov, V.A. (1979) O metode ekvivalentnykh zamen rascheta fragmentov setey svyazi. Ch. 1 [On the equivalent substitutions method for computing fragments of communication networks. Part 1]. *Izvestiya AN SSSR. Tekhnicheskaya kibernetika – Proceedings of USSR Academy of Sciences. Technical cybernetics*. 6. pp. 92–99.
5. Basharin, G.P., Kokotushkin, V.A. & Naumov, V.A. (1980) O metode ekvivalentnykh zamen rascheta fragmentov setey svyazi. Ch. 2 [On the equivalent substitutions method for computing fragments of communication networks. Part 2]. *AN SSSR. Tekhnicheskaya kibernetika – Proceedings of USSR Academy of Sciences. Technical cybernetics*. 1. pp. 55–61.
6. Neuts, M.F. (1979) A versatile Markovian point process. *Journal of Applied Probability*. 16. pp. 764–779. DOI: 10.2307/3213143
7. Snyder, D.L. & Miller, M.I. (1991) *Random Point Processes in Time and Space*. Heidelberg: Springer-Verlag.
8. Lucantoni, D.M. (1991) New results on the single server queue with a batch markovian arrival process. *Communications in Statistics Stochastic Models*. 7. pp. 1–46. DOI: 10.1080/15326349108807174
9. Dudin, A.N. & Klimenok, V.I. (2000) *Sistemy massovogo obsluzhivaniya s korrelirovannymi potokami* [Queueing systems with correlated flows]. Minsk: Belarusian State University.
10. Nezhelskaya, L. & Sidorova, E. (2018) Optimal estimation of the states of synchronous generalized flow of events of the second order under its complete observability. *Communications in Computer and Information Science*. 912. pp. 157–171. DOI: 10.1007/978-3-319-97595-5_13

11. Nezhelskaya, L.A. & Sidorova, E.F. (2018) Optimal estimate of the states of a generalized synchronous flow of second-order events under conditions of incomplete observability. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science*. 45. pp. 30–41. (In Russian). DOI: 10.17223/19988605/45/4
12. Nezhelskaya, L.A. & Sidorova, E.F. (2019) [Estimation of the probability density parameters of the values of inter-event interval duration in correlated synchronous generalized flow of the second order]. *Informatsionnye tekhnologii i matematicheskoe modelirovanie (ITMM-2019)* [Information Technologies and Mathematical Modelling (ITMM-2019)]. Proc. of the 18th International Conference named after A.F. Terpugov. Part 2. Tomsk: NTL. pp. 358–363. (In Russian).
13. 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.
14. 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.
15. Okamura, H., Dohi, T. & Trivedi, K.S. (2009) Markovian arrival process parameter estimation with group data. *IEEE/ACM Transactions on Networking (TON)*. 17(4). pp. 1326–1339.
16. Apanasovich, V.V., Kolyada, A.A. & Chernyavskii, A.F. (1988) *Statisticheskii analiz sluchaynykh potokov v fizicheskom eksperimente* [The Statistical Analysis of Series of Random Events in Physical Experiment]. Minsk: Universitetskoe.
17. Normey-Rico, J.E. (2007) *Control of dead-time process*. London: Springer-Verlag.
18. Malinkovsky, Yu.V. (2004) *Teoriya veroyatnostey i matematicheskaya statistika* [Probability Theory and Mathematical Statistics]. Gomel: Francisk Skorina Gomel State University.
19. Kamke, E. (1976) *Spravochnik po obyknovennym differentsial'nym uravneniyam* [Handbook of Ordinary Differential Equations]. Moscow: Nauka.

УДК 62-50

DOI: 10.17223/19988605/48/4

A.I. Rouban

**THE SENSITIVITY FUNCTIONALS IN THE BOLTS'S PROBLEM
FOR MULTIVARIATE DYNAMIC SYSTEMS DESCRIBED
BY INTEGRO-DIFFERENTIAL EQUATIONS WITH DELAY TIME**

The variational method of calculation of sensitivity functionals (connecting first variation of quality functionals with variations of variable parameters) and sensitivity coefficients (components of vector gradient from the quality functional to constant parameters) for multivariate non-linear dynamic systems described by continuous vectorial Volterra's integro-differential equations of the second-kind with delay time is developed. The base of calculation is the decision of corresponding integro-differential conjugate equations for Lagrange's multipliers in the opposite direction of time. The presence of a discontinuity in an initial value of coordinates and dependence the initial and final instants and magnitude of delay time from parameters are taken into account also.

Keywords: variational method; sensitivity functional; sensitivity coefficient; integro-differential equation; conjugate equation; delay time.

The sensitivity functional (SF) connect the first variation of quality functional with variations of variable and constant parameters and the sensitivity coefficients (SC) are components of vector gradient from quality functional according to constant parameters. Sensitivity coefficients are components of SF.

The problem of calculation of SF and SC of dynamic systems is principal in the syntheses and analysis of control laws, of identification and optimization algorisms, in the stability criterions [1–27]. The first-order sensitivity characteristics are mostly used. Later on we shall examine only SC and SF of the first-order. The most difficult are the distributed objects which are described by the dynamic (differential, integral, integro-differential, difference) equations with delays and by the dynamic equations in partial derivatives [2, 10, 11, 13, 17, 18, 20–23, 27].

Consider a vector output $x(t)$ of dynamic object model under continuous time $t \in [t_0, t^1]$, implicitly depending on parameters vectors $\tilde{\alpha}(t)$, $\bar{\alpha}$ and functional I constructed on $x(t)$ under $t \in [t_0, t^1]$. The first variation δI of functional I and variations $\delta\tilde{\alpha}(t)$ are connected with each other with the help of a single-

line functional – SF with respect to variable parameters $\tilde{\alpha}(t)$: $\delta_{\tilde{\alpha}(t)} I = \int_{t_0}^{t^1} V(t) \delta\tilde{\alpha}(t) dt$. SC with respect to con-

stant parameters $\bar{\alpha}$ are called a gradient of I on $\bar{\alpha}$: $(dI/d\bar{\alpha})^T \equiv \nabla_{\bar{\alpha}} I$. SC are a coefficients of single-line relationship between the first variation of functional δI and the variations $\delta\bar{\alpha}$ of constant parameters $\bar{\alpha}$:

$$\delta_{\bar{\alpha}} I = (\nabla_{\bar{\alpha}} I)^T \delta\bar{\alpha} = (dI/d\bar{\alpha}) \delta\bar{\alpha} \equiv \sum_{j=1}^m \frac{\partial I}{\partial \bar{\alpha}_j} \delta\bar{\alpha}_j.$$

The direct method of SC calculation (by means of the differentiation of quality functional with respect to constant parameters) inevitably requires a solution of cumbersome sensitivity equations to sensitivity functions $W(t)$. $W(t)$ is the matrix of single-line relationship of the first variation of dynamic model output

with parameter variations: $\delta x(t) = W(t) \delta\bar{\alpha}$. For instance, for functional $I = \int_{t_0}^{t^1} f_0(x(t), \bar{\alpha}, t) dt$ we have following

SC vector (row vector): $dI/d\bar{\alpha} = \int_{t_0}^{t^1} [(\partial f_0/\partial x)W(t) + \partial f_0/\partial \bar{\alpha}]dt$. For obtaining the matrix $W(t)$ it is necessary to decide a bulky system equations – sensitivity equations. The j -th column of matrix $W(t)$ is made of the sensitivity functions $dx(t)/d\bar{\alpha}_j$ with respect to component $\bar{\alpha}_j$ of vector $\bar{\alpha}$. They satisfy a vector equation (if y is a vector) resulting from dynamic model (for x) by derivation on a parameter $\bar{\alpha}_j$.

To variable parameters such a method is inapplicable because the sensitivity functions exist with respect to constant parameters.

For relatively simply classes of dynamic systems it is shown that in the SC calculation it is possible to get rid of deciding the bulky sensitivity equations due to the passage of deciding the conjugate equations – conjugate with respect to dynamic equations of object. Method of receipt of conjugate equations (it was offered in 1962) is cumbersome, because it is based on the analysis of sensitivity equations, and it does not get its developments.

Variational method [7], ascending to Lagrange's, Hamilton's, Euler's memoirs, makes possible to simplify the process of determination of conjugate equations and formulas of account of SC and SF. On the basis of this method it is an extension of quality functional by means of inclusion into it an object dynamic equations by means of Lagrange's multipliers and obtaining the first variation of extended functional on phase coordinates of object and on interesting parameters. Dynamic equations for Lagrange's multipliers are obtained due to set equal to a zero (in the first variation of extended functional) the functions before the first variations of phase coordinates. Given simplification first variation of extended functional brings at presence in the right part only parameter variations, i.e. it is got the SF. If all parameters are constant that the parameters variations are carried out from corresponding integrals and at the final result in obtained functional variation the coefficients before parameters variations are the required SC. Given method was used in [21–23, 25–27] for calculation of sensitivity coefficients and sensitivity functionals in the Bolts's problem for multivariate dynamic systems described by the differential, integral, integro-differential ordinary equations and equations with delay time under various initial conditions.

In [21] the sensitivity coefficients for many-dimensional dynamic systems described by the continuous and discontinuous differential equations with delay time are calculated.

In [22. 23] for dynamic systems described by ordinary continuous Volterra's of the second-kind integral equations [22] with delay time and integro-differential equations with delay time [23] the SC are received. Calculation possibility also SF for variable parameters is noted.

In [25] the SF and SC for multivariate dynamic systems described by generalized ordinary integral equations are calculated.

In [26] the same problem for multivariate dynamic systems described by generalized ordinary integro-differential equations is solved.

In [27] the SF and SC for multivariate dynamic systems described by generalized integral equations with delay time are calculated.

In this paper the variational method of account of SF is developed to more general continuous many-dimensional non-linear dynamic systems circumscribed by the vectorial non-linear continuous Volterra's integro-differential equations of the second genus with delay time, with variable $\tilde{\alpha}(t)$ and constant $\bar{\alpha}$ parameters and with reviewing of generalised quality functional (the Bolts problem) and registration of dependencies: 1) disturbing actions of a object model from initial instant; 2) of initial t_0 and final t^1 instants and of dead time from constant parameters $\bar{\alpha}$.

1. Problem statement

We suppose that the dynamic object is described by system of non-linear continuous Volterra's of the second-kind integro-differential equations (I-DE) with delay time τ

$$\begin{aligned} \dot{x}(t) &= f(x(t), x(t-\tau), y(t), y(t-\tau), \tilde{\alpha}(t), \bar{\alpha}, t), \quad t_0 < t \leq t^1, \quad 0 < \tau, \quad (1) \\ y(t) &= r(\tilde{\alpha}(t), \bar{\alpha}, t_0, t) + \int_{t_0}^t K(t, x(s), x(s-\tau), y(s), y(s-\tau), \tilde{\alpha}(s), \bar{\alpha}, s) ds, \quad t_0 \leq t \leq t^1, \\ t_0 &= t_0(\bar{\alpha}), \quad t^1 = t^1(\bar{\alpha}), \quad \tau = \tau(\bar{\alpha}), \quad x(t) = \psi_x(\tilde{\alpha}(t), \bar{\alpha}, t), \quad t \in [t_0 - \tau, t_0), \quad x(t_0) = x_0(\bar{\alpha}, t_0), \\ & \quad y(t) = \psi_y(\tilde{\alpha}(t), \bar{\alpha}, t), \quad t \in [t_0 - \tau, t_0). \end{aligned}$$

Here: the magnitudes of initial t_0 and final t^1 instants and also dead time τ and initial values $x(t_0)$ are known functions from constant parameters $\bar{\alpha} : t_0 = t_0(\bar{\alpha}), t^1 = t^1(\bar{\alpha}), \tau = \tau(\bar{\alpha}), x(t_0) = x_0(\bar{\alpha}, t_0)$; x, y – vector-columns of phase coordinates; $\tilde{\alpha}(t), \bar{\alpha}$ – vector-columns of interesting variable and constant parameters; $f(\cdot), \psi_x(\cdot), r(\cdot), K(\cdot), \psi_y(\cdot), t_0(\bar{\alpha}), t^1(\bar{\alpha}), \tau(\bar{\alpha}), x_0(\cdot)$ – known continuously differentiated limited vector-functions. The phase coordinates x, y in an index point t_0 makes a discontinuity if:

$$\begin{aligned} x^+(t_0) &\equiv x(t_0 + 0) \equiv x_0(\alpha, t_0) \neq x(t_0 - 0) \equiv x^-(t_0) \equiv \psi_x(\tilde{\alpha}(t_0), \bar{\alpha}, t_0), \\ y^+(t_0) &\equiv y(t_0 + 0) \equiv r(\tilde{\alpha}(t_0), \bar{\alpha}, t_0, t_0) \neq y(t_0 - 0) \equiv y^-(t_0) \equiv \psi_y(\tilde{\alpha}(t_0), \bar{\alpha}, t_0). \end{aligned}$$

But at the expense of an integration in a model (1) phase coordinates become continuous in instants $t_0 + n\tau, n = 1, 2, \dots$. Here is designated: $x^+(t_0) \equiv x(t_0 + 0)$ – value of a phase coordinate to the right of a point t_0 and accordingly $x^-(t_0) \equiv x(t_0 - 0)$ – to the left of a point t_0 .

The model of a measuring device is given as

$$\eta(t) = \eta(x(t), y(t), \tilde{\alpha}(t), \bar{\alpha}, t), \quad t \in [t_0, t^1], \quad (2)$$

where $\eta(\cdot)$ is also continuous, differentiated, limited (together with the first derivatives) vector-function. The required parameters $\tilde{\alpha}(t), \bar{\alpha}$ are inserted also in (2). Dimensionality of vectors x, y and η in general event can be different.

Let's notice that models (1), (2) are more the general in comparison with used in [17. P. 88].

On the basis of output coordinates of a measuring device η the quality functional of a dynamic system is constructed:

$$I(\alpha) = \int_{t_0}^{t^1} f_0(\eta(t), \tilde{\alpha}(t), \bar{\alpha}, t) dt + I_1(\eta(t^1), \bar{\alpha}, t^1). \quad (3)$$

Conditions for functions $f_0(\cdot), I_1(\cdot)$ are the same as for $f(\cdot), r(\cdot), K(\cdot), t_0(\cdot), t^1(\cdot), \tau(\cdot), \psi_x(\cdot), x_0(\cdot), \psi_y(\cdot)$, etc.

With use of a functional (3) in the optimization problem (in the theory of optimal control) known as the Bolts's problem. From it as the individual variants follows: Lagrange's problem (when there is only integrated component) and Mayer's problem (when there is only second component – function from phase coordinates at a finishing point).

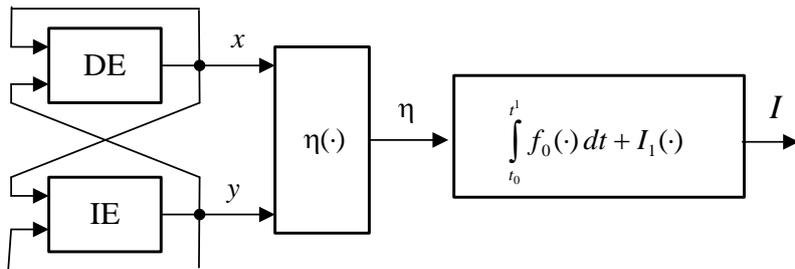


Fig. 1. The scheme of interaction between variables in (1)–(3)

With the purpose of simplification of appropriate deductions with preservation of a generality in all transformations (1)–(3) there are a two vectors of parameters $\tilde{\alpha}(t), \bar{\alpha}$. If in the equations (1)–(3) the parameters are different then it is possible formally to unite them in one vector $\tilde{\alpha}(t)$ or $\bar{\alpha}$, to use obtained outcomes and then, taking into account a structure of a vectors $\tilde{\alpha}(t), \bar{\alpha}$, to make appropriate simplifications.

The scheme of interaction between variables of object model, measuring device and quality functional is shown in fig. 1.

By obtaining of results the obvious designations:

$$\begin{aligned} f(t) &\equiv f(x(t), x(t-\tau), y(t), y(t-\tau), \tilde{\alpha}(t), \bar{\alpha}, t), \quad r(t) \equiv r(\tilde{\alpha}(t), \bar{\alpha}, t_0, t), \\ K(t, s) &\equiv K(t, x(s), x(s-\tau), y(s), y(s-\tau), \tilde{\alpha}(t), \bar{\alpha}, s), \quad \eta(t) \equiv \eta(x(t), y(t), \tilde{\alpha}(t), \bar{\alpha}, t), \\ f_0(t) &\equiv f_0(\eta(t), \tilde{\alpha}(t), \bar{\alpha}, t), \quad I_1(t^1) \equiv I_1(\eta(t^1), \bar{\alpha}, t^1) \end{aligned}$$

are used.

The convenience of a integro-differential model consists in its structural universality. At simplification of a model it is enough in final results to convert in a zero appropriate addends. This reception we shall apply in a next paper.

Let's pass from the I-DE to integral equations (IE).

Is shown also that the variation method without basic modifications allows to receive SF $\delta I(\alpha) = \int_{t_0-\tau}^{t^1} V(t) \delta \tilde{\alpha}(t) dt + (dI(\alpha)/d\tilde{\alpha}(t^1)) \delta \tilde{\alpha}(t^1) + (dI(\alpha)/d\bar{\alpha}) \delta \bar{\alpha}$ in relation to variable and constant parameters.

2. Passage to IE

In I-DE (1) the differential equations we write in the integral form

$$x(t) = x_0(\bar{\alpha}, t_0) + \int_{t_0}^t f(x(s), x(s-\tau), y(s), y(s-\tau), \tilde{\alpha}(s), \bar{\alpha}, s) ds, \quad t_0 \leq t \leq t^1 \quad (4)$$

and use notations

$$\begin{aligned} \tilde{y}(t) &= \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}, \quad \tilde{r}(\tilde{\alpha}(t), \bar{\alpha}, t_0, t) = \begin{pmatrix} x_0(\bar{\alpha}, t_0) \\ r(\tilde{\alpha}(t), \bar{\alpha}, t_0, t) \end{pmatrix} \equiv \begin{pmatrix} x_0(\bar{\alpha}, t_0) \\ r(t) \end{pmatrix} \equiv \tilde{r}(t), \\ \tilde{K}(t, \tilde{y}(s), \tilde{y}(s-\tau), \tilde{\alpha}(s), \bar{\alpha}, s) &= \begin{pmatrix} f(x(s), x(s-\tau), y(s), y(s-\tau), \tilde{\alpha}(s), \bar{\alpha}, s) \\ K(t, x(s), x(s-\tau), y(s), y(s-\tau), \tilde{\alpha}(s), \bar{\alpha}, s) \end{pmatrix} \equiv \begin{pmatrix} f(s) \\ K(t, s) \end{pmatrix} \equiv \tilde{K}(t, s), \\ \psi(\tilde{\alpha}(t), \bar{\alpha}, t) &= \begin{pmatrix} \psi_x(\tilde{\alpha}(t), \bar{\alpha}, t) \\ \psi_y(\tilde{\alpha}(t), \bar{\alpha}, t) \end{pmatrix}. \end{aligned} \quad (5)$$

Then we obtain IE

$$\begin{aligned} \tilde{y}(t) &= \tilde{r}(\tilde{\alpha}(t), \bar{\alpha}, t_0, t) + \int_{t_0}^t \tilde{K}(t, \tilde{y}(s), \tilde{y}(s-\tau), \tilde{\alpha}(s), \bar{\alpha}, s) ds, \quad t_0 \leq t \leq t^1, \\ \tilde{y}(t) &= \psi(\tilde{\alpha}(t), \bar{\alpha}, t), \quad t \in [t_0 - \tau, t_0]. \end{aligned} \quad (6)$$

In further also a notation

$$\eta(t) \equiv \eta(\tilde{y}(t), \tilde{\alpha}(t), \bar{\alpha}, t) \quad (7)$$

is used for a model of a measuring device.

3. SF with use of a models (6), (7) and a quality functional (3)

We use results of paper [27] for models (6), (7), (3) with variables and constant parameters $\tilde{\alpha}(t), \bar{\alpha}$ and write out in the beginning the conjugate equations for basic Lagrange's multipliers $\gamma(t), \bar{\gamma}(t)$:

$$\gamma^T(t) = \Phi(t^1) \frac{\partial \tilde{K}(t^1, t)}{\partial \tilde{y}(t)} + \frac{\partial f_0(t)}{\partial \eta(t)} \frac{\partial \eta(t)}{\partial \tilde{y}(t)} + \int_t^{t^1} \gamma^T(s) \frac{\partial \tilde{K}(s, t)}{\partial \tilde{y}(t)} ds + \quad (8)$$

$$+ 1(t^1 - \tau - t) [\Phi(t^1) \frac{\partial \tilde{K}(t^1, t + \tau)}{\partial \tilde{y}(t)} + \int_{t+\tau}^{t^1} \gamma^T(s) \frac{\partial \tilde{K}(s, t + \tau)}{\partial \tilde{y}(t)} ds], \quad t_0 \leq t \leq t^1,$$

$$\bar{\gamma}^T(t) = 1(t^1 - \tau - t) [\Phi(t^1) \frac{\partial \tilde{K}(t^1, t + \tau)}{\partial \tilde{y}(t)} + \int_{t+\tau}^{t^1} \gamma^T(s) \frac{\partial \tilde{K}(s, t + \tau)}{\partial \tilde{y}(t)} ds], \quad t_0 - \tau \leq t \leq t_0. \quad (9)$$

Here: $\gamma(t), \bar{\gamma}(t)$ are column vectors; $\Phi(t^1) \equiv \frac{\partial I_1(t^1)}{\partial \eta(t^1)} \frac{\partial \eta(t^1)}{\partial \tilde{y}(t^1)}$; $1(z)$ is single function: it is equal to zero under negative values of argument and is equal to unit under positive values z . These conjugate equations are decided in the opposite direction of time (from t^1).

From the conjugate equations (8), (9) it is possible to remove single function and to add them a customary aspect.

If $t_0 \leq t^1 - \tau \leq t^1$, i.e. length of an interval $[t_0, t^1]$ transcends magnitude of a delay time τ , then:

$$\gamma^T(t) = \Phi(t^1) \frac{\partial \tilde{K}(t^1, t)}{\partial \tilde{y}(t)} + \frac{\partial f_0(t)}{\partial \eta(t)} \frac{\partial \eta(t)}{\partial \tilde{y}(t)} + \int_t^{t^1} \gamma^T(s) \frac{\partial \tilde{K}(s, t)}{\partial \tilde{y}(t)} ds \quad \text{for } t^1 - \tau \leq t \leq t^1,$$

$$\gamma^T(t) = \Phi(t^1) \frac{\partial \tilde{K}(t^1, t)}{\partial \tilde{y}(t)} + \frac{\partial f_0(t)}{\partial \eta(t)} \frac{\partial \eta(t)}{\partial \tilde{y}(t)} + \int_t^{t^1} \gamma^T(s) \frac{\partial \tilde{K}(s, t)}{\partial \tilde{y}(t)} ds +$$

$$+ \Phi(t^1) \frac{\partial \tilde{K}(t^1, t + \tau)}{\partial \tilde{y}(t)} + \int_{t+\tau}^{t^1} \gamma^T(s) \frac{\partial \tilde{K}(s, t + \tau)}{\partial \tilde{y}(t)} ds \quad \text{for } t_0 \leq t \leq t^1 - \tau,$$

$$\bar{\gamma}^T(t) = \Phi(t^1) \frac{\partial \tilde{K}(t^1, t + \tau)}{\partial \tilde{y}(t)} + \int_{t+\tau}^{t^1} \gamma^T(s) \frac{\partial \tilde{K}(s, t + \tau)}{\partial \tilde{y}(t)} ds \quad \text{for } t_0 - \tau \leq t \leq t_0.$$

If $t^1 - \tau \leq t_0$, i.e. the magnitude of delay τ transcends length of an interval $[t_0, t^1]$, (in this case magnitude $t_0 + \tau$ exceeds t^1 - goes out for an interval of object work):

$$\gamma^T(t) = \Phi(t^1) \frac{\partial \tilde{K}(t^1, t)}{\partial \tilde{y}(t)} + \frac{\partial f_0(t)}{\partial \eta(t)} \frac{\partial \eta(t)}{\partial \tilde{y}(t)} + \int_t^{t^1} \gamma^T(s) \frac{\partial \tilde{K}(s, t)}{\partial \tilde{y}(t)} ds \quad \text{for } t_0 \leq t \leq t^1, \quad \bar{\gamma}^T(t) = 0 \quad \text{for } t^1 - \tau \leq t \leq t_0,$$

$$\bar{\gamma}^T(t) = \Phi(t^1) \frac{\partial \tilde{K}(t^1, t + \tau)}{\partial \tilde{y}(t)} + \int_{t+\tau}^{t^1} \gamma^T(s) \frac{\partial \tilde{K}(s, t + \tau)}{\partial \tilde{y}(t)} ds \quad \text{for } t_0 - \tau \leq t \leq t^1 - \tau.$$

SF for a models (6), (7), (3) has the form:

$$\delta I = \delta_{\tilde{\alpha}(t)} I + \delta_{\tilde{\alpha}(t^1)} I + \delta_{\tilde{\alpha}} I, \quad \delta_{\tilde{\alpha}(t)} I = \int_{t_0}^{t^1} \left[\frac{\partial f_0(t)}{\partial \eta(t)} \frac{\partial \eta(t)}{\partial \tilde{\alpha}(t)} + \frac{\partial f_0(t)}{\partial \tilde{\alpha}(t)} + \gamma^T(t) \frac{\partial \tilde{r}(t)}{\partial \tilde{\alpha}(t)} \right] \delta \tilde{\alpha}(t) dt + \quad (10)$$

$$+ \Phi(t^1) \frac{\partial \tilde{K}(t^1, t)}{\partial \tilde{\alpha}(t)} + \int_t^{t^1} \gamma^T(s) \frac{\partial \tilde{K}(s, t)}{\partial \tilde{\alpha}(t)} ds \delta \tilde{\alpha}(t) dt + \int_{t_0 - \tau}^{t_0} \bar{\gamma}^T(t) \frac{\partial \psi(t)}{\partial \tilde{\alpha}(t)} \delta \tilde{\alpha}(t) dt;$$

$$\delta_{\tilde{\alpha}(t^1)} I = \left[\frac{\partial I_1(t^1)}{\partial \eta(t^1)} \frac{\partial \eta(t^1)}{\partial \tilde{\alpha}(t^1)} + \Phi(t^1) \frac{\partial \tilde{r}(t^1)}{\partial \tilde{\alpha}(t^1)} \right] \delta \tilde{\alpha}(t^1);$$

$$\delta_{\tilde{\alpha}} I = \left\{ \frac{\partial I_1(t^1)}{\partial \eta(t^1)} \frac{\partial \eta(t^1)}{\partial \tilde{\alpha}} + \frac{\partial I_1(t^1)}{\partial \tilde{\alpha}} + \Phi(t^1) \left[\frac{\partial \tilde{r}(t^1)}{\partial \tilde{\alpha}} + \int_{t_0}^{t^1} \frac{\partial \tilde{K}(t^1, s)}{\partial \tilde{\alpha}} ds \right] \right\} +$$

$$\begin{aligned}
& + \int_0^1 \left[\frac{\partial f_0(t)}{\partial \eta(t)} \frac{\partial \eta(t)}{\partial \bar{\alpha}} + \frac{\partial f_0(t)}{\partial \bar{\alpha}} + \gamma^T(t) \frac{\partial \tilde{r}(t)}{\partial \bar{\alpha}} + \int_t^1 \gamma^T(s) \frac{\partial \tilde{K}(s,t)}{\partial \bar{\alpha}} ds \right] dt + \int_{t_0-\tau}^{t_0} \bar{\gamma}^T(t) \frac{\partial \Psi(t)}{\partial \bar{\alpha}} dt + \\
& + \left[\Phi(t^1) \left[\frac{\partial \tilde{r}(t^1)}{\partial t_0} - \tilde{K}(t^1, t_0) + 1(t^1 - t_0 - \tau)(\tilde{K}(t^1, t_0 + \tau - 0) - \tilde{K}(t^1, t_0 + \tau + 0)) \right] - f_0(t_0) + \right. \\
& \left. + \int_0^1 \gamma^T(t) \left(\frac{\partial \tilde{r}(t)}{\partial t_0} - \tilde{K}(t, t_0) \right) dt + 1(t^1 - t_0 - \tau) \int_{t_0+\tau}^1 \gamma^T(t) [\tilde{K}(t, t_0 + \tau - 0) - \tilde{K}(t, t_0 + \tau + 0)] dt \right] \frac{dt_0}{d\bar{\alpha}} + \\
& + \left[\Phi(t^1) \left[\frac{\partial \tilde{r}(t^1)}{\partial t^1} + \tilde{K}(t^1, t^1) + \int_0^1 \frac{\partial \tilde{K}(t^1, s)}{\partial t^1} ds \right] + \frac{\partial I_1(t^1)}{\partial \eta(t^1)} \frac{\partial \eta(t^1)}{\partial t^1} + \frac{\partial I_1(t^1)}{\partial t^1} + f_0(t^1) \right] \frac{dt^1}{d\bar{\alpha}} + \\
& + \left[\Phi(t^1) [1(t^1 - t_0 - \tau)(\tilde{K}(t^1, t_0 + \tau - 0) - \tilde{K}(t^1, t_0 + \tau + 0)) - \int_0^1 \frac{\partial \tilde{K}(t^1, s)}{\partial \tilde{y}(s - \tau)} \frac{d\tilde{y}(s - \tau)}{d(s - \tau)} ds] + \right. \\
& \quad \left. + 1(t^1 - t_0 - \tau) \int_{t_0+\tau}^1 \gamma^T(t) [\tilde{K}(t, t_0 + \tau - 0) - \tilde{K}(t, t_0 + \tau + 0)] dt - \right. \\
& \quad \left. - \int_0^1 \gamma^T(t) \int_{t_0}^t \frac{\partial \tilde{K}(t, s)}{\partial \tilde{y}(s - \tau)} \frac{d\tilde{y}(s - \tau)}{d(s - \tau)} ds dt \right] \frac{d\tau}{d\bar{\alpha}} \Bigg\} d\bar{\alpha} \equiv \frac{dI}{d\bar{\alpha}} d\bar{\alpha}.
\end{aligned}$$

It is necessary in (8)–(10) to fulfil matrix transformations (differentiation, multiplication) with the registration earlier entered notations (5), and also

$$\gamma(t) = \begin{pmatrix} \gamma_x(t) \\ \gamma_y(t) \end{pmatrix}, \quad \gamma^T(t) = (\gamma_x^T(t); \quad \gamma_y^T(t)), \quad \bar{\gamma}(t) = \begin{pmatrix} \bar{\gamma}_x(t) \\ \bar{\gamma}_y(t) \end{pmatrix}, \quad \bar{\gamma}^T(t) = (\bar{\gamma}_x^T(t); \quad \bar{\gamma}_y^T(t)), \quad (11)$$

$$\text{i.e. } \Phi(t^1) \equiv \frac{\partial I_1(t^1)}{\partial \eta(t^1)} \frac{\partial \eta(t^1)}{\partial \tilde{y}(t^1)} = \left(\frac{\partial I_1(t^1)}{\partial \eta(t^1)} \frac{\partial \eta(t^1)}{\partial x(t^1)}; \quad \frac{\partial I_1(t^1)}{\partial \eta(t^1)} \frac{\partial \eta(t^1)}{\partial y(t^1)} \right) \equiv (\Phi_x(t^1); \quad \Phi_y(t^1)),$$

$$\frac{\partial \tilde{K}(t^1, t)}{\partial \tilde{y}(t)} = \begin{pmatrix} \frac{\partial f(t)}{\partial x(t)}; & \frac{\partial f(t)}{\partial y(t)} \\ \frac{\partial K(t^1, t)}{\partial x(t)}; & \frac{\partial K(t^1, t)}{\partial y(t)} \end{pmatrix}, \quad \frac{\partial \tilde{r}(t^1)}{\partial \bar{\alpha}} = \begin{pmatrix} \frac{\partial x_0(\bar{\alpha}, t_0)}{\partial \bar{\alpha}} \\ \frac{\partial r(t^1)}{\partial \bar{\alpha}} \end{pmatrix},$$

$$\Phi(t^1) \frac{\partial \tilde{K}(t^1, t)}{\partial \tilde{y}(t)} = \left(\Phi_x(t^1) \frac{\partial f(t)}{\partial x(t)} + \Phi_y(t^1) \frac{\partial K(t^1, t)}{\partial x(t)}; \quad \Phi_x(t^1) \frac{\partial f(t)}{\partial y(t)} + \Phi_y(t^1) \frac{\partial K(t^1, t)}{\partial y(t)} \right),$$

$$\frac{\partial f_0(t)}{\partial \eta(t)} \frac{\partial \eta(t)}{\partial \tilde{y}(t)} = \left(\frac{\partial f_0(t)}{\partial \eta(t)} \frac{\partial \eta(t)}{\partial x(t)}; \quad \frac{\partial f_0(t)}{\partial \eta(t)} \frac{\partial \eta(t)}{\partial y(t)} \right),$$

$$\gamma^T(s) \frac{\partial \tilde{K}(s, t)}{\partial \tilde{y}(t)} = \left(\gamma_x^T(s) \frac{\partial f(t)}{\partial x(t)} + \gamma_y^T(s) \frac{\partial K(s, t)}{\partial x(t)}; \quad \gamma_x^T(s) \frac{\partial f(t)}{\partial y(t)} + \gamma_y^T(s) \frac{\partial K(s, t)}{\partial y(t)} \right),$$

$$\Phi(t^1) \frac{\partial \tilde{r}(t^1)}{\partial \bar{\alpha}} = \Phi_x(t^1) \frac{\partial x_0(\bar{\alpha}, t_0)}{\partial \bar{\alpha}} + \Phi_y(t^1) \frac{\partial r(t^1)}{\partial \bar{\alpha}}, \quad \gamma^T(t) \frac{\partial \tilde{r}(t)}{\partial \bar{\alpha}} = \gamma_x^T(t) \frac{\partial x_0(\bar{\alpha}, t_0)}{\partial \bar{\alpha}} + \gamma_y^T(t) \frac{\partial r(t)}{\partial \bar{\alpha}},$$

$$\Phi(t^1) \frac{\partial \tilde{K}(t^1, t)}{\partial \bar{\alpha}} = \Phi_x(t^1) \frac{\partial f(t)}{\partial \bar{\alpha}} + \Phi_y(t^1) \frac{\partial K(t^1, t)}{\partial \bar{\alpha}}, \quad \int_t^1 \gamma^T(s) \frac{\partial \tilde{K}(s, t)}{\partial \bar{\alpha}} ds = \int_t^1 [\gamma_x^T(s) \frac{\partial f(t)}{\partial \bar{\alpha}} + \gamma_y^T(s) \frac{\partial K(s, t)}{\partial \bar{\alpha}}] ds, \text{ etc.}$$

In the total we obtain the conjugate equations for Lagrange's multipliers

$$\gamma_x^T(t) = \Phi_x(t^1) \frac{\partial f(t)}{\partial x(t)} + \Phi_y(t^1) \frac{\partial K(t^1, t)}{\partial x(t)} + \frac{\partial f_0(t)}{\partial \eta(t)} \frac{\partial \eta(t)}{\partial x(t)} + \int_t^1 [\gamma_x^T(s) \frac{\partial f(t)}{\partial x(t)} + \gamma_y^T(s) \frac{\partial K(s, t)}{\partial x(t)}] ds + \quad (12)$$

$$\begin{aligned}
 & +1(t^1 - \tau - t)[\Phi_x(t^1) \frac{\partial f(t + \tau)}{\partial x(t)} + \Phi_y(t^1) \frac{\partial K(t^1, t + \tau)}{\partial x(t)} + \int_{t+\tau}^{t^1} [\gamma_x^T(s) \frac{\partial f(t + \tau)}{\partial x(t)} + \gamma_y^T(s) \frac{\partial K(s, t + \tau)}{\partial x(t)}] ds], \\
 & \gamma_y^T(t) = \Phi_x(t^1) \frac{\partial f(t)}{\partial y(t)} + \Phi_y(t^1) \frac{\partial K(t^1, t)}{\partial y(t)} + \frac{\partial f_0(t)}{\partial \eta(t)} \frac{\partial \eta(t)}{\partial y(t)} + \int_t^{t^1} [\gamma_x^T(s) \frac{\partial f(t)}{\partial y(t)} + \gamma_y^T(s) \frac{\partial K(s, t)}{\partial y(t)}] ds + \\
 & +1(t^1 - \tau - t)[\Phi_x(t^1) \frac{\partial f(t + \tau)}{\partial y(t)} + \Phi_y(t^1) \frac{\partial K(t^1, t + \tau)}{\partial y(t)} + \int_{t+\tau}^{t^1} [\gamma_x^T(s) \frac{\partial f(t + \tau)}{\partial y} + \gamma_y^T(s) \frac{\partial K(s, t + \tau)}{\partial y}] ds], t_0 \leq t \leq t^1; \\
 & \bar{\gamma}_x^T(t) = 1(t^1 - \tau - t)[\Phi_x(t^1) \frac{\partial f(t + \tau)}{\partial x(t)} + \Phi_y(t^1) \frac{\partial K(t^1, t + \tau)}{\partial x(t)} + \int_{t+\tau}^{t^1} [\gamma_x^T(s) \frac{\partial f(t + \tau)}{\partial x(t)} + \gamma_y^T(s) \frac{\partial K(s, t + \tau)}{\partial x(t)}] ds], \\
 & \bar{\gamma}_y^T(t) = 1(t^1 - \tau - t)[\Phi_x(t^1) \frac{\partial f(t + \tau)}{\partial y(t)} + \Phi_y(t^1) \frac{\partial K(t^1, t + \tau)}{\partial y(t)} + \\
 & + \int_{t+\tau}^{t^1} [\gamma_x^T(s) \frac{\partial f(t + \tau)}{\partial y(t)} + \gamma_y^T(s) \frac{\partial K(s, t + \tau)}{\partial y(t)}] ds], t_0 - \tau \leq t \leq t_0.
 \end{aligned}$$

The first variation of a functionality I in relation to variable $\tilde{\alpha}(t)$ and constant $\tilde{\alpha}(t^1)$, $\bar{\alpha}$ parameters, has three components:

$$\delta I = \delta_{\tilde{\alpha}(t)} I + \delta_{\tilde{\alpha}(t^1)} I + \delta_{\bar{\alpha}} I; \quad (13)$$

$$\begin{aligned}
 \delta_{\tilde{\alpha}(t)} I &= \int_{t_0}^{t^1} \left[\frac{\partial f_0(t)}{\partial \eta(t)} \frac{\partial \eta(t)}{\partial \tilde{\alpha}(t)} + \frac{\partial f_0(t)}{\partial \tilde{\alpha}(t)} + \gamma_y^T(t) \frac{\partial r(t)}{\partial \tilde{\alpha}(t)} + \Phi_x(t^1) \frac{\partial f(t)}{\partial \tilde{\alpha}(t)} + \Phi_y(t^1) \frac{\partial K(t^1, t)}{\partial \tilde{\alpha}(t)} + \right. \\
 & \left. + \int_t^{t^1} [\gamma_x^T(s) \frac{\partial f(t)}{\partial \tilde{\alpha}(t)} + \gamma_y^T(s) \frac{\partial K(s, t)}{\partial \tilde{\alpha}(t)}] ds \right] \delta \tilde{\alpha}(t) dt + \int_{t_0 - \tau}^{t_0} [\bar{\gamma}_x^T(t) \frac{\partial \psi_x(t)}{\partial \tilde{\alpha}(t)} + \bar{\gamma}_y^T(t) \frac{\partial \psi_y(t)}{\partial \tilde{\alpha}(t)}] \delta \tilde{\alpha}(t) dt; \\
 \delta_{\tilde{\alpha}(t^1)} I &= \left[\frac{\partial I_1(t^1)}{\partial \eta(t^1)} \frac{\partial \eta(t^1)}{\partial \tilde{\alpha}(t^1)} + \Phi_y(t^1) \frac{\partial r(t^1)}{\partial \tilde{\alpha}(t^1)} \right] \delta \tilde{\alpha}(t^1); \quad \delta_{\bar{\alpha}} I = \left\{ \frac{\partial I_1(t^1)}{\partial \eta(t^1)} \frac{\partial \eta(t^1)}{\partial \bar{\alpha}} + \frac{\partial I_1(t^1)}{\partial \bar{\alpha}} + \right. \\
 & + \Phi_x(t^1) \left[\frac{\partial x_0(\bar{\alpha}, t_0)}{\partial \bar{\alpha}} + \int_{t_0}^{t^1} \frac{\partial f(s)}{\partial \bar{\alpha}} ds \right] + \Phi_y(t^1) \left[\frac{\partial r(t^1)}{\partial \bar{\alpha}} + \int_{t_0}^{t^1} \frac{\partial K(t^1, s)}{\partial \bar{\alpha}} ds \right] + \\
 & + \int_{t_0}^{t^1} \left[\frac{\partial f_0(t)}{\partial \eta(t)} \frac{\partial \eta(t)}{\partial \bar{\alpha}} + \frac{\partial f_0(t)}{\partial \bar{\alpha}} + \gamma_x^T(t) \frac{\partial x_0(\bar{\alpha}, t_0)}{\partial \bar{\alpha}} + \gamma_y^T(t) \frac{\partial r(t)}{\partial \bar{\alpha}} + \right. \\
 & \left. + \int_t^{t^1} [\gamma_x^T(s) \frac{\partial f(t)}{\partial \bar{\alpha}} + \gamma_y^T(s) \frac{\partial K(s, t)}{\partial \bar{\alpha}}] ds \right] dt + \int_{t_0 - \tau}^{t_0} [\bar{\gamma}_x^T(t) \frac{\partial \psi_x(t)}{\partial \bar{\alpha}} + \bar{\gamma}_y^T(t) \frac{\partial \psi_y(t)}{\partial \bar{\alpha}}] dt + \\
 & + \left[\Phi_x(t^1) \left[\frac{\partial x_0(\bar{\alpha}, t_0)}{\partial t_0} - f(t_0) + 1(t^1 - t_0 - \tau)(f(t_0 + \tau - 0) - f(t_0 + \tau + 0)) \right] + \right. \\
 & + \Phi_y(t^1) \left[\frac{\partial r(t^1)}{\partial t_0} - K(t^1, t_0) + 1(t^1 - t_0 - \tau)(K(t^1, t_0 + \tau - 0) - K(t^1, t_0 + \tau + 0)) \right] - f_0(t_0) + \\
 & + \int_{t_0}^{t^1} \gamma_x^T(t) dt \left(\frac{\partial x_0(\bar{\alpha}, t_0)}{\partial t_0} - f(t_0) \right) + \int_{t_0}^{t^1} \gamma_y^T(t) \left(\frac{\partial r(t)}{\partial t_0} - K(t, t_0) \right) dt + \\
 & \left. + 1(t^1 - t_0 - \tau) \int_{t_0 + \tau}^{t^1} \gamma_x^T(t) dt [f(t_0 + \tau - 0) - f(t_0 + \tau + 0)] + \right.
 \end{aligned}$$

$$\begin{aligned}
 & + 1(t^1 - t_0 - \tau) \int_{t_0+\tau}^{t^1} \gamma_y^T(t) [K(t, t_0 + \tau - 0) - K(t, t_0 + \tau + 0)] dt \left. \frac{dt_0}{d\bar{\alpha}} \right. + \\
 & + \left[\Phi_x(t^1) f(t^1) + \Phi_y(t^1) \left[\frac{\partial r(t^1)}{\partial t^1} + K(t^1, t^1) + \int_{t_0}^{t^1} \frac{\partial K(t^1, s)}{\partial t^1} ds \right] + \frac{\partial I_1(t^1)}{\partial \eta(t^1)} \frac{\partial \eta(t^1)}{\partial t^1} + \frac{\partial I_1(t^1)}{\partial t^1} + f_0(t^1) \right] \frac{dt^1}{d\bar{\alpha}} + \\
 & + \left[\Phi_x(t^1) [1(t^1 - t_0 - \tau)(f(t_0 + \tau - 0) - f(t_0 + \tau + 0))] - \int_{t_0}^{t^1} \left(\frac{\partial f(s)}{\partial x(s-\tau)} \frac{dx(s-\tau)}{d(s-\tau)} + \frac{\partial f(s)}{\partial y(s-\tau)} \frac{dy(s-\tau)}{d(s-\tau)} \right) ds \right] + \\
 & + \Phi_y(t^1) [1(t^1 - t_0 - \tau)(K(t^1, t_0 + \tau - 0) - K(t^1, t_0 + \tau + 0))] - \\
 & - \int_{t_0}^{t^1} \left(\frac{\partial K(t^1, s)}{\partial x(s-\tau)} \frac{dx(s-\tau)}{d(s-\tau)} + \frac{\partial K(t^1, s)}{\partial y(s-\tau)} \frac{dy(s-\tau)}{d(s-\tau)} \right) ds + 1(t^1 - t_0 - \tau) \int_{t_0+\tau}^{t^1} \gamma_x^T(t) dt [f(t_0 + \tau - 0) - f(t_0 + \tau + 0)] + \\
 & + 1(t^1 - t_0 - \tau) \int_{t_0+\tau}^{t^1} \gamma_y^T(t) [K(t, t_0 + \tau - 0) - K(t, t_0 + \tau + 0)] dt - \int_{t_0}^{t^1} \gamma_x^T(t) \int_{t_0}^t \left(\frac{\partial f(s)}{\partial x(s-\tau)} \frac{dx(s-\tau)}{d(s-\tau)} + \frac{\partial f(s)}{\partial y(s-\tau)} \frac{dy(s-\tau)}{d(s-\tau)} \right) ds dt - \\
 & - \int_{t_0}^{t^1} \gamma_y^T(t) \int_{t_0}^t \left(\frac{\partial K(t, s)}{\partial x(s-\tau)} \frac{dx(s-\tau)}{d(s-\tau)} + \frac{\partial K(t, s)}{\partial y(s-\tau)} \frac{dy(s-\tau)}{d(s-\tau)} \right) ds dt \left. \frac{d\tau}{d\bar{\alpha}} \right\} d\bar{\alpha}.
 \end{aligned}$$

It is expedient to add the conjugate equations for Lagrange's multipliers (12) too form of the integro-differential equations.

We enter new variable $\Phi_x(t^1) + \int_t^{t^1} \gamma_x^T(s) ds = \lambda_x^T(t)$, either $\gamma_x^T(t) = -\dot{\lambda}_x^T(t)$, $\lambda_x^T(t^1) = \Phi_x(t^1)$, and change an order of integrating in double integral inside of triangular area (see paper [25]) (i.e. $\int_{t_0}^{t^1} \int_{t_0}^t A(t, s) ds dt = \int_{t_0}^{t^1} \int_t^{t^1} A(s, t) ds dt$). Then conjugate equations (12) are noted as (4) and SF (see (13)) are calculated under the formula (5).

4. Basic result

Conjugate equations have the form

$$\begin{aligned}
 -\dot{\lambda}_x^T(t) &= \lambda_x^T(t) \frac{\partial f(t)}{\partial x(t)} + \Phi_y(t^1) \frac{\partial K(t^1, t)}{\partial x(t)} + \frac{\partial f_0(t)}{\partial \eta(t)} \frac{\partial \eta(t)}{\partial x(t)} + \int_t^{t^1} \gamma_y^T(s) \frac{\partial K(s, t)}{\partial x(t)} ds + \tag{14} \\
 & + 1(t^1 - \tau - t) \left[\lambda_x^T(t + \tau) \frac{\partial f(t + \tau)}{\partial x(t)} + \Phi_y(t^1) \frac{\partial K(t^1, t + \tau)}{\partial x(t)} + \int_{t+\tau}^{t^1} \gamma_y^T(s) \frac{\partial K(s, t + \tau)}{\partial x(t)} ds \right], t \in [t_0, t^1], \\
 \lambda_x^T(t^1) &= \Phi_x(t^1), \quad \gamma_y^T(t) = \lambda_x^T(t) \frac{\partial f(t)}{\partial y(t)} + \Phi_y(t^1) \frac{\partial K(t^1, t)}{\partial y(t)} + \frac{\partial f_0(t)}{\partial \eta(t)} \frac{\partial \eta(t)}{\partial y(t)} + \int_t^{t^1} \gamma_y^T(s) \frac{\partial K(s, t)}{\partial y(t)} ds + \\
 & + 1(t^1 - \tau - t) \left[\lambda_x^T(t + \tau) \frac{\partial f(t + \tau)}{\partial y(t)} + \Phi_y(t^1) \frac{\partial K(t^1, t + \tau)}{\partial y(t)} + \int_{t+\tau}^{t^1} \gamma_y^T(s) \frac{\partial K(s, t + \tau)}{\partial y(t)} ds \right], t_0 \leq t \leq t^1; \\
 \bar{\gamma}_x^T(t) &= 1(t^1 - \tau - t) \left[\lambda_x^T(t + \tau) \frac{\partial f(t + \tau)}{\partial x(t)} + \Phi_y(t^1) \frac{\partial K(t^1, t + \tau)}{\partial x(t)} + \int_{t+\tau}^{t^1} \gamma_y^T(s) \frac{\partial K(s, t + \tau)}{\partial x(t)} ds \right], t_0 - \tau \leq t \leq t_0, \\
 \bar{\gamma}_y^T(t) &= 1(t^1 - \tau - t) \left[\lambda_x^T(t + \tau) \frac{\partial f(t + \tau)}{\partial y(t)} + \Phi_y(t^1) \frac{\partial K(t^1, t + \tau)}{\partial y(t)} + \int_{t+\tau}^{t^1} \gamma_y^T(s) \frac{\partial K(s, t + \tau)}{\partial y(t)} ds \right], t_0 - \tau \leq t \leq t_0.
 \end{aligned}$$

Here $\Phi_x(t^1) \equiv \frac{\partial I_1(t^1)}{\partial \eta(t^1)} \frac{\partial \eta(t^1)}{\partial x(t^1)}$, $\Phi_y(t^1) \equiv \frac{\partial I_1(t^1)}{\partial \eta(t^1)} \frac{\partial \eta(t^1)}{\partial y(t^1)}$.

SF are calculated under the formula:

$$\begin{aligned}
 \delta I = & \delta_{\tilde{\alpha}(t)} I + \delta_{\tilde{\alpha}(t^1)} I + \delta_{\bar{\alpha}} I; \delta_{\tilde{\alpha}(t)} I = \int_{t_0}^{t^1} \left[\frac{\partial f_0(t)}{\partial \eta(t)} \frac{\partial \eta(t)}{\partial \tilde{\alpha}(t)} + \frac{\partial f_0(t)}{\partial \tilde{\alpha}(t)} + \lambda_x^T(t) \frac{\partial f(t)}{\partial \tilde{\alpha}(t)} + \right. \\
 & \left. + \gamma_y^T(t) \frac{\partial r(t)}{\partial \tilde{\alpha}(t)} + \Phi_y(t^1) \frac{\partial K(t^1, t)}{\partial \tilde{\alpha}(t)} + \int_t^{t^1} \gamma_y^T(s) \frac{\partial K(s, t)}{\partial \tilde{\alpha}(t)} ds \right] \delta \tilde{\alpha}(t) dt + \int_{t_0-\tau}^{t_0} \left[\bar{\gamma}_x^T(t) \frac{\partial \Psi_x(t)}{\partial \tilde{\alpha}(t)} + \bar{\gamma}_y^T(t) \frac{\partial \Psi_y(t)}{\partial \tilde{\alpha}(t)} \right] \delta \tilde{\alpha}(t) dt; \\
 & \delta_{\tilde{\alpha}(t^1)} I = \left[\frac{\partial I_1(t^1)}{\partial \eta(t^1)} \frac{\partial \eta(t^1)}{\partial \tilde{\alpha}(t^1)} + \Phi_y(t^1) \frac{\partial r(t^1)}{\partial \tilde{\alpha}(t^1)} \right] \delta \tilde{\alpha}(t^1); \\
 & \delta_{\bar{\alpha}} I = \left\{ \frac{\partial I_1(t^1)}{\partial \eta(t^1)} \frac{\partial \eta(t^1)}{\partial \bar{\alpha}} + \frac{\partial I_1(t^1)}{\partial \bar{\alpha}} + \lambda_x^T(t_0) \frac{\partial x_0(\bar{\alpha}, t_0)}{\partial \bar{\alpha}} + \int_{t_0}^{t^1} \lambda_x^T(t) \frac{\partial f(t)}{\partial \bar{\alpha}} dt + \right. \\
 & \left. + \Phi_y(t^1) \left[\frac{\partial r(t^1)}{\partial \bar{\alpha}} + \int_{t_0}^{t^1} \frac{\partial K(t^1, s)}{\partial \bar{\alpha}} ds \right] + \int_{t_0}^{t^1} \left[\frac{\partial f_0(t)}{\partial \eta(t)} \frac{\partial \eta(t)}{\partial \bar{\alpha}} + \frac{\partial f_0(t)}{\partial \bar{\alpha}} + \gamma_y^T(t) \frac{\partial r(t)}{\partial \bar{\alpha}} + \int_t^{t^1} \gamma_y^T(s) \frac{\partial K(s, t)}{\partial \bar{\alpha}} ds \right] dt + \right. \\
 & \left. + \int_{t_0-\tau}^{t_0} \left[\bar{\gamma}_x^T(t) \frac{\partial \Psi_x(t)}{\partial \bar{\alpha}} + \bar{\gamma}_y^T(t) \frac{\partial \Psi_y(t)}{\partial \bar{\alpha}} \right] dt + \left[\lambda_x^T(t_0) \left(\frac{\partial x_0(\bar{\alpha}, t_0)}{\partial t_0} - f(t_0) \right) + \right. \\
 & \left. + 1(t^1 - t_0 - \tau) \lambda_x^T(t_0 + \tau) (f(t_0 + \tau - 0) - f(t_0 + \tau + 0)) + \right. \\
 & \left. + \Phi_y(t^1) \left[\frac{\partial r(t^1)}{\partial t_0} - K(t^1, t_0) + 1(t^1 - t_0 - \tau) (K(t^1, t_0 + \tau - 0) - K(t^1, t_0 + \tau + 0)) \right] - \right. \\
 & \left. - f_0(t_0) + \int_{t_0}^{t^1} \gamma_y^T(t) \left(\frac{\partial r(t)}{\partial t_0} - K(t, t_0) \right) dt + 1(t^1 - t_0 - \tau) \int_{t_0+\tau}^{t^1} \gamma_y^T(t) [K(t, t_0 + \tau - 0) - K(t, t_0 + \tau + 0)] dt \right] \frac{dt_0}{d\bar{\alpha}} + \\
 & \left[\Phi_x(t^1) f(t^1) + \Phi_y(t^1) \left[\frac{\partial r(t^1)}{\partial t^1} + K(t^1, t^1) + \int_{t_0}^{t^1} \frac{\partial K(t^1, s)}{\partial t^1} ds \right] + \frac{\partial I_1(t^1)}{\partial \eta(t^1)} \frac{\partial \eta(t^1)}{\partial t^1} + \frac{\partial I_1(t^1)}{\partial t^1} + f_0(t^1) \right] \frac{dt^1}{d\bar{\alpha}} + \\
 & \left[1(t^1 - t_0 - \tau) \lambda_x^T(t_0 + \tau) (f(t_0 + \tau - 0) - f(t_0 + \tau + 0)) - \int_{t_0}^{t^1} \lambda_x^T(t) \left(\frac{\partial f(t)}{\partial x(t-\tau)} \frac{dx(t-\tau)}{d(t-\tau)} + \frac{\partial f(t)}{\partial y(t-\tau)} \frac{dy(t-\tau)}{d(t-\tau)} \right) dt + \right. \\
 & \left. + \Phi_y(t^1) [1(t^1 - t_0 - \tau) (K(t^1, t_0 + \tau - 0) - K(t^1, t_0 + \tau + 0)) - \int_{t_0}^{t^1} \left(\frac{\partial K(t^1, s)}{\partial x(s-\tau)} \frac{dx(s-\tau)}{d(s-\tau)} + \frac{\partial K(t^1, s)}{\partial y(s-\tau)} \frac{dy(s-\tau)}{d(s-\tau)} \right) ds] + \right. \\
 & \left. + 1(t^1 - t_0 - \tau) \int_{t_0+\tau}^{t^1} \gamma_y^T(t) (K(t, t_0 + \tau - 0) - K(t, t_0 + \tau + 0)) dt - \right. \\
 & \left. - \int_{t_0}^{t^1} \gamma_y^T(t) \int_{t_0}^t \left(\frac{\partial K(t, s)}{\partial x(s-\tau)} \frac{dx(s-\tau)}{d(s-\tau)} + \frac{\partial K(t, s)}{\partial y(s-\tau)} \frac{dy(s-\tau)}{d(s-\tau)} \right) ds dt \right] \frac{d\tau}{d\bar{\alpha}} \Bigg\} d\bar{\alpha} \equiv \frac{dI}{d\bar{\alpha}} d\bar{\alpha}.
 \end{aligned}
 \tag{15}$$

Conclusion

In this paper the variational method of calculation SF and SC for the multivariate nonlinear dynamic systems described by general continuous vectorial Volterra's integro-differential equations of the second genus with dead time is developed.

The variational method is based on invariant expansion of initial functional for system due to inclusion in it of the dynamic equations of object model and of measuring device model with the help of Lagrange's

multipliers and on computation of the first variation expanded functional on phase coordinates of model and in required parameters. The equating with a zero of the functions facing to variations of phase coordinates gives the dynamic equations for Lagrange's multipliers. The simplified first variation represents required sensitivity functional.

Novelty and generality of results consists in a generality of dynamic object model, of the measuring device model and of quality functional. In models both variables and constant parameters are present. In a basis of calculation of sensitivity indexes the decision of the integro-differential equations of object model in a forward direction of time and obtained integro-differential equations for Lagrange's multipliers in the opposite direction of time lays.

Received in paper SF are more general in comparison with known in the scientific literature.

Results are applicable at the decision of problems of identification, adaptive optimal control and optimization of dynamic systems, i.e. they allow to create precision systems and devices.

This paper continues research in [17, 21–23, 25–27].

Integro-differential models structurally include separately differential and integrated models, and also 4 kinds of more simple integro-differential models which differ character of interaction of phase coordinates of integrated and differential parts. Examples of reception of these results will be presented in special paper.

Variational method of calculation of SC and SF allows to do a generalization on more complex dynamic system classes, described by: differential, integral and integro-differential equations with additional some dead times and different classes of discontinuous dynamic equations.

REFERENCES

1. Ostrovskiy, G.M. & Volin, Yu.M. (1967) *Methods of Optimization of Chemical Reactors*. Moscow: Khimiya.
2. Bellman, R. & Cooke, K.L. (1967) *Differential-difference equation*. Moscow: Mir.
3. Rosenvasser, E.N. & Yusupov, R.M. (1969) *Sensitivity of Automatic Control Systems*. Leningrad: Energiya.
4. Krutyko, P.D. (1969) The decision of a identification problem by a sensitivity theory method. *News of Sciences Academy of the USSR. Technical Cybernetics*. 6. pp. 146–153. (In Russian).
5. Petrov, B.N. & Krutyko, P.D. (1970) Application of the sensitivity theory in automatic control problems. *News of Sciences Academy of the USSR. Technical Cybernetics*. 2. pp. 202–212. (In Russian).
6. Gorodetskiy, V.I., Zacharin, F.M., Rosenvasser, E.N. & Yusupov, R.M. (1971) *Methods of Sensitivity Theory in Automatic Control*. Leningrad: Energiya.
7. Bryson, A.E. & Ho, Ju-Chi. (1972) *Applied Theory of Optimal Control*. Moscow: Mir.
8. Speedy, C.B., Brown, R.F. & Goodwin, G.C. (1973) *Control Theory: Identification and Optimal Control*. Moscow: Mir.
9. Gekher, K. (1973) *Theory of Sensitivity and Tolerances of Electronic Circuits*. Moscow: Sovetskoe radio.
10. Ruban, A.I. (1975) *Nonlinear Dynamic Objects Identification on the Base of Sensitivity Algorithm*. Tomsk: Tomsk State University.
11. Bedy, Yu.A. (1976) About asymptotic properties of decisions of the equations with delay time. *Differential Equations*. 12(9). pp. 1669–1682.
12. Rosenvasser, E.N. & Yusupov, R.M. (eds) (1977) *Cybernetics Problems. Sensitivity Theory and Its Application*. Vol. 23. Moscow: Svyaz'.
13. Mishkis, A.D. (1977) Some problems of the differential equations theory with deviating argument. *Successes of Mathematical Sciences*. 32(2). pp. 173–202.
14. Voronov, A.A. (1979) *Stability, Controllability, Observability*. Moscow: Nauka.
15. Rosenvasser, E.N. & Yusupov, R.M. (1981) *Sensitivity of Control Systems*. Moscow: Nauka.
16. Kostyuk, V.I. & Shirokov, L.A. (1981) *Automatic Parametrical Optimization of Regulation Systems*. Moscow: Energoizdat.
17. Ruban, A.I. (1982) *Identification and Sensitivity of Complex Systems*. Tomsk: Tomsk State University.
18. Tsikunov, A.M. (1984) *Adaptive Control of Objects with Delay Time*. Moscow: Nauka.
19. Haug, E.J., Choi, K.K. & Komkov, V. (1988) *Design Sensitivity Analysis of Structural Systems*. Moscow: Mir.
20. Afanasyev, V.N., Kolmanovskiy, V.B. & Nosov V.R. (1998) *The Mathematical Theory of Designing of Control Systems*. Moscow: Vysshaya shkola.
21. Rouban, A.I. (1999) Sensitivity coefficients for many-dimensional continuous and discontinuous dynamic systems with delay time. *AMSE Jourvajs, Series Advances A*. 36(2). pp. 17–36.
22. Rouban, A.I. (2002) Coefficients and functionals of sensitivity for dynamic systems described by integral equations with dead time. *AMSE Jourvajs, Series Advances C*. 57(3). pp. 15–34.

23. Rouban, A.I. (2006) Sensitivity Coefficients in the Bolts's Problem for Multivariate Dynamic Systems described by Integro-Differential Equations with Dead Time. *AMSE Journals, Series: Advances A*. 43(1). pp 15–34.
24. Gorodetskiy, Yu.I. (2006) *The sensitivity functions and dynamics of complex mechanical systems*. Nizhny Novgorod: Nizhny Novgorod State University.
25. Rouban, A.I. (2017) The sensitivity functionals in the Bolts's problem for multivariate dynamic systems described by ordinary integral equations. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science*. 38. pp. 30–36. DOI: 10.17223/19988605/38/5
26. Rouban, A.I. (2017) The sensitivity functionals in the Bolts's problem for multivariate dynamic systems described by ordinary integro-differential equations. *Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science*. 39. pp. 57–64. DOI: 10.17223/19988605/39/8
27. Rouban, A.I. (2019) The sensitivity functionals in the Bolts's problem for multivariate dynamic systems described by integral equations with delay time. *Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science*. 46. pp. 83–92. DOI: 10.17223/19988605/46/10

Received: December 10, 2018

Rouban A.I. (2019) THE SENSITIVITY FUNCTIONALS IN THE BOLTS'S PROBLEM FOR MULTIVARIATE DYNAMIC SYSTEMS DESCRIBED BY INTEGRO-DIFFERENTIAL EQUATIONS WITH DELAY TIME. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie vychislitel'naya tekhnika i informatika* [Tomsk State University Journal of Control and Computer Science]. 48. pp. 31–41

DOI: 10.17223/19988605/48/4

Рубан А.И. ФУНКЦИОНАЛЫ ЧУВСТВИТЕЛЬНОСТИ В ЗАДАЧЕ БОЛЬЦА ДЛЯ МНОГОМЕРНЫХ ДИНАМИЧЕСКИХ СИСТЕМ, ОПИСЫВАЕМЫХ ИНТЕГРО-ДИФФЕРЕНЦИАЛЬНЫМИ УРАВНЕНИЯМИ С ЗАПАЗДЫВАЮЩИМ АРГУМЕНТОМ. *Вестник Томского государственного университета. Управление, вычислительная техника и информатика*. 2019. № 48. С. 31–41

Вариационный метод применен для расчета функционалов чувствительности, которые связывают первую вариацию функционалов качества работы систем (функционалов Больца) с вариациями переменных и постоянных параметров, для многомерных нелинейных динамических систем, описываемых обобщенными интегро-дифференциальными уравнениями Вольтерра второго рода с запаздывающим аргументом.

Ключевые слова: вариационный метод; функционал чувствительности; интегро-дифференциальное уравнение с запаздывающим аргументом; функционал качества работы системы; задача Больца; сопряженное уравнение.

ROUBAN Anatoly Ivanovich (Doktor of Technical Sciences, Professor of Computer Science Department of Siberian Federal University, Krasnoyarsk, Russian Federation).
E-mail: ai-rouban@mail.ru

УДК 519.234:621.391
DOI: 10.17223/19988605/48/5

И.М. Рудько

ОБНАРУЖЕНИЕ ПОДВИЖНОГО ОБЪЕКТА С ИСПОЛЬЗОВАНИЕМ ПОРЯДКОВЫХ СТАТИСТИК

Рассматривается задача обнаружения подвижного объекта неподвижным наблюдателем. Проводится сравнение двух алгоритмов обнаружения при использовании ими энергетического критерия обнаружения – «классическая» задача проверки двух простых гипотез и применение фильтра на основе порядковой статистики (УПС-фильтра). На моделях показано, что применение УПС-фильтра обеспечивает существенный выигрыш в вероятности обнаружения.

Ключевые слова: порядковая статистика; проверка статистических гипотез; системы обнаружения; статистическое моделирование.

В простейшей форме операция обнаружения – это задача проверки двух простых гипотез: нулевой гипотезы H_0 , когда данные относятся только к шуму, и альтернативной гипотезы H_1 , когда данные относятся к совместному воздействию сигнала и шума. При решении задачи обнаружения приемник вычисляет отношение правдоподобия [1], которое представляет собой отношение условных плотностей вероятностей для гипотез H_1 и H_0 .

При этом, например, в пассивных гидролокационных системах [2] приемная система наблюдателя работает периодически, каждый раз накапливая и обрабатывая сигнал в течение фиксированного времени T_0 (время разового наблюдения), за которое положение обнаруживаемого объекта и его скорость практически не изменяются. По истечении этого времени принимается решение об отсутствии или наличии сигнала.

За время T_0 система обнаружения наблюдает сигнал, состоящий либо из одного шума в случае отсутствия обнаруживаемого объекта, либо из аддитивной смеси помехи и полезного сигнала, порожденного объектом. Математической моделью входного сигнала системы обнаружения служит случайный процесс с определенными статистическими свойствами. Распределение этого процесса часто предполагается нормальным с нулевым математическим ожиданием как в случае одной помехи (гипотеза H_0), так и в случае смеси помехи с сигналом (гипотеза H_1). В этом случае приходится решать задачу обнаружения сигнала на фоне помехи, обладающей близкими к сигналу статистическими свойствами. Распределения вероятностей для шума и смеси полезного сигнала с шумом различаются только дисперсией (мощностью) наблюдаемого сигнала.

Модель обнаружения в этих задачах обычно представляется как энергетический порог, установленный над средним значением помехи, т.е. задается вероятность ложной тревоги $P_{лт}$ (энергетический критерий обнаружения) [1].

Задача входного фильтра, стоящего перед блоком принятия решения, – максимально «разделить» статистики H_0 и H_1 , используя различия в статистических и частотных параметрах шума и сигнала (согласованная фильтрация). В случае, когда таких различий (кроме различных дисперсий) нет или, например, частотный спектр сигнала неизвестен или изменяется случайным образом, целесообразно использовать фильтр на основе усеченной порядковой статистики (УПС-фильтр) [3–5].

1. Энергетический критерий обнаружения

Пусть на интервале $[0, T_0]$ наблюдается сигнал

$$X = \sum_{i=1}^n S^2(i\Delta t), \quad (1)$$

где $T_0 = n\Delta t$; $\Delta t = 1/2\Delta F$; Δt – интервал дискретизации по времени; ΔF – полоса пропускания входного фильтра системы обнаружения.

В случае дискретизации времени решение о наличии или отсутствии сигнала принимается по набору полученных в результате предварительной обработки значений n независимых гауссовских случайных величин S_1, S_2, \dots, S_n , имеющих нулевое математическое ожидание и одинаковую дисперсию σ^2 . Здесь $\sigma^2 = \sigma_{\text{ш}}^2$ в случае отсутствия сигнала ($\sigma_{\text{ш}}^2$ – дисперсия помехи) и $\sigma^2 = \sigma_{\text{ш}}^2 + \sigma_c^2$ в случае присутствия сигнала (σ_c^2 – дисперсия полезной составляющей).

При этих условиях оптимальным правилом решения является сравнение статистики (1) с порогом h . При заданном пороге h вероятность ложной тревоги $P_{\text{лт}}$ и вероятность обнаружения $P_{\text{обн}}$ определяются соответственно равенствами

$$P_{\text{лт}} = \int_h^{\infty} f_0(x) dx, \quad P_{\text{обн}} = \int_h^{\infty} f_1(x) dx, \quad (2)$$

где $f_0(x)$ – плотность вероятности статистики X в случае отсутствия полезного сигнала, $f_1(x)$ – плотность вероятности статистики X в случае наличия полезного сигнала от объекта.

Статистику X можно представить в виде $X = \sigma^2 Z$, где Z – случайная величина, имеющая χ^2 -распределение с n степенями свободы, а $\sigma^2 = \sigma_{\text{ш}}^2$ в случае гипотезы H_0 и $\sigma^2 = \sigma_c^2 + \sigma_{\text{ш}}^2$ в случае альтернативы H_1 .

Плотность распределения статистики X для гауссовского процесса $S(t)$ описывается формулой

$$f_n(x) = \frac{1}{\sigma^2} k_n \left(\frac{x}{\sigma^2} \right), \quad x \geq 0, \quad (3)$$

где $k_n(\cdot)$ – плотность центрального χ^2 -распределения с n степенями свободы; σ^2 – дисперсия.

Функция распределения X

$$F_n(x) = \frac{1}{(2\sigma^2)^{n/2} \Gamma(n/2)} \int_0^x x^{n/2-1} e^{-x/2\sigma^2} dx = K_n(x/\sigma^2), \quad (4)$$

где $K_n(\cdot)$ – функция центрального χ^2 -распределения с n степенями свободы.

В рассматриваемой задаче проверки двух простых гипотез сигналы отличаются только дисперсиями:

$$\sigma_0^2 = 2n\sigma_{\text{ш}}^4 \text{ для } H_0, \quad \sigma_1^2 = 2n\sigma_{\text{ш}}^4(1 + \rho^2) \text{ для } H_1, \quad (5)$$

где $\rho = \sigma_c^2 / \sigma_{\text{ш}}^2$ – отношение сигнал/помеха в полосе ΔF .

Пусть h – порог обнаружения. Тогда вероятность ложной тревоги $P_{\text{лт}} = 1 - F_n(h / \sigma_0^2) = 1 - K_n(h)$,

а вероятность обнаружения $P_{\text{обн}} = 1 - F_n(h / \sigma_1^2) = 1 - K_n\left(\frac{h}{1 + \rho}\right)$.

Если аппроксимировать χ^2 -распределение нормальным распределением $N(\mu, \sigma^2)$, то для гипотезы H_0 : $N(\mu_0, \sigma_0^2)$, где $\mu_0 = n\sigma_{\text{ш}}^2$, $\sigma_0^2 = 2n\sigma_{\text{ш}}^4$, а для гипотезы H_1 : $N(\mu_1, \sigma_1^2)$, где $\mu_1 = n\sigma_{\text{ш}}^2(1 + \rho)$, $\sigma_1^2 = 2n\sigma_{\text{ш}}^4(1 + \rho^2)$.

Для нормальных распределений удобным способом сравнения статистик H_0 и H_1 является использование коэффициента делимости [4]

$$\gamma = (\mu_1 - \mu_0) / (\sigma_0 + \sigma_1). \quad (6)$$

Очевидно, что чем больше γ , тем при заданном пороге больше $P_{\text{обн}}$.

2. Фильтр на основе усеченной порядковой статистики

В работах [4, 5] рассматриваются УПС-фильтры, реализуемые во временной [4] или частотной [5] областях.

Алгоритм работы УПС-фильтра следующий (T_0 – время наблюдения):

– T_0 разбивается на m одинаковых интервалов, в каждом из которых согласно (1) вычисляется последовательность выборок $\mathbf{X}_j \sim \{X_1, \dots, X_i, \dots, X_m\}$.

– Накапливается c оцениваемых выборок \mathbf{X}_j .

– По накопленным выборкам строится матрица \mathbf{X}_{ij} размерностью m строк на c столбцов (c – «глубина» матрицы памяти) – $\{X_1, \dots, X_i, \dots, X_m\}_j$, где $1 \leq j \leq c$.

– В каждом столбце матрицы \mathbf{X}_{ij} строится порядковая статистика $\mathbf{X}_{(i)j}$ ($1 \leq i \leq m$, $\mathbf{X}_{(i)j}$ – упорядоченные величины статистики X_i , такие, что $X_{(1)} \leq X_{(2)} \leq \dots \leq X_{(i)} \leq \dots \leq X_{(m)}$).

– В каждой строке полученной матрицы $\mathbf{X}_{(i)j}$ определяются оценки математических ожиданий (вектор \hat{m}) $\hat{m}_i = \frac{1}{c} \sum_{j=1}^c X_{(i)j}$, где $1 \leq i \leq m$.

– Порог отсека k (первый порог) определяется из условия

$$k = \arg \min_i |h_0 - \hat{m}_i|, \quad (7)$$

где $1 \leq i \leq m$, а h_0 определяется, как будет показано ниже, по формуле (10).

– Вычисляется оценка $W_j = \sum_{i=k}^m X_{(i)j}$

Таким образом, на выходе УПС-фильтра получаем последовательность отфильтрованных оценок W_j , задача обнаружения по которым решается по «классическому» алгоритму обнаружения (задача проверки двух гипотез).

УПС-фильтр работает по принципу скользящего окна, т.е. каждый новый вектор \mathbf{X}_j с индексом $c + 1$ вытесняет из матрицы $\mathbf{X}_{(i)j}$ вектор \mathbf{X}_j с индексом 1.

В отличие от алгоритма проверки двух гипотез, для реализации предлагаемого алгоритма обнаружения необходимо предварительное накопление выборок $\{\mathbf{X}_1, \dots, \mathbf{X}_i, \dots, \mathbf{X}_m\}_j$, где $1 \leq j \leq c$, что приводит к задержке в принятии решения на время $T = jT_0$, где $1 \leq j \leq c$. Такая задержка во многих задачах не является существенной.

Следует подчеркнуть, что если в алгоритме проверки двух простых гипотез для принятия решения используется только вектор \mathbf{X} , то в рассматриваемом алгоритме – матрица $\mathbf{X}_{(i)j}$, в которой текущий вектор \mathbf{X}_j является одним из столбцов.

3. Моменты усеченной порядковой статистики

Работа УПС-фильтра основана на свойствах порядковых статистик [6].

Рассмотрим выборку, состоящую из m случайных величин X_i : $\{X_1, \dots, X_i, \dots, X_m\}$. Пусть случайная величина X_i описывается плотностью распределения (3) и функцией распределения (4). Вычислим моменты случайной величины $W = \sum_{i=k}^m X_{(i)}$, где $X_{(i)}$, $1 \leq i \leq m$, – упорядоченные величины (порядковые статистики) статистики X_i , такие что $X_{(1)} \leq X_{(2)} \leq \dots \leq X_{(i)} \leq \dots \leq X_{(m)}$. Если случайные величины X_i статистически независимы и одинаково распределены, то случайные величины $X_{(i)}$ зависимы из-за неравенств между ними. В дальнейшем будем называть статистику W усеченной порядковой статистикой (УПС), а параметр k – порогом отсека.

В работе [6] приведены в общем виде выражения для вычисления моментов порядковых статистик, используя которые в работе [4] для $X_{(i)}$ выведены формулы вычисления математического ожидания μ_j , дисперсии σ_j^2 и ковариации σ_{jk} порядковых статистик для χ^2 -распределения.

Математическое ожидание случайной величины W определяется по формуле

$$\mu_w(k) = \sum_{j=k}^m \mu_j, \quad 1 \leq k \leq m, \quad (8)$$

а дисперсия с учетом зависимости случайных величин $X_{(i)}$ [7]:

$$\sigma_w^2(k) = \sum_{l=k}^m \sigma_l^2 + 2 \sum_{k \leq j < l \leq m} \sigma_{jl}, \quad 1 \leq k \leq m, \quad (9)$$

и в силу центральной предельной теоремы при достаточно больших значениях m ее функция плотности распределения также нормализуется: $W \sim N(\mu_w, \sigma_w^2)$.

Таким образом, если на вход УПС-фильтра поступает процесс $X \sim N(\mu_x, \sigma_x^2)$, то на выходе – процесс $W \sim N(\mu_w, \sigma_w^2)$.

При решении задачи проверки двух гипотез для входного процесса X коэффициент разделимости γ_x определяется по (6). В работе [4] показано, что для выходного процесса W коэффициент разделимости γ_w будет зависеть от порога k и принимает максимальное значение при k , близком к медиане математического ожидания плотности вероятности шума (3). При этом $\gamma_w > \gamma_z$.

Аналитические исследования статистических свойств случайной величины W очень сложны, поэтому большинство последующих результатов получено на ПК с использованием символьного программирования в среде MATLAB (для малых значений m и n) и статистического моделирования [4, 5].

Как уже указывалось выше, известными параметрами являются только статистические свойства помехи, а именно математическое ожидание μ_0 и дисперсия σ_0^2 (см. уравнения (5)). Поэтому порог h_0 определяется согласно формуле (2) из следующего уравнения:

$$\alpha = \int_0^{h_0} f_{ш}(x) dx = \frac{1}{2^{n/2} \sigma_0^n \Gamma(n/2)} \int_0^{h_0} x^{n/2-1} e^{-x^2/2\sigma_0^2} dx, \quad (10)$$

где α – заданный квантиль, который должен быть близок к медиане шума, чтобы обеспечить максимальное значение коэффициента разделимости γ .

4. Применение УПС в задачах обнаружения подвижного объекта

Сравним рассмотренные выше алгоритмы («классический» и на основе УПС) на примере обработки гидроакустической информации в пассивном режиме [8]. Рассмотрим задачу обнаружения подвижного объекта (ПО) неподвижным наблюдателем (НН).

При обработке гидроакустической информации в пассивном режиме обнаружение осуществляется по результатам обработки излученного объектом и принятого наблюдателем сигнала при наличии помех. Решение о наличии или отсутствии сигнала от объекта принимается наблюдателем периодически, после предварительной обработки поступившей на интервале наблюдения (накопления) длительностью T_0 реализации гауссовских случайных величин $S_1, \dots, S_i, \dots, S_n$ с нулевым математическим ожиданием. Обозначим символом $\sigma_{ш}^2$ дисперсию помех на входе наблюдателя, символом $\sigma_c^2 = \sigma_c^2(v, D)$ – дисперсию сигнала, излученного движущимся объектом и поступившего на вход наблюдателя, зависящую от текущей скорости движения объекта v и текущего расстояния D между ним и наблюдателем. При отсутствии сигнала от объекта случайные величины S_i имеют дисперсию $\sigma_{ш}^2$, при наличии сигнала от объекта – дисперсию $\sigma_c^2 + \sigma_{ш}^2$.

Таким образом, задача обработки гидроакустической информации в пассивном режиме полностью описывается моделью, использующей энергетический критерий обнаружения, а система обнаружения описывается рассмотренной выше моделью задачи проверки двух простых гипотез [1].

Как показано в [9], вероятность обнаружения ПО определяется по формуле

$$P_{\text{обн}}(v, D) = 1 - F_N \left(\frac{h_\alpha}{\rho(v, D) + 1} \right), \quad (12)$$

где h_α квантиль уровня $(1 - \alpha)$ для χ^2 -распределения с N степенями свободы, $N = 2T_0\Delta F$, $\alpha = P_{\text{лт}}$ – вероятность ложной тревоги.

Сравним вероятности обнаружения для задачи проверки двух простых гипотез и для задачи обнаружения с использованием УПС-фильтра.

Для заданной вероятности ложной тревоги $P_{\text{лт}}$ вероятности обнаружения в зависимости от дистанции D для «классической» задачи определяются по формуле (12) – обозначим их как $P_{\text{обн}}^Z(v, D)$.

В случае использования в задаче обнаружения УПС-фильтра первоначально нужно сформировать матрицу $\mathbf{X}_{(ij)}$. Для этого входной сигнал длительностью T_0 разбивается на m фрагментов, причем разбиение может происходить как во временной [4], так и в частотной [5] области. В результате каждый из m фрагментов имеет плотность вероятности, описываемую χ^2 -распределением с $n = N/m$ степенями свободы. Формулы (8) и (9) позволяют рассчитать математическое ожидание μ_0 и дисперсию σ_0^2 помехи для УПС W по известным параметрам помехи Z и заданному порогу отсечения k . «Потенциальные» вероятности обнаружения $P_{\text{обн}}^W(v, D)$ на каждом интервале усреднения зависят от отношения сигнал/помеха ρ , которое используется в формуле (12). Зная ρ , по формулам (5), (8) и (9) можно рассчитать μ_1 и σ_1^2 . Здесь используется термин «потенциальные» вероятности обнаружения, так как по определению УПС W , как уже указывалось выше, необходимо вычислить оценку \hat{m}_k , для чего надо обеспечить условие $k = \text{const}$ для достаточно большого числа реализаций с $\rho = \text{const}$. В рассматриваемой задаче, как следует из формулы (12), ρ является переменным и неизвестным параметром. Поэтому под «потенциальной» вероятностью обнаружения здесь понимается вероятностью обнаружения, которая могла бы быть достигнута, если бы условие $\rho = \text{const}$ выполнялось для достаточно большого числа реализаций.

При использовании гауссовского приближения для гипотезы H_0 при заданной вероятности ложной тревоги $P_{\text{лт}}$ квантиль h_0 уровня $(1 - P_{\text{лт}})$ определяется из решения уравнения

$$P_{\text{лт}} = 1 - \Phi((h_0 - m_0)/\sigma_0), \quad (13)$$

а для гипотезы H_1 вероятность обнаружения $P_{\text{обн}}$ – из уравнения

$$P_{\text{обн}} = 1 - \Phi((h_0 - m_1)/\sigma_1), \quad (14)$$

где $\Phi(\bullet)$ – функция нормального распределения.

На каждом интервале наблюдения T_0 новый вектор \mathbf{X}_j заносится в матрицу $\mathbf{X}_{(ij)}$ «глубиной» c , по формуле (7) вычисляются порог отсечения k (первый порог) и оценка W_j , которая сравнивается с порогом h_0 (второй порог). Для заданной вероятности ложной тревоги $P_{\text{лт}}$ по формулам (13) и (14) рассчитываются вероятности обнаружения в зависимости от дистанции $D - P_{\text{обн}}^W(v, d)$.

Если для «классической» задачи проверки двух гипотез все параметры модели могут быть рассчитаны аналитически, то для задачи проверки двух гипотез с использованием УПС из-за сложности модели (при больших n и m) все параметры модели могут быть рассчитаны только путем математического моделирования.

5. Результаты моделирования

Целями моделирования были:

- сравнительный анализ (там, где это возможно) статистических параметров статистик Z и W , рассчитанных по результатам теоретических расчетов и по результатам моделирования на ЭВМ;
- для алгоритма обнаружения, основанного на УПС, по результатам моделирования на ЭВМ проводилось сравнение оценок вероятности обнаружения с оценками вероятности обнаружения для «классического» алгоритма.

Математическая модель содержит генераторы случайных чисел, имеющих χ^2 -распределения с n степенями свободы и дисперсиями $\sigma_{ш}^2$ и $\sigma_c^2 + \sigma_{ш}^2$ для гипотез H_0 и H_1 соответственно. Согласно приведенным выше формулам генерируются случайные величины Z и W . Для гипотез H_0 и H_1 набираются статистики для случайных величин Z и W , по которым строятся оценки $\hat{\mu}_{Z0}$, $\hat{\sigma}_{Z0}^2$ и $\hat{\mu}_{W0}$, $\hat{\sigma}_{W0}^2$ для гипотезы H_0 и $\hat{\mu}_{Z1}$, $\hat{\sigma}_{Z1}^2$ и $\hat{\mu}_{W1}$, $\hat{\sigma}_{W1}^2$ для гипотезы H_1 . По этим оценкам для заданной вероятности ложной тревоги $P_{лт}$ по формулам (13) и (14) определяется вероятность обнаружения $P_{обн}$.

На рис. 1 приведены результаты математического моделирования случайных процессов Z и W , а именно: теоретические значения плотностей $f_Z(\sigma_n)$, $f_Z(\sigma_{n+s})$ и $f_W(\sigma_n)$, $f_W(\sigma_{n+s})$ и их нормированные гистограммы, построенные по математической модели.

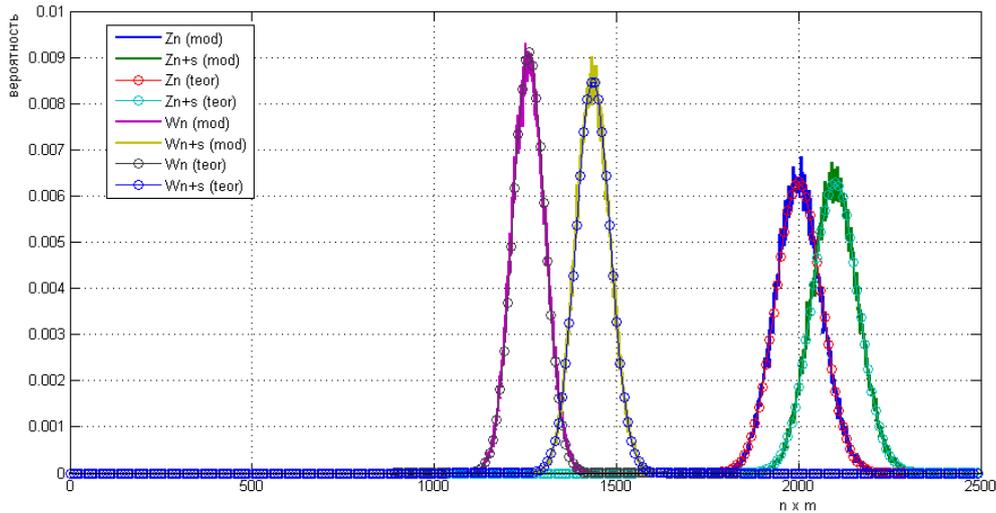


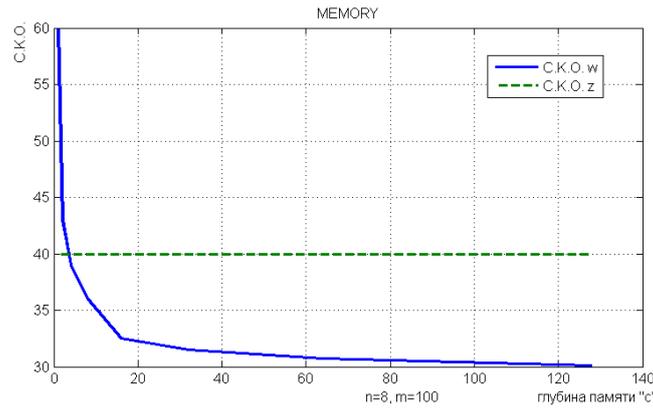
Рис. 1. Плотности вероятности процессов Z и W и их гистограммы для гипотез H_0 и H_1
 Fig. 1. Probability densities of Z and W processes and their histograms for hypotheses H_0 and H_1

Параметры модели имеют следующие значения: $n = 20$, $m = 100$, $\rho = 0,05$, первый порог равен медиане порядковой статистики $X_{(i)}$, $k = 19,34$. (Размеры массивов для построения гистограмм – 100 000). На рис. 1 хорошо видно, что использование УПС-фильтра привело к существенному увеличению разделимости случайных процессов на входе (Z) и выходе (W) УПС-фильтра. Коэффициенты разделимости γ , полученные при моделировании: $\gamma_Z = 0,048$ и $\gamma_W = 0,386$, что близко к теоретическим значениям. Наблюдается достаточно близкое совпадение между теоретическими и модельными результатами. Следовательно, результаты, полученные при моделировании, могут быть использованы в качестве оценок в тех случаях, когда теоретические параметры рассчитать сложно.

На рис. 1 приняты следующие обозначения:

- $Zn(teor)$ и $Zn(mod)$ – теоретическая плотность вероятности и ее гистограмма для шума на входе УПС-фильтра – $Z_{ш}$;
- $Zn + s(teor)$ и $Zn + s(mod)$ – теоретическая плотность вероятности и ее гистограмма для смеси сигнала и шума на входе УПС-фильтра – $Z_{ш+c}$;
- $Wn(teor)$ и $Wn(mod)$ – теоретическая плотность вероятности и ее гистограмма для шума на выходе УПС-фильтра – $W_{ш}$;
- $Wn + s(teor)$ и $Wn + s(mod)$ – теоретическая плотность вероятности и ее гистограмма для смеси сигнала и шума на выходе УПС-фильтра – $W_{ш+c}$.

На рис. 2 (пример взят из [4]) приведены результаты математического моделирования среднего квадратического отклонения УПС W в зависимости от «глубины» (числа столбцов) памяти c , содержащей матрицу порядковых статистик $X_{(ij)}$ для следующих значений параметров: $n = 8$, $m = 100$. При $c = 1$ $\sigma_W > \sigma_Z$, затем σ_W спадает и уже при $c > 4$ $\sigma_W < \sigma_Z$ и быстро стремится к своему «потенциальному» значению, т.е. к значению, когда «глубина» памяти велика.

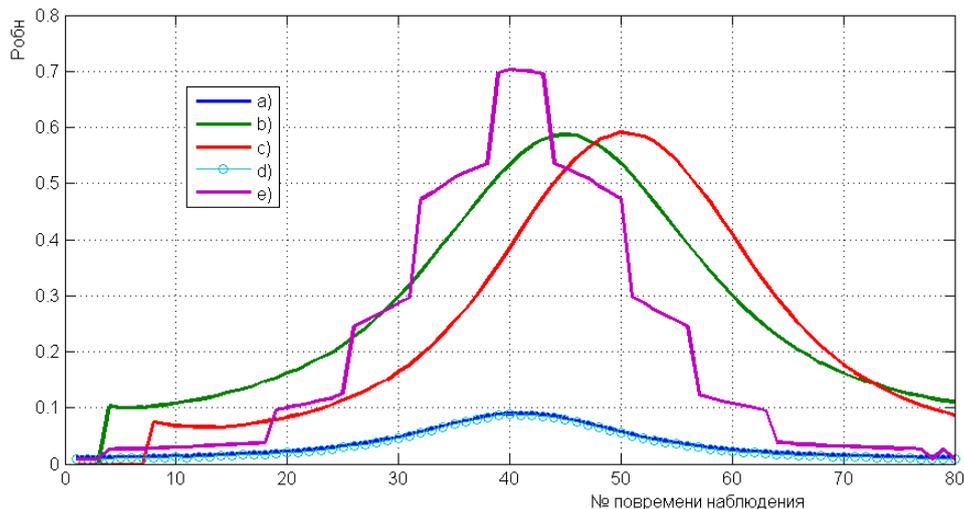
Рис. 2. Зависимость σ_w от числа столбцов памяти c Fig. 2. The dependence of σ_w on the number of columns of memory c

Далее приведены результаты моделирования системы обнаружения, реализующей «классический» (однопороговый) алгоритм, и системы обнаружения, реализующей алгоритм на основе УПС (двухпороговый).

Моделирование проводилось при следующих предположениях:

- ПО пересекает район, контролируемый НН, двигаясь прямолинейно с постоянной скоростью;
- число независимых интервалов наблюдения за время прохождения трассы $K = 80$;
- закон изменения интенсивности излучаемого ПО сигнала в зависимости от скорости – квадратичный;
- закон затухания сигнала в среде – сферический;
- суммарное число степеней свободы $N = nm = 2000$.

На рис. 3 приведены зависимости вероятности обнаружения $P_{\text{обн}}$ от дистанции D (т.е. от отношения сигнал/помеха ρ (см. уравнение (12)) для однопорогового и двухпорогового алгоритмов для «глубины» памяти $c = 4$ и $c = 8$. Параметры модели имеют следующие значения: $n = 20$, $m = 100$, $P_{\text{лт}} = 0,005$, первый порог равен медиане статистики X_i для помехи (в формуле (10) $\alpha = 0,5$). Размеры массивов для набора статистик – 100 000.

Рис. 3. Зависимости вероятностей обнаружения $P_{\text{обн}}$ от дистанции D Fig. 3. Dependence of the probability of detecting P_{det} from the distance D

На рис. 3 приняты следующие обозначения:

- a) – оценки $P_{\text{обн}}$ для однопорогового алгоритма, полученные на модели;
- b) – оценки $P_{\text{обн}}$ для двухпорогового алгоритма, полученные на модели ($c = 4$);

- с) – оценки $P_{\text{обн}}$ для двухпорогового алгоритма, полученные на модели ($c = 8$);
- d) – теоретическая $P_{\text{обн}}$ для однопорогового алгоритма;
- е) – оценки «потенциальной» $P_{\text{обн}}$ для двухпорогового алгоритма, полученные на модели.

Сдвиг начала графиков «b» и «с» определяется временем заполнения матрицы $\mathbf{X}_{(ij)} - c = 4$ или 8.

Сдвиг максимума графиков «b» и «с» относительно максимума графика «d» определяется «глубиной» памяти – $c = 4$ или 8.

Значение максимума графика «с» ($c = 8$) больше максимума графика «b» ($c = 4$), но меньше максимума графика «е», что соответствует данным, приведенным на рис. 2.

График «е» «потенциальной» $P_{\text{обн}}$ имеет ступенчатый вид, так как согласно формуле (7) при изменении \hat{m}_k изменение индекса суммирования k происходит «скачком» на $k + 1$ или $k - 1$ и,

следовательно, «скачком» происходит изменение оценки $W_j = \sum_{i=k}^m X_{(ij)}$.

Заключение

Исследован алгоритм обнаружения сигналов на фоне шума, основанный на свойствах усеченных порядковых статистик, который позволяет обеспечить большую вероятность обнаружения $P_{\text{обн}}$ при заданной вероятности ложной тревоги $P_{\text{лт}}$ по сравнению с «классическим» алгоритмом проверки двух гипотез. Выигрыш достигается за счет введения дополнительного порога, отсекающего малые значения обрабатываемого сигнала, и использования для построения оценки этого порога информации, не используемой в «классическом» алгоритме и содержащейся в предыдущих реализациях сигнала.

Приводятся результаты математического моделирования. Показано, что применение алгоритма на основе УПС (двухпорогового) в задаче обнаружения ПО неподвижным наблюдателем позволяет обеспечить существенно большую вероятность обнаружения $P_{\text{обн}}$ при заданной вероятности ложной тревоги $P_{\text{лт}}$ по сравнению с «классическим» алгоритмом проверки двух гипотез или при фиксированных вероятностях обнаружения $P_{\text{обн}}$ и ложной тревоги $P_{\text{лт}}$ обеспечить перекрытие заданного района меньшим количеством неподвижных наблюдателей.

ЛИТЕРАТУРА

1. Левин Б.Р. Теоретические основы статистической радиотехники. М. : Сов. радио, 1968. Т. 2. 504 с.
2. Urick R.J. Principles of underwater sound. McGraw-Hill Book Company, 1975. 445 p.
3. Рудько И.М. Статистические свойства суммы членов усеченного вариационного ряда // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2013. № 1 (22). С. 124–130.
4. Рудько И.М. Применение порядковых статистик в задачах обнаружения // Управление большими системами. М. : ИПУ РАН, 2012. Вып. 37. С. 63–83.
5. Рудько И.М. Применение порядковых статистик в задачах обнаружения в частотной области // Управление большими системами. М. : ИПУ РАН, 2016. Вып. 62. С. 6–29.
6. David H.A., Nagaraja H.N. Order Statistics. Wiley Series in Probability and Statistics, 2003. 458 p.
7. Левин Б.Р. Теоретические основы статистической радиотехники. М. : Сов. радио, 1969. Т. 1. 752 с.
8. Burdic W.S. Acoustic system analysis. Englewood Cliffs, NJ : Prentice Hall Inc., 1984. V. 392.
9. Абрамянц Т.Г., Галяев А.А., Маслов Е.П., Рудько И.М., Яхно В.П. Уклонение подвижного объекта в конфликтной среде от обнаружения системой разнородных наблюдателей // Проблемы управления. 2015. № 2. С. 31–37.

Поступила в редакцию 12 марта 2019 г.

Rudko I.M. (2019) DETECTION OF MOVING OBJECT USING ORDER STATISTICS. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie vychislitel'naja tehnika i informatika* [Tomsk State University Journal of Control and Computer Science]. 48. pp. 42–50

DOI: 10.17223/19988605/48/5

There are known problems of detecting a signal in the background noise, in which the statistical properties of the signal and noise are the same and their only difference are the energies (variances). In its simplest form, a detection operation is the task of testing two statistical hypotheses: null hypothesis H_0 - data refer only to noise and alternative hypothesis H_1 - data refer to the combined effect of signal and noise. The detection model in these tasks is usually represented as an energy threshold set above the average interference value. The literature also describes the algorithm for detecting signals in the background of noise, based on the statistical properties of truncated order statistics (TOS-filter).

The purpose: a comparison of the results of the detection of a moving object by a fixed observer using the “classical” problem of testing two hypotheses and for the problem of detection using a TOS -filter.

Methods: if for the “classical” problem of testing two hypotheses all the parameters of the model can be calculated analytically, then for the task of testing two hypotheses using the TOS, due to the complexity of the analytical model, all the parameters of the model can be calculated only by statistical simulation of the TOS filter, which is implemented very simply.

The author is not aware of works in which order statistics would be applied to detection tasks. According to the results of statistical simulation of the probability of detecting a moving object crossing an area controlled by a fixed observer, it is shown that using an TOS filter provides a gain compared to the “classical” task of testing two hypotheses. The gain is achieved due to the fact that if the “classical” algorithm for decision-making uses only vector X (a single observation act), then the proposed algorithm has a matrix $X_{(i)j}$, in which the current vector X_j is one of the columns. The TOS filter works on the principle of a sliding window, each new vector X_j with the index $c + 1$ (c is the number of columns of the matrix) displaces the vector X_j with the index 1 from the matrix $X_{(i)j}$. The fact that for a moving object the distance to the observer changes (signal/noise ratio changes) imposes restrictions on the time of the single act of observation T_0 and on the value of c .

The use of an TOS-based (two-threshold) algorithm in the problem of detecting a moving object by a stationary observer allows you to provide a significantly higher probability of detection P_{det} at a given probability of false alarm P_{fal} compared to the “classical” algorithm for testing two hypotheses or with fixed probabilities of detecting P_{det} and a false alarm T_0 to ensure overlap of a given area with a smaller number of fixed observers.

Keywords: order statistics; statistical hypotheses testing; detection systems; statistical simulation.

RUDKO Igor Mikhailovich (Candidate of Technical Sciences, Senior Researcher, V.A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, Moscow, Russian Federation).
E-mail: igor-rudko@mail.ru

REFERENCES

1. Levin, B.R. (1969) *Teoreticheskie osnovy statisticheskoy radiotekhniki* [Theoretical foundations of statistical radio engineering]. Vol.1. Moscow: Sovetskoe radio.
2. Urick, R.J. (1975) *Principles of Underwater Sound*. McGraw– Hill Book Company.
3. Rudko, I.M. (2013) Statistical properties of the sum of members of truncated variational series. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science*. 1(22). pp. 124–130. (In Russian).
4. Rudko, I.M. (2014) Applying Order Statistics to Detection Problems. *Automation and Remote Control*. 75(7). pp. 152–166. DOI: 10.1134/S0005117914010123
5. Rudko, I.M. (2016) Applying order statistics to detection problems in the frequency domain. *Upravlenie Bol'shimi Sistemami*. 62(3). pp. 6–29. (In Russian).
6. David, H.A. & Nagaraja, H.N. (2003) *Order Statistics*. Wiley Series in Probability and Statistics.
7. Levin, B.R. (1968) *Teoreticheskie osnovy statisticheskoy radiotekhniki* [Theoretical foundations of statistical radio engineering]. Vol.2. Moscow: Sovetskoe radio.
8. Burdic, W.S. (1984) *Acoustic System Analysis*. Vol. 392. Prentice Hall Inc. Englewood Cliffs.
9. Abramyantz, T.G., Galiaev, A.A., Maslov, E.P., Rudko, I.M. & Yakhno, V.P. (2017) Evasion of a Moving Object from Detection by a System of Heterogeneous Observers in the Threat Environment. *Automation and Remote Control*. 78(4). pp. 714–723. DOI: 10.1134/S0005117917040129

УДК 519.218.72
DOI: 10.17223/19988605/48/6

Г.Ш. Цициашвили, М.А. Осипова

ИССЛЕДОВАНИЕ ПРОЦЕССА СБОРКИ ПУАССОНОВСКИХ ПОТОКОВ

Работа выполнена при частичной финансовой поддержке гранта РФФИ № 17-07-00177.

С помощью обобщения теоремы Бурке, основанном на Хинчиновской модели пуассоновского потока, установлено, что сборка независимых пуассоновских потоков является нестационарным пуассоновским потоком. Доказано, что интенсивность сборки при устремлении времени к бесконечности стремится к меньшей из интенсивностей исходных пуассоновских потоков, и оценивается скорость этой сходимости.

Ключевые слова: сборка пуассоновских потоков; интенсивность сборки; скорость сходимости; моделирование сборки марковским процессом.

В работе строится модель сборки независимых пуассоновских потоков, под которой понимается соединение заявок с одинаковыми номерами в потоках. Процесс сборки встречается в компьютерных сетях [1, 2], системах изготовления изделий [3] и т.д. Однако исследование потока заявок, выходящих после сборки, затрудняется сложными аналитическими вычислениями, поскольку не определена удобная математическая модель этого потока.

В настоящей работе строится математическая модель процесса сборки. Эта модель основана на дискретном марковском процессе, описывающем число заявок в исходных потоках на полуинтервалах времени $[0, t)$. Скачки определенного типа у этого марковского процесса можно рассматривать как точки потока сборки. Такая модель возникает при изучении выходных потоков в моделях массового обслуживания с показательными распределениями времени обслуживания и пуассоновским входным потоком. В теореме Бурке доказано, что стационарный выходной поток в подобной системе обслуживания совпадает по распределению с входным потоком. В работе [5] дается обобщение теоремы Бурке на основе модели пуассоновского потока, предложенной А.Я. Хинчиным [6].

В настоящей работе доказывается, что сборка независимых пуассоновских потоков является нестационарным пуассоновским потоком, что существенно затрудняет исследование потока, получающегося в результате сборки. Однако с помощью вероятностных неравенств удается доказать, что интенсивность этого потока при устремлении времени t к бесконечности стремится к меньшей из интенсивностей исходных пуассоновских потоков.

Получена оценка скорости этой сходимости. В случае двух потоков с одинаковой интенсивностью степенная скорость сходимости оценивается с помощью известной асимптотики функции Инфельда. В случае $r > 2$ потоков с одинаковой интенсивностью строятся верхние степенные оценки скорости сходимости, поскольку асимптотики получающихся в результате гипергеометрических рядов использовать не удалось. Однако если у исходных потоков интенсивности разные, то интенсивность потока сборки стремится к предельной гораздо быстрее.

1. Сборка пуассоновских потоков

Пусть имеется r независимых пуассоновских потоков с интенсивностью λ . Представим эти потоки в виде $T_i = \{0 \leq t_{i,1} \leq t_{i,2} \leq \dots\}$, $i = 1, \dots, r$. Назовем поток

$$\otimes_{i=1}^r T_i = \{0 \leq \max(t_{1,1}, \dots, t_{r,1}) \leq \max(t_{1,2}, \dots, t_{r,2}) \leq \dots\}$$

сборкой потоков T_1, \dots, T_r .

Определим следующие множества индексов: $J_i = \{1, \dots, r\} \setminus \{i\}, i = 1, \dots, r$. Обозначим r -мерный вектор, состоящий из $r-1$ нулей и единицы на i -м месте, 1_i . Рассмотрим марковский процесс $(n_1(t), \dots, n_r(t)), t \geq 0$, где $n_k(t)$ – число точек потока T_k на полуинтервале $[0, t), k = 1, \dots, r$. Скачок этого процесса из состояния $(n_1, \dots, n_m), n_i < \min_{k \in J_i} n_k$, в состояние $(n_1, \dots, n_r) + 1_i$ приводит к появлению в момент времени t новой точки у потока $\otimes_{i=1}^r T_i$. Поэтому поток $\otimes_{i=1}^r T_i$ также является пуассоновским [6] с интенсивностью

$$\bar{\lambda}(t) = \lambda \sum_{i=1}^r P\left(n_i(t) < \min_{k \in J_i} n_k(t)\right). \quad (1)$$

Лемма 1. Выполняются равенства

$$\bar{\lambda}(t) = \lambda(1 - P(n_1(t) = \dots = n_r(t))). \quad (2)$$

Доказательство. Перепишем равенство (1) в виде

$$\begin{aligned} \bar{\lambda}(t) &= \lambda P\left(\bigcup_{i=1}^r (n_i(t) < \min_{k \in J_i} n_k(t))\right) = \lambda \left(1 - P\left(\bigcap_{i=1}^r (n_i(t) \geq \min_{k \in J_i} n_k(t))\right)\right) = \\ &= \lambda \left(1 - P\left(\bigcap_{i=1}^r (n_i(t) \geq \min_{k \in J} n_k(t))\right)\right) = \lambda(1 - P(n_1(t) = \dots = n_r(t))). \end{aligned}$$

Лемма доказана.

2. Предельные соотношения для интенсивности сборки потоков $\otimes_{i=1}^r T_i$

Всюду далее полагаем $a = \lambda t$.

Теорема 1. При $\lambda > 0, r = 2$ справедливо предельное соотношение

$$P(n_1(t) = n_2(t)) 2\sqrt{\pi \lambda t} \rightarrow 1, t \rightarrow \infty. \quad (3)$$

Доказательство. Действительно, выполняются равенства

$$P(n_1(t) = n_2(t)) = \sum_{k=0}^{\infty} \exp(-2a) \frac{a^{2k}}{(k!)^2} = \exp(-2a) B, B = \sum_{k=0}^{\infty} \frac{a^{2k}}{(k!)^2}.$$

Здесь $B = B(a)$ – функция Инфельда [7. Гл. 4, п. 11, формула (2.60)], удовлетворяющая следующему асимптотическому соотношению:

$$B(a) = \frac{\exp(2a)}{2\sqrt{\pi a}} \left(1 + O\left(\frac{1}{a}\right)\right). \quad (4)$$

Заменяя a на λt в формуле (4), приходим к соотношению (3). Теорема доказана.

Теорема 2. При $\lambda > 0, r > 2$ для любого $\gamma = 1/2 + \varepsilon, 0 < \varepsilon \ll 1/2$, справедливо соотношение

$$P(n_1(t) = \dots = n_r(t)) = O(t^{\gamma - \gamma/2}) = O(t^{\varepsilon - (r-1)/2}) \rightarrow 0, t \rightarrow \infty. \quad (5)$$

Доказательство. Выберем произвольное число $\gamma, 1/2 < \gamma < 1$, тогда из формулы (2) получаем равенство

$$\begin{aligned} P(n_1(t) = \dots = n_r(t)) &= \sum_{k=0}^{\infty} \left(\frac{e^{-a} a^k}{k!}\right)^r = A_1(a) + A_2(a) + A_3(a), \\ A_1(a) &= \sum_{a-a^\gamma \leq k < a+a^\gamma} \left(\frac{e^{-a} a^k}{k!}\right)^r, A_2(a) = \sum_{0 \leq k < a-a^\gamma} \left(\frac{e^{-a} a^k}{k!}\right)^r, A_3(a) = \sum_{a+a^\gamma \leq k} \left(\frac{e^{-a} a^k}{k!}\right)^r. \end{aligned}$$

Так как последовательность $\frac{a^k}{k!}$ монотонно не убывает при $0 \leq k \leq a$ и монотонно не возрастает при $a < k$, то при $a \rightarrow \infty$ из формулы Стирлинга (b – целая часть числа a) имеем

$$A_1(a) \leq 2a^\gamma \left(\frac{e^{-a} a^b}{b!} \right)^r : 2a^\gamma \left(\frac{e^{-a} a^b}{b^b e^{-b} \sqrt{2\pi b}} \right)^r \leq \frac{2a^\gamma}{(2\pi a)^{r/2}} \left(1 + \frac{1}{b} \right)^{br} : 2 \left(\frac{e}{\sqrt{2\pi}} \right)^r a^{\gamma-r/2}, A_1(a) = O(t^{\gamma-r/2}). \quad (6)$$

Построим при $a \rightarrow \infty$ оценку $A_2(a)$, полагая $c = [a - a^\gamma] : a$:

$$A_2(a) \leq c \left(\frac{e^{-a} a^c}{c!} \right)^r : a \left(\frac{e^{-a} a^c}{c^c e^{-c} \sqrt{2\pi a}} \right)^r \leq \frac{a}{(2\pi a)^{r/2}} \left(\frac{e^{-a} a^{a-a^\gamma}}{(a-a^\gamma-1)^{a-a^\gamma-1} e^{-a+a^\gamma}} \right)^r =$$

$$= \frac{a}{(2\pi a)^{r/2}} e^{rF(a)}, F(a) = -a^\gamma + (a-a^\gamma) \ln a - (a-a^\gamma-1) \ln(a-a^\gamma-1) = -\frac{a^{2\gamma-1}}{2} (1+o(1)).$$

Таким образом, из условия $1/2 < \gamma < 1$ следует предельное соотношение

$$A_2(a) = o(t^{\gamma-r/2}), a \rightarrow \infty. \quad (7)$$

Перейдем теперь при $a \rightarrow \infty$ к оценке $A_3(a)$, полагая $d = [a + a^\gamma] : a$:

$$A_3(a) \leq \sum_{d \leq k} \left(\frac{e^{-a} a^k}{k!} \right)^r \leq \left(\frac{e^{-a} a^d}{d!} \right)^r \sum_{k \geq 0} \left(\frac{a}{d} \right)^{kr} \left(\frac{e^{-a} a^d}{d!} \right)^r \frac{a^{1-\gamma}}{r}. \quad (8)$$

В свою очередь, при $a \rightarrow \infty$

$$\left(\frac{e^{-a} a^d}{d!} \right)^r \sim \left(\frac{e^{-a} a^d}{d^d e^{-d} \sqrt{2\pi a}} \right)^r \leq \left(\frac{e^{-a} a^{a+a^\gamma}}{(a+a^\gamma-1)^{a+a^\gamma-1} e^{-a+a^\gamma} \sqrt{2\pi a}} \right)^r =$$

$$= \frac{1}{(2\pi a)^{r/2}} e^{rG(a)}, G(a) = a^\gamma + (a+a^\gamma) \ln a - (a+a^\gamma-1) \ln(a+a^\gamma-1) = -\frac{a^{2\gamma-1}}{2} (1+o(1)). \quad (9)$$

Из формул (8), (9) и условия $1/2 < \gamma < 1$, приходим к предельному соотношению

$$A_3(a) = o(t^{\gamma-r/2}), a \rightarrow \infty. \quad (10)$$

Объединяя формулы (6), (7), (10), приходим к соотношению (5). Теорема доказана.

Следствие 1. Из леммы 1 и теорем 1, 2 следует, что при $r = 2$ справедливо соотношение

$$\delta_2(t) = |\bar{\lambda}(t) - \lambda| = O(t^{-1/2}) \rightarrow 0, t \rightarrow \infty,$$

а при $r > 2$ справедливо соотношение

$$\delta_r(t) = |\bar{\lambda}(t) - \lambda| = O(t^{\varepsilon-(r-1)/2}), t \rightarrow \infty, \varepsilon \ll 1/2.$$

Иными словами, с ростом числа собираемых потоков r величина $\delta_r(t)$ убывает по r достаточно быстро.

Замечание 1. Ряд $\sum_{k=0}^{\infty} (a^k / k!)^r$, исследуемый в теореме 2, является обобщенным гипергеометрическим рядом. Однако воспользоваться известными асимптотическими формулами для этого ряда не удастся [8. Ch. 16. Formula (16.11.5)]. Поэтому для него в теореме 3 приходится строить верхние оценки.

3. Сборка потоков с разными интенсивностями

Рассмотрим теперь случай, когда имеется два пуассоновских потока T_1, T_2 с интенсивностями $\lambda_1, \lambda_2, \lambda_1 < \lambda_2$. Полагая $d = \lambda_2 t, cd = \lambda_1 t, 0 < c = \frac{\lambda_1}{\lambda_2} < 1$, исследуем функцию

$$P(n_1(t) \geq n_2(t)) = \sum_{k=0}^{\infty} e^{-\lambda_2 t} \frac{(\lambda_2 t)^k}{k!} \sum_{i=k}^{\infty} e^{-\lambda_1 t} \frac{(\lambda_1 t)^i}{i!} = \sum_{k=0}^{\infty} e^{-d} \frac{d^k}{k!} \sum_{i=k}^{\infty} e^{-cd} \frac{(cd)^i}{i!} = G(d).$$

Лемма 2. При любом $\gamma = 1 - \varepsilon$, $0 < \varepsilon \ll 1/2$, справедливо предельное соотношение

$$G(d) = O\left(d^{1/2} \exp\left(-\frac{d^{2\gamma-1}}{2}(1+o(1))\right)\right), d \rightarrow \infty. \quad (11)$$

Доказательство. Представим $G(d)$ в виде суммы $G(d) = G_1(d) + G_2(d)$,

$$G_1(d) = \sum_{k>d} e^{-d} \frac{d^k}{k!} \sum_{i=k}^{\infty} e^{-cd} \frac{(cd)^i}{i!}, G_2(d) = \sum_{k \leq d} e^{-d} \frac{d^k}{k!} \sum_{i=k}^{\infty} e^{-cd} \frac{(cd)^i}{i!},$$

тогда при $cd > 1$, $0 < c < 1$, $d \rightarrow \infty$ имеем

$$\begin{aligned} G_1(d) &\leq \sum_{k>d} e^{-d} \frac{d^k}{k!} \sum_{i=[d]}^{\infty} e^{-cd} \frac{(cd)^i}{i!} \leq \sum_{i=[d]}^{\infty} \frac{e^{-cd} (cd)^i}{i!} \leq \frac{e^{-cd} (cd)^d}{[d]!} \sum_{i=[d]}^{\infty} \left(\frac{cd}{[d]}\right)^{i-[d]} = \\ &= \frac{e^{-cd} (cd)^d}{[d]!} \left(1 - \frac{cd}{[d]}\right)^{-1} \sim \frac{e^{-cd} (cd)^d}{[d]!(1-c)}. \end{aligned}$$

В свою очередь, вследствие формулы Стирлинга и неравенства $c - 1 - \ln c > 0$, $0 < c < 1$,

$$\frac{e^{-cd} (cd)^d}{[d]!} : \frac{e^{-cd} (cd)^d}{[d]^d e^{-d} \sqrt{2\pi[d]}} : \frac{e^{-cd} (cd)^d}{[d]^d e^{-d} \sqrt{2\pi d}} \leq \frac{e^{-cd} (cd)^d}{(d-1)^d e^{-d} \sqrt{2\pi d}} : \frac{\exp(-d(c-1-\ln c)+1)}{\sqrt{2\pi d}} \rightarrow 0, d \rightarrow \infty.$$

Таким образом, имеем

$$G_1(d) = O(d^{-1/2} e^{-d(c-1-\ln c)}), d \rightarrow \infty. \quad (12)$$

Оценим теперь $G_2(d) \leq \sum_{k \leq d} \frac{e^{-d} d^k}{k!}$, полагая $1/2 < \gamma < 1$:

$$G_2(d) < G_2'(d) + G_2''(d), G_2'(d) = \sum_{k < d-d^\gamma} \frac{e^{-d} d^k}{k!}, G_2''(d) \leq \sum_{d \geq k > d-d^\gamma} \frac{e^{-d} d^k}{k!} \sum_{i \geq d-d^\gamma} \frac{e^{-cd} (cd)^i}{i!}.$$

Так как при $0 \leq k \leq d$ последовательность $\frac{d^k}{k!}$ возрастает, то при $d \rightarrow \infty$ и, значит, при $l = [d - d^\gamma] \rightarrow \infty$

$$\begin{aligned} G_2'(d) &\leq l e^{-l-d^\gamma} \frac{d^l}{l!} : \frac{d e^{-d^\gamma}}{\sqrt{2\pi d}} \left(\frac{d}{l}\right)^l \leq \frac{d e^{-d^\gamma}}{\sqrt{2\pi d}} \left(\frac{d}{d-d^\gamma-1}\right)^{d-d^\gamma} = \\ &= \left(\frac{d}{2\pi}\right)^{1/2} \exp(Q(d)), Q(d) = -d^\gamma - (d-d^\gamma) \ln(1-d^{\gamma-1}-d^{-1}) = -\frac{d^{2\gamma-1}}{2}(1+o(1)), d \rightarrow \infty. \end{aligned}$$

Таким образом, получаем

$$G_2'(d) = O\left(d^{1/2} \exp\left(-\frac{d^{2\gamma-1}}{2}(1+o(1))\right)\right), d \rightarrow \infty. \quad (13)$$

Оценим теперь $G_2''(d)$ при $d \rightarrow \infty$:

$$\sum_{d \geq k > d-d^\gamma} \frac{e^{-d} d^k}{k!} \leq d^\gamma e^{-[d]} \frac{d^{[d]}}{[d]!} : d^\gamma \frac{e^{-[d]} d^{[d]}}{[d]^{[d]} e^{-[d]} \sqrt{2\pi[d]}} : \frac{d^{\gamma-1/2} e}{\sqrt{2\pi}}, d \rightarrow \infty.$$

В то же время при $d \rightarrow \infty$

$$\begin{aligned} \sum_{i \geq d-d^\gamma} e^{-cd} \frac{(cd)^i}{i!} &\leq e^{-cd} \frac{(cd)^l}{l!} \sum_{i=0}^{\infty} \left(\frac{cd}{l}\right)^i \sim \frac{e^{-cd} (cd)^l}{l^l e^{-l} \sqrt{2\pi d} (1-c)} = \\ &= \frac{1}{(1-c)\sqrt{2\pi d}} \exp(R(d)), R(d) = -cd + l(\ln c + \ln d) + l - l \ln l = -d(c-1-\ln c)(1+o(1)). \end{aligned}$$

Так как $c - 1 - \ln c > 0$, $0 < c < 1$, то справедливо следующее соотношение:

$$G_2''(d) = O(d^\gamma e^{-d(c-1-\ln c)(1+o(1))}), \quad d \rightarrow \infty, \quad (14)$$

поэтому из формул (12)–(14) следует соотношение (11). Лемма доказана.

Предположим теперь, что имеется $m > r$ независимых пуассоновских потоков с интенсивностями $\lambda_1 = \dots = \lambda_r < \lambda_{r+1} \leq \dots \leq \lambda_m$. По аналогии с формулой (2) при $Q_i(t) = P(n_i(t) < \min_{k \in J_i} n_k(t))$, $J_i = \{1, \dots, m\} \setminus \{i\}$, $i = 1, \dots, m$, имеем

$$\bar{\lambda}(t) = \sum_{i=1}^m \lambda_i Q_i(t) = \lambda_1 (1 - P(n_1(t) = \dots = n_m(t))) + \sum_{i=r+1}^m (\lambda_i - \lambda_1) Q_i(t), \quad (15)$$

$$P(n_1(t) = \dots = n_m(t)) \leq P(n_1(t) \geq n_{r+1}(t)), \quad Q_i(t) \leq P(n_1(t) \geq n_i(t)), \quad i = r+1, \dots, m. \quad (16)$$

Теорема 3. При любых $1 \leq r < m$, $\lambda_1 = \dots = \lambda_r < \lambda_{r+1} \leq \dots \leq \lambda_m$ и при любом $\gamma = 1 - \varepsilon$, $0 < \varepsilon \ll 1/2$, справедлива сходимость

$$|\bar{\lambda}(t) - \lambda_1| = O\left(t^{1/2} \exp\left(-\frac{(\lambda_{r+1} t)^{2\gamma-1}}{2} (1 + o(1))\right)\right), \quad t \rightarrow \infty.$$

Утверждение данной теоремы вытекает из формул (11), (15), (16).

Заключение

Таким образом, работе построена марковская модель сборки независимых пуассоновских потоков и доказано, что этот поток является нестационарным пуассоновским. При устремлении времени к бесконечности интенсивность нестационарного пуассоновского потока сходится к наименьшей из интенсивностей потоков, подвергающихся сборке. Построены оценки скорости этой сходимости при различном числе исходных пуассоновских потоков и при различных интенсивностях этих потоков. Для получения оценок скорости сходимости используются верхние оценки рядов, представляющих разности между нестационарными и стационарными интенсивностями пуассоновского потока сборки.

ЛИТЕРАТУРА

1. Горбунова А.В., Зарядов И.С., Самуйлов К.Е., Сопин Э.С. Обзор систем параллельной обработки заявок. Часть I // Вестник Российского университета дружбы народов. Сер. Математика. Информатика. Физика. 2017. Т. 25, № 4. С. 350–362.
2. Горбунова А.В., Зарядов И.С., Самуйлов К.Е. Обзор систем параллельной обработки заявок. Часть II // Вестник Российского университета дружбы народов. Сер. Математика. Информатика. Физика. 2018. Т. 26, № 1. С. 13–27.
3. Колесникова О.В., Лелюхин В.Е. Алгоритм определения последовательности изготовления элементов изделия «Опадающие листья» // Глобальный научный потенциал. 2015. № 2 (47). С. 54–58.
4. Burke P.J. The output of a queuing system // Operations Research. 1956. V. 4. P. 699–704.
5. Tsitsiashvili G.Sh., Osipova M.A. Modelling of output flows in queuing systems and networks // Information Technologies and Mathematical Modelling // Communications in Computer and Information Science. V. 912. P.106–116.
6. Khinchin A. Ya. Mathematical methods in the theory of queueing. London : Griffin, 1960.
7. Свешников А.Г., Боголюбов А.Н., Кравцов В.В. Лекции по математической физике : учеб. пособие. М. : Изд-во МГУ, 1993.
8. Frank W.J., Olver D.W., Lozier R.F. Boisvert and Charles W. Clark. NIST Handbook of Mathematical Functions. Cambridge : Cambridge University Press. 2010.

Поступила в редакцию 27 января 2019 г.

Tsitsiashvili G.Sh., Osipova M.A. (2019) THE STUDY OF THE ASSEMBLY OF POISSON FLOWS. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie vychislitel'naja tehnika i informatika* [Tomsk State University Journal of Control and Computer Science]. 48. pp. 51–56

DOI: 10.17223/19988605/48/6

In this paper the Markov model of assembly of independent Poisson flows is constructed and it is proved that this assembly is a non-stationary Poisson flow. When time tends to infinity, the intensity of the unsteady Poisson flow converges to the lowest of the intensities of the flows subjected to the assembly. Estimates of the rate of this convergence are constructed for different numbers of initial Poisson flows and for different intensities of these flows. To obtain estimates of the rate of convergence, upper estimates of series, representing the differences between the nonstationary and stationary intensities of the Poisson flows assembly are used.

Let there be independent Poisson flows with intensity λ . Imagine these flows in the following form: $T_i = \{0 \leq t_{i,1} \leq t_{i,2} \leq \dots\}$, $i = 1, \dots, r$. Call the flow $\otimes_{i=1}^r T_i = \{0 \leq \max(t_{1,1}, \dots, t_{r,1}) \leq \max(t_{1,2}, \dots, t_{r,2}) \leq \dots\}$ by the assembly of the flows T_1, \dots, T_r .

Define the following sets of indexes: $J_i = \{1, \dots, r\} \setminus \{i\}$, $i = 1, \dots, r$. Denote the r -dimensional vector, consisting the $r-1$ zeros and the unit on i -th place by 1_i . Consider the Markov process $(n_1(t), \dots, n_r(t))$, $t \geq 0$, where $n_k(t)$ is the number of flow points T_k on the half-interval $[0, t)$, $k = 1, \dots, r$. This process jump from state (n_1, \dots, n_m) , $n_i < \min_{k \in J_i} n_k$, to state $(n_1, \dots, n_r) + 1_i$ leads to the appearance at a moment t the new point of the flow $\otimes_{i=1}^r T_i$. So, $\otimes_{i=1}^r T_i$ is also the Poisson flow with the intensity

$$\bar{\lambda}(t) = \lambda \sum_{i=1}^r P \left(n_i(t) < \min_{k \in J_i} n_k(t) \right).$$

Lemma 1. The following equalities are performed: $\bar{\lambda}(t) = \lambda(1 - P(n_1(t) = \dots = n_r(t)))$.

Theorem 1. For $\lambda > 0$, $r = 2$, the following limit relation $P(n_1(t) = n_2(t))2\sqrt{\pi\lambda t} \rightarrow 1$, $t \rightarrow \infty$, holds.

Theorem 2. For $\lambda > 0$, $r > 2$, and for any $\gamma = 1/2 + \varepsilon$, $0 < \varepsilon \ll 1/2$, we have the limit relation

$$P(n_1(t) = \dots = n_r(t)) = O(t^{\gamma-r/2}) = O(t^{\varepsilon-(r-1)/2}) \rightarrow 0, t \rightarrow \infty.$$

Theorem 3. For any $1 \leq r < m$, $\lambda_1 = \dots = \lambda_r < \lambda_{r+1} \leq \dots \leq \lambda_m$ and for all γ , $1/2 < \gamma < 1$, the following limit relation holds:

$$|\bar{\lambda}(t) - \lambda_1| = O \left(t^{1/2} \exp \left(-\frac{(\lambda_{r+1} t)^{2\gamma-1}}{2} (1 + o(1)) \right) \right), t \rightarrow \infty.$$

Keywords: an assembly of Poisson flows; an intensity of an assembly; a rate of convergence; simulation of assembly by Markov process.

TSITSIASHVILI Gurami Shalvovich (Doctor of Physics and Mathematics, Professor, Main Researcher of Institute for Applied Mathematics, Far Eastern Branch of Russian Academy Sciences, Vladivostok, Russian Federation).

E-mail: guram@iam.dvo.ru

OSIPOVA Marina Anatolievna (Candidate of Physics and Mathematics, Associate Professor, Researcher of Institute for Applied Mathematics, Far Eastern Branch of Russian Academy Sciences, Vladivostok, Russian Federation).

E-mail: mao1975@list.ru

REFERENCES

1. Gorbunova, A.V., Zaryadov, I.S., Samuylov, K.E. & Sopin, A.S. (2017) A Survey on Queuing Systems with Parallel Serving of Customers. Part I. *Vestnik Rossiyskogo universiteta druzhby narodov. Ser. Matematika. Informatika. Fizika – Bulletin of the Peoples' Friendship University of Russia. Series: Mathematics. Information Sciences. Physics.* 25(4). pp. 350–362. (In Russian). DOI: 10.22363/2312-9735-2017-25-4-350-362
2. Gorbunova, A.V., Zaryadov, I.S. & Samuylov, K.E. (2018) A Survey on Queuing Systems with Parallel Serving of Customers. Part II. *Vestnik Rossiyskogo universiteta druzhby narodov. Ser. Matematika. Informatika. Fizika – Bulletin of the Peoples' Friendship University of Russia. Series: Mathematics. Information Sciences. Physics.* 26(1). pp. 13–27. (In Russian). DOI: 10.22363/2312-9735-2018-26-1-13-27
3. Kolesnikova, O.V. & Lelyukhin, V.E. (2015) Algorithm for determining the manufacturing steps sequence of the “Falling leaves” product. *Global'nyy nauchnyy potentsial – Global Scientific Potential.* 2(47). pp. 54–58. (In Russian).
4. Burke, P.J. (1956) The output of a queuing system. *Operations Research.* 4. pp. 699–704. DOI: 10.1287/opre.4.6.699
5. Tsitsiashvili, G.Sh. & Osipova, M.A. (2018) Modelling of output flows in queuing systems and networks. In: Dudin, A., Nazarov, A. & Moiseev, A. (eds) *Information Technologies and Mathematical Modelling. Queuing Theory and Applications.* Vol. 912. Springer, Cham. pp.106–116.
6. Khinchin, A.Ya. (1960) *Mathematical Methods in the Theory of Queuing.* London: Griffin.
7. Sveshnikov, A.G., Bogolyubov, A.N. & Kravtsov, V.V. (1993) *Lektsii po matematicheskoy fizike* [Lectures on Mathematical Physics: Studies Manual]. Moscow: Moscow State University.
8. Frank, W.J., Olver, D.W. & Lozier, R.F. (2010) *Boisvert and Charles W. Clark. NIST Handbook of Mathematical Functions.* England: Cambridge University Press.

ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ

УДК 004.931

DOI: 10.17223/19988605/48/7

А.В. Приступа, И.Л. Лапатин, А.В. Замятин**ПРИМЕНЕНИЕ ГЛОБАЛЬНЫХ ДЕСКРИПТОРОВ ФОРМЫ
ДЛЯ ИДЕНТИФИКАЦИИ 3D ОБЪЕКТОВ**

*Результаты получены в рамках выполнения государственного задания Минобрнауки России,
проект № 2.4218.2017/4.6.*

Предлагается методика последовательного применения глобальных дескрипторов, позволяющего на первом этапе произвести «грубое» отсеивание заведомо отличающихся объектов, после чего на значительно уменьшенной базе объектов применить более точные алгоритмы. Достижимый при этом эффект заключается в значительном увеличении скорости идентификации объекта (в некоторых случаях на порядок и более) без снижения точности. Для повышения точности идентификации объектов в работе также предлагается методика одновременного применения существующих глобальных дескрипторов формы.

Ключевые слова: 3D объект; облако точек; распознавание; идентификация; глобальные дескрипторы формы.

Вследствие быстрого развития технологий трехмерного сканирования объектов возникают все новые задачи и возможности использования этих технологий в различных приложениях и областях. На стыке компьютерного зрения с другими областями исследований возникают задачи, ориентированные на распознавание и классификацию объектов по трехмерным (3D) данным. Типичными примерами таких задач являются автоматическая классификация объектов и слежение за объектами [1, 2], восстановление общей поверхности по съемке ее отдельных фрагментов, распознавание в беспилотных транспортных средствах [3, 4]. Однако существующие подходы не всегда обеспечивают необходимую точность классификации при заданных требованиях к скорости работы алгоритма.

В связи с этим актуальной является задача разработки эффективных методов для решения задач классификации полученных моделей объектов, включающих построение алгоритмов идентификации 3D-объектов (сравнение с некоторой базой эталонных объектов) с заданной точностью при определенных требованиях к быстродействию. Зачастую требуется, чтобы указанные алгоритмы работали в режиме реального времени.

Существующие сегодня подходы к сравнению 3D-объектов разделяют на две принципиально разные группы.

Первая группа представляет собой алгоритмы и методы непосредственного сравнения моделей 3D-объектов. Приведем примеры некоторых из них.

Базовым алгоритмом сопоставления двух облаков точек является итеративный алгоритм ближайших точек [5, 6]. Алгоритм использует пары «ближайших» точек двух облаков точек и минимизирует расстояние между ними через поиск преобразований (параметров сдвига и поворота). Недостатками данного подхода являются чувствительность сходимости алгоритма к начальному приближению и достаточно высокая вычислительная сложность.

В работах [7–9] рассматривается подход, основанный на прямом сопоставлении однолистных поверхностей. Ограничениями данного подхода является отсутствие инвариантности относительно

масштаба и поворота, а также требование однозначности поверхности, описываемой облаком точек или полигональной моделью.

Ко второй группе подходов относится классификация и идентификация объектов с помощью дескрипторов.

Дескриптором (вектором признаков) называется набор численных параметров, описывающих характеристики объекта (или его части), например цвет, форму и т.д. Векторы признаков принимают значения в пространстве признаков. Если на таком пространстве задать меру близости, то можно сравнивать объекты друг с другом, вычисляя расстояние между соответствующими векторами признаков. Это позволяет находить более или менее похожие друг на друга 3D-объекты среди их множества.

Дескрипторы разделяют на глобальные, описывающие объект целиком, и локальные, описывающие значимые части объекта или изображения. Основной областью применения локальных дескрипторов является анализ изображений при моделировании компьютерного зрения. Примерами алгоритмов вычисления локальных дескрипторов для изображений являются SURF [10], BRIEF [11], BRISK [12].

Для идентификации 3D-объектов по данным лазерного сканирования применяются алгоритмы, основанные на вычислении глобальных дескрипторов модели объекта. Под глобальными дескрипторами подразумеваются векторы признаков, полученные при анализе всего объекта в целом. Как правило, каждая точка изображения в таких методах вносит вклад в значение дескриптора. Фактически глобальный дескриптор отображает пространство объектов в некоторое конечномерное векторное пространство.

Представления о базовых дескрипторах формы подробно описаны в работе [13]. В статье сравнивается эффективность идентификации объектов по дескрипторам A_3 , D_1 , D_2 , D_3 , D_4 , которые строятся на основании вычисления геометрических характеристик случайных точек поверхности объекта.

Для полученного набора значений рассчитывается эмпирическое распределение. Для сопоставления объектов полученные распределения сравниваются.

Среди других подходов построения глобальных дескрипторов следует отметить, например, FFT-дескриптор [14], который является комплексной функцией двух переменных на сфере. Значениями дескриптора фактически являются коэффициенты быстрого преобразования Фурье. Также к классу Фурье дескрипторов относится алгоритм, описанный в работе ученых Южного федерального университета [15].

Применение глобальных дескрипторов для идентификации 3D-объектов позволяет за меньшее по сравнению с алгоритмами прямого сопоставления моделей объектов время определять принадлежность объекта к определенному классу. При этом эффективность идентификации зависит от вида выбранного дескриптора и самих моделей объектов.

Современный дескриптор GASD [16] является достаточно эффективным, обеспечивая точность распознавания около 90% на тестовых моделях. Однако он чувствителен к точности выравнивания моделей, которое реализуется с помощью метода главных компонент. Хорошими описательными характеристиками обладает дескриптор SPIN [17], построенный на основе спиновых изображений, но эффективность применения данного дескриптора также зависит от качества выравнивания моделей (выбора вектора нормали). Таким образом, каждый самостоятельно применяемый дескриптор не всегда обеспечивает необходимую точность идентификации объектов, которая, как правило, зависит от представления исходных моделей объектов. При этом указанные алгоритмы производят полное сравнение по всей базе эталонных объектов, даже если объекты заведомо не похожи. Это приводит к избыточной вычислительной сложности.

В данной работе предлагается алгоритм последовательного применения дескрипторов, который позволяет на первом этапе произвести «грубое» отсеивание заведомо отличающихся объектов, после чего на значительно уменьшенной базе объектов применить более точные, но и более трудоемкие, алгоритмы. Для повышения точности идентификации объектов в работе предлагается методика одно-

временного применения существующих глобальных дескрипторов различных принципов построения. При сопоставлении и усреднении результатов идентификации объекта с помощью каждого из дескрипторов применяется ранговый подход.

Структура работы имеет следующий вид. В разделе 1 описываются принципы построения существующих глобальных дескрипторов G ASD, SPIN, D_1 . В разделе 2 описаны методики последовательного и одновременного применения нескольких дескрипторов. Раздел 3 содержит результаты экспериментов по идентификации объектов с помощью предлагаемых подходов.

1. Глобальные дескрипторы формы

Поскольку предполагается, что входное облако точек объектов сцены содержит только координаты точек (X, Y, Z) и не содержит никакой дополнительной информации, например направлений векторов нормалей к поверхности в каждой точке облака, то для исследования применимости глобальных дескрипторов к задаче идентификации 3D-объектов были выбраны дескрипторы, для вычисления которых достаточно знать только координаты точек (X, Y, Z) . Несмотря на то, что на основе значений координат точек можно вычислить оценку направлений векторов нормалей и использовать дескрипторы, основанные уже на этих входных данных, сам по себе процесс вычисления нормалей вычислительно затратен, может содержать ошибки при выборе ориентации векторов (+ или -); неочевидным также является выбор размера окрестности, точки которой используются для определения вектора нормали в заданной точке.

1.1. Дескриптор G ASD

Этот относительно новый метод, предложенный в 2016 г. J.P.S. Lima и V. Teichrieb [16], состоит из двух шагов. На первом шаге осуществляется выравнивание модели, т.е. переход к новой системе координат в соответствии с методом главных компонент (PCA). Для всего множества точек $P_i, i \in [1..N]$ рассчитывается центр масс точек облака $\bar{P} = \frac{1}{N} \sum_{i=1}^N P_i$, и составляется ковариационная матрица

$$C = \frac{1}{N} \sum_{i=1}^N (P_i - \bar{P})(P_i - \bar{P})^T, \quad (1)$$

по собственным значениям и векторам которой происходит выравнивание модели.

Второй шаг выполняется уже для повернутого и сдвинутого облака точек и заключается, собственно, в построении дескриптора. Для всего облака точек рассчитывается минимальный охватывающий куб, который делится на $m \times m \times m$ ячеек, для каждой из которых считается количество принадлежащих ей точек облака. Дополнительно значение каждой ячейки после подсчета количества точек делится на общее количество точек облака. Результирующий дескриптор может быть получен путем последовательной записи значений всех ячеек в виде одномерного массива из m^3 элементов.

Важным достоинством дескриптора является его способность для входного облака точек некоторого объекта A сцены не только определить ближайшую похожую модель B в базе 3D-объектов, но и определить параметры аффинного преобразования (поворот, перенос), которое необходимо применить к модели B , чтобы она наилучшим образом наложилась на облако точек объекта A .

1.2. Интегральные спиновые изображения

Спиновые изображения [17] являются одними из самых популярных на сегодняшний день локальных дескрипторов. Они строятся относительно выбранной опорной точки и описывают форму некоторого участка поверхности модели. На первом шаге для каждой точки A из окрестности опорной точки O рассчитываются ее относительные координаты по формулам

$$\alpha = \sqrt{\|A - O\|^2 - (n_0 \times (A - O))^2}; \quad \beta = n_0 \times (A - O). \quad (2)$$

Далее все точки из окрестности опорной на основе вычисленных относительных координат α и β распределяются по ячейкам таким образом, что точки с близкими относительными координатами попадают в одну и ту же ячейку. Таким образом, для заданной опорной точки с известным направлением вектора нормали спиновое изображение есть не что иное, как двумерная матрица значений (количество точек, попавших в ячейку). Достоинством спиновых изображений является их инвариантность к операциям вращения и переноса.

Идея построения локальных спиновых изображений может быть расширена для описания всей трехмерной модели и использования в качестве глобального дескриптора. Для этого необходимо выбрать опорную точку и опорный вектор, а относительные координаты рассчитать для всех точек модели, не ограничиваясь локальной окрестностью. В качестве опорной точки удобнее всего взять центр масс, рассчитанный как среднее арифметическое координат всех точек модели. Для определения опорного вектора можно воспользоваться методом главных компонент и использовать вектор, соответствующий наибольшему собственному значению ковариационной матрицы, геометрически определяющий направление наибольшей вытянутости облака точек модели. Построенное таким образом интегральное спиновое изображение также является инвариантным к операциям поворота и переноса. Однако если предполагается, что модели могут иметь разный масштаб, для корректной работы такого дескриптора необходима предварительная нормализация [18].

1.3. Функция формы D_1

Несмотря на то, что перечисленные выше дескрипторы обладают хорошими описательными характеристиками, их использование сопряжено с некоторыми проблемами. Дело в том, что вычислению каждого из этих дескрипторов предшествует этап выравнивания (определения «правильной» ориентации моделей в пространстве) в соответствии с методом главных компонент. Эта процедура выполняется для всех моделей независимо друг от друга. Однако при наложении друг на друга двух облаков точек, выровненных по методу главных компонент, может оказаться, что существует более удачный поворот, совмещающий эти модели между собой. То есть при решении задачи выравнивания / наложения двух моделей PCA дает хоть и близкий, но не всегда оптимальный результат. В связи с этим в дополнение к уже рассмотренным целесообразно рассмотреть дескриптор, для вычисления которого не требуется предварительного выравнивания моделей. В данной работе в качестве такого дескриптора выбрана функция формы D_1 [13].

Важными свойствами дескриптора являются линейная трудоемкость его вычисления и инвариантность к аффинным преобразованиям, что позволяет рассматривать его как быстрый способ «грубого распознавания», позволяющий на раннем этапе исключить из рассмотрения заведомо отличающиеся модели.

2. Методика последовательного применения дескрипторов

Поскольку входные объекты сцены представлены в виде облаков точек, а объекты базы данных, как правило, – в виде полигональных моделей, то для их корректного сравнения модели необходимо привести к единому виду. Для этого предлагается для всех объектов, имеющихся в библиотеке, наряду с полигональными моделями получить их представление в виде облака точек. В настоящей работе использован способ, предложенный в [13].

Наконец, при использовании дескрипторов, требующих поворота моделей в пространстве (GASD, SPIN и т.д.), применяется метод главных компонент, в ходе которого вместо стандартных координатных осей определяется новый базис в трехмерном пространстве (тройка ортогональных друг к другу векторов). Отметим, что после применения метода главных компонент получаемые собственные векторы необходимо нормировать – для этого значения каждого собственного вектора необходимо разделить на его длину. Координаты всех точек модели пересчитываются в соответствии с новым базисом, т.е. осуществляется поворот модели в пространстве. Отметим, что на практике для

каждой модели получается не один, а сразу четыре по-разному повернутых варианта, при этом с целью уменьшения количества сравнений целесообразно для объектов из базы использовать все четыре варианта поворота, и только один (любой из четырех) – для входящих объектов-запросов. Это обусловлено тем, что дескрипторы для объектов из базы данных рассчитываются только один раз (при появлении модели нового объекта в БД), а объекты-запросы могут появляться гораздо чаще, возможно, в режиме реального времени.

Несмотря на то, что алгоритмы построения всех рассмотренных выше дескрипторов (D_1 , GASD, SPIN) имеют линейную относительно количества точек облака трудоемкость, необходимо принимать во внимание размерность получаемых векторов. Для дескриптора D_1 она оказывается минимальной, поскольку в данном случае множество точек разбивается на «корзины» по единственному параметру (расстояние до центра масс), в то время как для SPIN и GASD разбиение осуществляется по двум и трем характеристикам соответственно. Кроме того, поскольку последние работают с моделями, предварительно повернутыми в пространстве, количество сравнений для каждой пары объектов возрастает с 1 до 4.

Таким образом, наиболее быстрое сравнение входного объекта со всеми объектами базы данных достигается при использовании дескриптора D_1 , однако его точность зачастую оказывается недостаточной по сравнению с GASD и SPIN. Поэтому дескриптор D_1 предлагается использовать для отсеивания объектов, заведомо «непохожих» на объект-запрос.

На основании вышеизложенного сформулируем предлагаемую методику быстрого поиска объекта в базе данных.

1. «Грубое» сравнение входного объекта со всеми объектами базы данных с использованием дескриптора D_1 . Удаление из рассмотрения тех объектов БД, расстояние до которых оказалось больше установленного порога β_1 («непохожие» по критерию D_1 объекты).

2. Определение собственных значений $\lambda_1, \lambda_2, \lambda_3$ ковариационной матрицы C (формула (1)), построенной для входного объекта-запроса, и исключение из рассмотрения тех объектов БД, для которых $\sum_{i=1}^3 |\lambda_i^k - \lambda_i| > \beta_2$, где k – номер объекта в базе данных, β_2 – некоторое пороговое значение.

3. Поиск в оставшемся множестве объектов с применением более точного дескриптора GASD (SPIN). Отметим, что с целью повышения точности распознавания возможно применение не одного конкретного дескриптора, а нескольких (GASD, SPIN, D_1). В данном случае предлагается отсортировать все объекты базы данных по расстоянию до объекта, который требуется распознать, ставя каждому объекту БД в соответствие определенный ранг (его номер в упорядоченном массиве). Количество массивов определяется числом используемых дескрипторов, а итоговый ранг каждого объекта вычисляется как среднее арифметическое его рангов в каждом массиве. Тогда ближайшим «похожим» объектом базы данных является тот, который имеет минимальный ранг.

3. Экспериментальные исследования

В качестве исходной базы трехмерных моделей выбрана известная и доступная для свободного использования база Princeton Shape Benchmark [19], которая часто используется для апробации различных методов решения задач поиска, классификации, кластеризации объектов. Она состоит из 1814 полигональных моделей объектов различных классов (всего 92 класса). В качестве моделей объектов сцены в настоящей работе использованы исходные объекты этой базы, подвергнутые тем или иным преобразованиям / искажениям. Для каждой полигональной модели осуществлялась ее конвертация в облако точек, при этом в качестве меры регулировки плотности (количества) точек облака использовались различные значения коэффициента K , связывающего площадь поверхности полигональной модели с итоговым количеством точек. Далее к полученным облакам применялись различные аффинные преобразования, такие как сдвиг, масштабирование, поворот. Заметим при этом, что даже без применения преобразований процесс конвертации полигональной модели в облако точек сам по себе носит случайный характер.

На рис. 1 представлены графики зависимостей точности распознавания объектов от плотности облаков точек, полученные при использовании дескрипторов D_1 , GASD, SPIN в отдельности, а также в результате их комбинированного применения.



Рис. 1. Точность распознавания при различной плотности облаков точек
Fig. 1. Recognition accuracy at different point cloud densities

Установлено, что при адекватном подборе пороговых значений β_1 и β_2 использование предложенной в работе методики последовательного применения дескрипторов и раннего отсеивания заведомо «непохожих» объектов точность распознавания не снижает, при этом наблюдается существенное повышение быстродействия. На рис. 2 приведены соответствующие графики зависимости времени распознавания от параметра β_1 при фиксированных K и β_2 ($K = 3\ 000$; $\beta_2 = 0$).

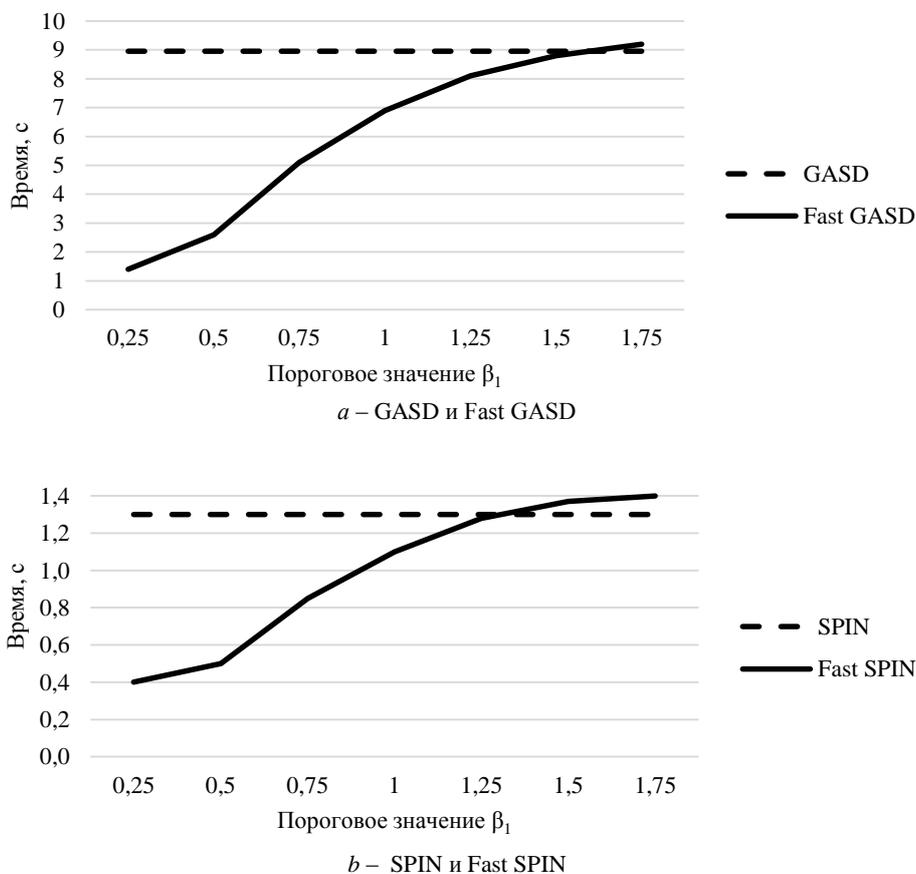


Рис. 2. Суммарное время распознавания 100 объектов при различных значениях параметра β_1
Fig. 2. Total recognition time of 100 objects at different values of parameter β_1

При изменении параметра β_2 наблюдается аналогичный эффект: с уменьшением порогового значения происходит раннее исключение из рассмотрения все большего числа объектов и, как следствие, повышается скорость распознавания. На рис. 3 приведены графики зависимости времени распознавания от параметра β_2 при фиксированных K и β_1 ($K = 3\ 000$; $\beta_1 = 0$).

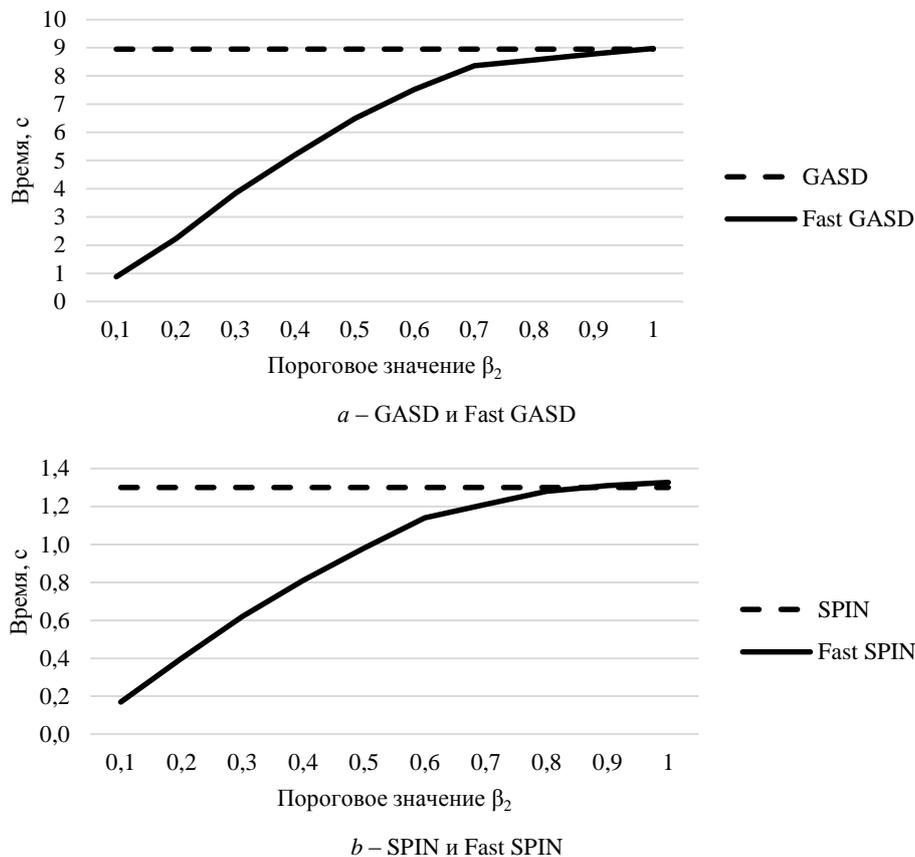


Рис. 3. Суммарное время распознавания 100 объектов при различных значениях параметра β_2
 Fig. 3. Total recognition time of 100 objects at different values of parameter β_2

Точность распознавания в зависимости от параметра β_2

Значение параметра β_2	0,1	0,2	0,3	0,4	0,5	0,6	0,7
Точность Fast SPIN, %	94,85	94,54	94,31	94,20	94,20	94,20	94,20
Точность Fast GASD, %	94,76	94,09	93,76	93,42	93,20	92,86	92,86

Заметим, что уменьшение значения параметра β_2 приводит не только к увеличению скорости поиска, но и к повышению точности распознавания (таблица).

Заключение

Настоящая работа посвящена решению актуальной задачи распознавания трехмерных объектов, представленных облаками точек. В качестве метода решения используются глобальные дескрипторы формы объектов. Авторами предложена оригинальная методика последовательного применения дескрипторов, позволяющая существенно увеличить скорость распознавания объектов, в некоторых случаях на порядок. Основная идея заключается в том, что поиск в некоторой базе известных объектов (эталонов) осуществляется в 2 этапа. На первом шаге из рассмотрения исключаются объекты, заведомо «непохожие» на входной объект, который требуется распознать. Это достигается путем применения не очень точных, но быстрых методов, таких как использование функции формы D_1 или

сравнение собственных значений, получаемых в процессе расчета главных компонент. На втором этапе для поиска в оставшемся множестве объектов применяются более точные дескрипторы.

Для повышения точности идентификации в работе также предложена методика комбинированного использования нескольких дескрипторов, как требующих предварительного выравнивания моделей (GASD, SPIN), так и не требующих такового (D_1). Для построения критерия выбора наиболее подходящего объекта из базы на основании результатов сравнения значений каждого дескриптора предложен ранговый подход. Экспериментальные исследования показали, что применение предлагаемой методики дает бóльшую точность идентификации объектов, чем любой из дескрипторов в отдельности.

ЛИТЕРАТУРА

1. Natao N., Kagami S. Cluster-Based SJPDAFs for Classification and Tracking of Multiple Moving Objects // *Field and Service Robotics*. Springer International Publishing, 2015. P. 303–317.
2. Martínez M.A., Martínez J.L., Morales J. Motion Detection from Mobile Robots with Fuzzy Threshold Selection in Consecutive 2D Laser Scans // *Electronics*. 2015. V. 4, No.1. P. 82–93.
3. Yan Zhuang, Guojian He, Huosheng Hu, Zhenwei Wu novel outdoor scene-understanding framework for unmanned ground vehicles with 3D laser scanners // *Transactions of the Institute of Measurement and Control*. 2015. V. 37, No. 4. P. 435–445.
4. Rejas J.I. et al. A varying density 3D laser scanner for unmanned ground vehicles mapping and obstacle detection // *Design of Circuits and Integrated Circuits (DCIS), 2014 Conference on*. IEEE, 2014. P. 1–6.
5. Besl P., McKay H. A method for registration of 3-d shapes // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1992. V. 14, No. 2. P. 239–256.
6. Chen Y., Medioni G. Object modelling by registration of multiple range images // *Image and Vision Computing*. 1992. V. 10. No. 3. P. 145–155.
7. Дышкант Н.Ф., Местецкий Л.М. Сравнение однолистных поверхностей, полученных при 3D сканировании // *Труды 18-й международной конференции по компьютерной графике и машинному зрению ГрафиКон'2008*. М., 2008. С. 270–277.
8. Dyshkant N. Measures for Surface Comparison on Unstructured Grids with Different Density // *Lecture Notes in Computer Science: Discrete Geometry for Computer Imagery*. 2011. V. 6607. P. 501–512.
9. Дышкант Н.Ф. Эффективные алгоритмы сравнения поверхностей, заданных облаками точек : дис. ... канд. физ.-мат. наук. М., 2011.
10. Lowe D.G. Object recognition from local scale-invariant features // *Proc. of the International Conference on Computer Vision (ICCV '99)*. 1999. V. 2. P. 142–150.
11. Calonder M., Lepetit V., Strecha C., Fua P. BRIEF: Binary Robust Independent Elementary Features // *Proc. of the 11th European Conference on Computer Vision, Heraklion, Crete, Greece*. 2009. Pt. IV. P. 778–792.
12. Leutenegger S., Chli M., Siegwart R.Y. BRISK: Binary Robust Invariant Scalable Keypoints // *Proc. of the 2011 International Conference on Computer Vision (ICCV '11)*. 2011. P. 2548–2555.
13. Osada R., Funkhouser T., Chazelle B., Dobkin D. Shape distributions // *ACM Transactions on Graphics (TOG)*. 2002. V. 21, is. 4. P. 807–832.
14. Vranić D.V., Saupe D.V. Description of 3D-shape using a complex function on the sphere // *Proc. of the IEEE International Conference on Multimedia and Expo (ICME 2002)*. Lausanne, 2002. P. 177–180.
15. Левашев С.П., Абраменко А.А. Анализ изображений сравнением замкнутых контуров с помощью Фурье-дескрипторов // *Фундаментальные и прикладные аспекты компьютерных технологий и информационной безопасности : сб. ст. I Всерос. науч.-техн. конф. молодых ученых, аспирантов и студентов*. Ростов н/Д. : Изд-во ЮФУ, 2015. С. 37–40.
16. Do Monte Lima J.P.S., Teichrieb V. An efficient global point cloud descriptor for object recognition and pose estimation // *Proc. of the 29th SIBGRAPI Conference on Graphics, Patterns and Images, Sao Paulo, Brazil, 4–7 October 2016*. P. 56–63.
17. Johnson A.E., Hebert M. Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes // *IEEE Trans. Pattern Analysis and Machine Intelligence*. 1999. V. 21 (5). P. 433–449.
18. Крыловецкий А.А., Черников И.С. Нормализация 3D-модели для вычисления интегрального спинового изображения // *Известия Южного федерального университета. Технические науки*. 2012. № 6. С. 135–139
19. Princeton Shape Benchmark. URL: <http://shape.cs.princeton.edu/benchmark/> (accessed: 31.08.2018).

Поступила в редакцию 23 декабря 2018 г.

Pristupa A.V., Lapatin I.L., Zamyatin A.V. (2019) 3D OBJECT IDENTIFICATION BASED ON GLOBAL SHAPE DESCRIPTORS. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie vychislitel'naja tehnika i informatika* [Tomsk State University Journal of Control and Computer Science]. 48. pp. 57–66

DOI: 10.17223/19988605/48/7

The rapid development of laser scanning technologies leads to the emergence of new challenges and opportunities for the use of these technologies in various applications and fields. Typical examples of such tasks are classification and recognition (identification) of objects. One way to solve such problems is to use global descriptors of object shape, but it is not always possible to achieve the required accuracy with given requirements for the speed of algorithm. Difficulties arise when the base of reference objects in which search is carried out is big, but search algorithm is required to work in real time. The paper proposes a method of sequential application of global descriptors, allowing the first stage to produce a "rough" screening of obviously different objects, and then to apply more accurate algorithms on a significantly reduced object base.

Let's formulate the proposed method of quick search of the object in the database.

1. A "rough" comparison of the input object with all database objects using the D_1 descriptor. Remove/exclude from consideration of those objects of the database, the distance to which turned out to be greater than the established threshold β_1 ("dissimilar" to the criterion D_1 objects).

2. Then we calculate the eigenvalues $\lambda_1, \lambda_2, \lambda_3$ of the covariance matrix, of the coordinates of the set of points P_i of the input object. Exclude from consideration database objects, for which $\sum_{i=1}^3 |\lambda_i^k - \lambda_i| > \beta_2$, k – object index (number) in the database, β_2 – some threshold level.

3. Search the remaining set of objects using a more accurate descriptor (GASD, SPIN). Note that in order to improve the accuracy of recognition, it is possible to use not one specific descriptor, but several (GASD, SPIN, D_1). We suggest sorting all database objects by the distance to the object you want to recognize and each database object is assigned a certain rank (its number in an ordered array). The number of arrays is determined by the number of descriptors used, and the total rank of each object is calculated as the arithmetic mean of its ranks in each array. Then the closest "similar" database object is the one that has the minimum rank. The effect reached at the same time consists in significant increase in speed of identification of an object (in certain cases much and more) without compromising accuracy. To improve the accuracy of object identification, the paper also proposes a method of simultaneous use of existing global shape descriptors. Experimental studies have shown that the application of the proposed method provides greater accuracy of object identification than any of the descriptors individually.

Keywords: 3D object; point cloud; identification; recognition; global shape descriptors.

PRISTUPA Andrey Viktorovich (Candidate of Technical Sciences, Associate Professor, National Research Tomsk State University, Tomsk, Russian Federation).

E-mail: pristupa@sibmail.com

LAPATIN Ivan Leonidovich (Candidate of Physics and Mathematics, Associate Professor, National Research Tomsk State University, Tomsk, Russian Federation).

E-mail: ilapatin@mail.ru

ZAMYATIN Alexander Vladimirovich (Doctor of Technical Sciences, Professor, National Research Tomsk State University, Tomsk, Russian Federation).

E-mail: zamyatin@mail.tsu.ru

REFERENCES

- Hatao, N. & Kagami, S. (2015) Cluster-Based SJPDFAs for Classification and Tracking of Multiple Moving Objects. *Field and Service Robotics*. Springer International Publishing. pp. 303–317.
- Martínez, M.A., Martínez, J.L. & Morales, J. (2015) Motion Detection from Mobile Robots with Fuzzy Threshold Selection in Consecutive 2D Laser Scans. *Electronics*. 4(1). pp. 82–93. DOI: 10.3390/electronics4010082
- Zhuang, Y. et al. (2015) A novel outdoor scene-understanding framework for unmanned ground vehicles with 3D laser scanners. *Transactions of the Institute of Measurement and Control*. 37(4). pp. 435–445. DOI: 10.1177/0142331214541140
- Rejas, J.I. et al. (2014) A varying density 3D laser scanner for unmanned ground vehicles mapping and obstacle detection. *Design of Circuits and Integrated Circuits (DCIS). Conference IEEE*. pp. 1–6. DOI: 10.1109/DCIS.2014.7035576
- Besl, P. & McKay, H. (1992) A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 14(2). pp. 239–256. DOI: 10.1109/34.121791
- Chen, Y. & Medioni, G. (1992) Object modelling by registration of multiple range images. *Image and Vision Computing*. 10(3). pp. 145–155. DOI: 10.1109/ROBOT.1991.132043
- Dyshkant, N. & Mestetsky, L. (2008) [Comparison of univalent surfaces obtained by 3D scanning]. *Proc. of the 18th International Conference on Computer Graphics and Computer Vision GraphiCon'2008*. Moscow: Moscow State University. pp. 270–277. (In Russian).
- Dyshkant, N. (2011) Measures for Surface Comparison on Unstructured Grids with Different Density. *Lecture Notes in Computer Science: Discrete Geometry for Computer Imagery*. 6607. pp. 501–512. DOI: 10.1007/978-3-642-19867-0_42

9. Dyshkant, N. (2011) *Effektivnye algoritmy sravneniya poverkhnostey, zadannykh oblakami toчек* [Effective algorithms for comparing surfaces defined by point clouds]. Physics and Mathematics Cand. Diss. Moscow.
10. Lowe, D. (1999) Object recognition from local scale-invariant features. *Proc. of the International Conference on Computer Vision (ICCV '99)*. 2. pp. 142–150.
11. Calonder, M., Lepetit, V., Strecha, C & Fua, P. (2009) BRIEF: Binary Robust Independent Elementary Features. *Proc. of the 11th European Conference on Computer Vision*. Heraklion, Crete, Greece. Pt. IV. pp. 778–792.
12. Leutenegger, S., Chli, M. & Siegwart, R.Y. (2011) BRISK: Binary Robust Invariant Scalable Keypoints. *Proc. of the 2011 International Conference on Computer Vision (ICCV '11)*. pp. 2548–2555.
13. Osada, R., Funkhouser, T., Chazelle, B. & Dobkin, D. (2002) Shape distributions. *ACM Transactions on Graphics (TOG)*. 21(4). pp. 807–832.
14. Vrani'c, D.V. & Saupe, D.V. (2002) Description of 3D-shape using a complex function on the sphere. *Proc. of the IEEE International Conference on Multimedia and Expo (ICME 2002)*. Lausanne. pp. 177–180.
15. Levashev, S.P. & Abramenko, A.A. (2015) Analiz izobrazheniy sravneniem zamknutykh konturov s pomoshch'yu Fur'e-deskriptorov [Image Analysis by Comparing Closed Loops Using Fourier Descriptors]. In: Veselov, G.E., Sviridov, A.S. & Kuliev, E.V. (eds) *Fundamental'nye i prikladnye aspekty komp'yuternykh tekhnologiy i informatsionnoy bezopasnosti* [Fundamental and Applied Aspects of Computer Technologies and Information Security]. Rostov-on-Don: Southern Federal University. pp. 37–40.
16. Do Monte Lima, J.P.S. & Teichrieb, V. (2016) An efficient global point cloud descriptor for object recognition and pose estimation. *Proceedings of the 29th SIBGRAPI Conference on Graphics, Patterns and Images*. Sao Paulo, Brazil. October 4–7, 2016. pp. 56–63.
17. Johnson, A.E. & Hebert, M. (1999) Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes. *IEEE Trans. Pattern Analysis and Machine Intelligence*. 21(5). pp. 433–449. DOI: 10.1109/34.765655
18. Krylovetsky, A.A. & Chernikov, I.S. (2012) 3d-model normalization for integral spin image estimation. *Izvestiya Yuzhnogo federal'nogo universiteta. Tekhnicheskie nauki – Izvestiya SFEDU. Engineering Sciences*. 6. pp. 135–139. (In Russian).
19. *Princeton Shape Benchmark*. [Online] Available from: <http://shape.cs.princeton.edu/benchmark/> (Accessed: 31st August 2018)

УДК 004.272

DOI: 10.17223/19988605/48/8

A.Y. Filimonov, V.N. Trishin**MODEL-ORIENTED CO-DESIGN OF HOMOGENEOUS COMPUTING SYSTEMS**

The paper was carried out with the partial funding of the scientific project of UB RAS No. 15-7-1-20 "Complex study of the architectures of 2D processor arrays that perform fine-grained processing parallelism and fast algorithms for processing speckled images to create VLSI video processors for real-time systems."

This article proposes an approach to the construction of a technological platform for designing function-oriented processors (FOP) built on the basis of homogeneous computing environments (OVS). A feature of the proposed approach is the use of co-design - a simultaneous and interconnected design of hardware (HW) and software (SW) FOP components. The paper presents proposals for the implementation of this approach, based on the use of the principle of model-based design, which allows you to make the co-design process iterative and effectively distribute data processing procedures between HW and SW - components of FOP. Based on the consideration of the experience in implementing such solutions, the paper presents the composition of functional subsystems of technological platforms, a diagram of information interaction between these subsystems based on intensive using of web-technologies at all stages of the design and operation of FOP.

Keywords: function-oriented processors; homogeneous computing systems; model-oriented design; parallel programming.

The current stage of the information systems development is characterized by an exponential growing amount of IoT (Internet of Things) connections and the volume of data they transfer increasing. According to forecasts, by 2020, the number of "smart things" connected to the Internet will be more than 20 times greater than human population [1], while the total amount of stored and processed data will reach $40 \cdot 10^{10}$ B [2]. The sharp increase in the amount of data generated by various IoT sensors led to the emergence of the concept of "fog computing", the main idea of which is to move the data preprocessing processes as close as possible to the points where these data were generated [3]. All these changes have a significant impact on the development of software and hardware platforms for real-time systems (RTEs), making such solutions urgent that provide scalability of the architectures of embedded computing subsystems, high degree of fault tolerance, minimization of power consumption, providing the maximum possible performance with given limitations on hardware costs [4]. This is most clearly manifested, for example, in such a class of RTE, when the embedded processor should provide a performance of the order of gigaflops with a mass of not more than 10 grams [5]. Experts note the need to conduct research and development of computing systems with mass parallelism, in particular, systolic architectures, which potentially realize the maximum possible performance when processing large amounts of data in real time at a new level of development of microelectronics [6]. Recently there are more and more examples of successful implementation of homogeneous components of computer systems [7]. The manufacturability of homogeneous computing systems (HCS) opens up wide prospects for the use of functionally-oriented processors (FOPs) based on them in various mobile systems, where high computational performance, low power consumption and resistance to failures are required. The programmability of the HCS architecture determines the variability in the implementation of hardware (HW) and software (SW) types of FOP, which requires the implementation of optimization procedures at all stages of its construction. Procedures for simultaneous and mutually conditioned design - co-design - make it possible to design of HW and SW HCS-FOP in such a way that the SW takes into account the available HW capabilities and vice versa, making changes to the HW project that would maximize the implementation of requirements for SW. Model-Driven Engineering (MDE) approach makes the design process of HCS-FOP iterative,

allows efficient distribution of data processing procedures between HW and SW, and the step-by-step verification of design solutions ensures the progressive nature of the design process and the achievement of predictable performance characteristics of the HCS-FOP, which is especially important when using such kind of devices in the IoT sphere [8]. In this paper considered an approach to the creation of a technological platform that is capable of ensuring the fulfillment of all stages of design, production, and operation of an HCS- FOP.

1. Main features of the HCS design process

A homogeneous computing system is a computer system consisting of identically connected identical processor elements (PEs), each of them is programmatically configured to perform an arithmetic or logical function, as well as to implement a data exchange with neighboring elements [9]. Programming HCS is a process of mutually coordinated adjustment of each of the PE array to perform corresponding processing and data transfer operations, which are necessary for mapping the data flow graph of the target task into the PE grid. By this process, the data stream graph of the programmable function must be transformed into a lattice graph [10] in a two-dimensional PE space, taking into account the design limitations of a particular implementation of the HCS [9]. Since the result of programming the HCS is the architecture of the computer, so the main feature of the design process for HCS is the need for its simultaneous execution with the programming process. The process of designing the HCS is designed to ensure the construction of a computer system that satisfies the set of algorithmic, constructive and operational requirements presented to it [11]. For HCS, which have to be operated in straightly limited conditions, it is very important to observe operational and design requirements, such as device's overall dimensions and its power consumption [9].

In addition to the requirements listed above, HCS that are designed to pre-process IoT sensors (Pre-processing level, level 3 of the fog computing architecture [12]) must meet the requirements for the components of the OpenFog Consortium Architecture [3]. The advantage of using HCS FOP at this level of fog computing architecture is that such systems are able to work indefinitely for a long time, implementing the principle of controlled degradation, which in this case can be provided by remote programming of HCS directly during operation [9]. In order to take advantage of this, it should be possible to interact with the design platform, which significantly limits the possibility of using local applications for its implementation [8]. The platform for designing computer systems needed to fulfill similar requirements and constraints are presented should provide the ability to predict (monitor) the degree to which these requirements are met during the design process and should have appropriate mechanisms to manage this process. The conducted researches show that application of the model-oriented approach in the design of embedded systems allows to significantly improve the design quality [13]. The main idea of this approach is to apply the procedures for sequential transformation of the model of the projected system in such a way that at each subsequent stage this model begins to correspond to a larger list of requirements for the system being designed than the model of the previous stage [14]. Thanks to the automatic or automated generation of such models, the use of multi-level verification becomes possible, which allows to identify and eliminate errors in the early design stages [15]. That is why it seems particularly expedient to apply the model-oriented approach to the co-design of the HCS fop, which in this case involves the construction of the following models:

- computational model that allows you to determine the requirements for the accuracy of data representation and the error in solving the target task;
- architectural model, which allows to determine the degree of fulfillment of design requirements;
- complex model, which allows to determine the degree of implementation of the entire set of requirements for HCS.

In case of using the model-oriented approach, the technological cycle of co-design of HCS FOP begins with the construction of a computational model of the task (task) and represents a sequence of stages of transformation of this model up to the stage of creating a complete set of design documentation necessary for production and programming of the computer system.

2. Construction of HCS FOP computational model

The traditional approach to constructing computational models based on the use of block diagrams and programs in imperative programming languages allows one to adequately reflect the processes typical for computers with von Neumann architecture, exactly because this approach was especially created to describe computational processes that are controlled by the command flow (Control Flow). As the consequence, using of this approach to describe processes in computing systems that have an unconventional architecture, in particular systems that are controlled by Data Flow, causes significant problems [16]. The use of graphical tools helps to overcome some of those problems, because it allows to adequately, concisely and visually display the structure and interaction of information or control flows, as evidenced by the using of graphic and visual methods among domestic [17] and foreign [18] developers of programming platforms for systems with mass parallelism calculations. Considering capability of using such solutions for constructing the computational model of HCS, however, it should be kept in mind that the technology of graph-symbol programming (GSP) [17] is designed to describe the interaction of distributed computing systems through shared memory areas, and therefore, first of all, is directed on constructing the of control flows graph, but not data flow graph. At the same time, some of principles and concepts underlying the GSP platform, for example, such as the polymorphism of the base modules (computable functions) and the certification of "actors", appear to be quite universal and can be used productively in the creation of the HCS programming subsystem. Much closer to the HCS programming platform is the GASPARD (Graphical Array Specification for Parallel and Distributed Computing) platform, which provides a solution to a set of tasks that arise during the simultaneous design of real-time embedded (Real-time / Embedded-RT / E) hardware and software. The GASPARD platform is based on the concept of Model-Driven Engineering (MDE), which, in accordance with the basic requirements of the object management group (OMG) presented in Recommendation [19], allows to create a hierarchies of interrelated models of software and hardware components of RT / E systems real time. The basic language used to build the model of the computational process in GASPARD is the graphical functional language Array-OL [20]. The functional paradigm of the language in this case can be explained by the fact that the use of imperative languages for programming systems with the architecture controlled by data flows causes serious problems associated with side effects such as data racings and data availability limitations [21]. That is why to ensure the expressiveness and adequacy of describing the computational process in streaming computing systems with massive parallelism, programming platforms based on imperative languages are compelled to adopt the basic principles of the declarative programming concept [22, 23]. In this case, the process of solving the target task (TT) can be represented in the form of a hierarchical composition of functional modules (FM), each of which generates a stream of output data, taking as parameters the output data streams of the lower hierarchy level or the target task input data streams. If the FM will be determined parametrically polymorphic, special procedures or constructions must be presented in the system that provide control and agreement of the types of actual parameters of the FM. The process of applying a scalar (or incomplete) function to a multidimensional flow of input data, in declarative programming systems, have special importance since this process forms the basis for parallelizing computations.

For example, in Wolfram Language (WL) [24], some of the scalar functions can be applied to multi-dimensional arguments. A somewhat different approach presented in SequenceL language, where the mechanisms of implicit parallelization of computations are also actively applied [25]. The developers of SequenceL, however, recognize that the use of a fully automatic approach when matching the dimension of the operands can lead to errors localization difficulties [26]. More promising for implementation in the system of programming of HCS seems to be an automated procedure for structural matching, which can be performed by selecting the appropriate data stream structures when calling a specific function in the composition.

Thus, for constructing a computational model of the FOS HCS, it seems expedient to use one of the modern functional or hybrid programming languages that allows describing the TT in the form of a hierarchical composition of functional modules. Additional advantages in this case are the use of symbolic

programming languages (LISP, WL) because it allows the formation of a functional connection graph, which is necessary for performing the function stacking in the HCS matrix.

3. Realization of technological cycle of co-design of HCS

As a basic component of the HCS co-design platform, a subsystem for programming data processing functions (a programming subsystem), an architecture layout subsystem (a layout subsystem), and a simulation subsystem (an emulation subsystem) are proposed in the work. The programming subsystem provides the transformation of a computational model into an architectural model, the layout subsystem transforms the architectural model into a complex one, the emulation subsystem provides a comprehensive verification of selected for implementation the software and hardware solutions of the FOS HCS. To ensure the optimization and adaptation of the emerging HCS architecture, the process of information interaction between subsystems have to be iterative and interactive one. The subsystems themselves are built according to the block-modular approach in order to ensure the possibility of further development of the programming platform. The operation of the HCS co-design subsystem begins after the computation model of the target task is defined, represented by the graph of the function blocks connected by the data streams. The programming subsystem is designed to map the TT algorithm to the operational basis of the HCS and the corresponding distribution of data processing functions between the PE groups. To provide debugging for this process control data of the task function (CDF) have to be used, for example, to provide control the accuracy of calculations. As an interface tool of the programming subsystem, in this case it is advisable to use the additional software "electronic notebook" - (EN) [27]. In modern programming systems, EN provides functions such as editing, formatting, text saving, on-line monitoring of syntax structures (including syntax-highlighting) [28] and context-sensitive support for the programming process [24]. The use of EN as the interface and binding element of the programming platform for IOT becomes even more useful since it allows to take into account all the programming features inherent in this platform. After debugging process completion, the programming system generates a complete data flow graph, which determine needed operations for every PE. The layout subsystem is designed for laying a complete graph of data flows of the TT in the HCS lattice and ensures the placement of programmed PEs on the HCS matrix and creation of links between them. In fulfilling this task, the construction limitations inherent in the specific implementation of the HCS (CL) should be ensured. In general, these requirements may limit:

- the total number of PEs that can be used to solve the TT;
- geometric parameters of the region where these PEs can be located;
- the mutual position of PE groups that are involved in the implementation of interrelated functions.

The final solution of the layout problem is the one of possible mappings decisions a complete graph of data streams in the PE matrix, which ensures that all CLs are fulfilled in condition of reaching required value of the predicted decision time of the TT. This solution will be formulated in the technological language of the styling specifications, which is the final result of the processes of programming and design of the HCS. The simulation subsystem is designed to simulate the implementation of the prepared decision of the TT on the HCS and provides final control over the implementation of the requirements for the implementation of this TT.

Figure 1 presents the composition of the components of the HCS co-design platform and the scheme of information interaction between them. Blocks 1.1-3.2 form the programming subsystem, block 4 corresponds to the layout subsystem, block 5 represents the simulation subsystem, and the remaining blocks provide an iterative and interactive mode of interaction between the main platform subsystems. This is especially important for responsible applications, for example, processing signal flows from mobile sensor outputs robot for the purpose of prompt decision making in the regime of hard real time from the command post at a considerable distance from the control direct object

An example of the implementation of the web interface of the simulation subsystem is shown in Fig. 2. In response to the request, the client's browser receives a page of the cloud-based integrated development environment (IDE), in which the PE link graph is edited. The PEs themselves are displayed using built-in SVG elements, which are a copy of the librarian one.

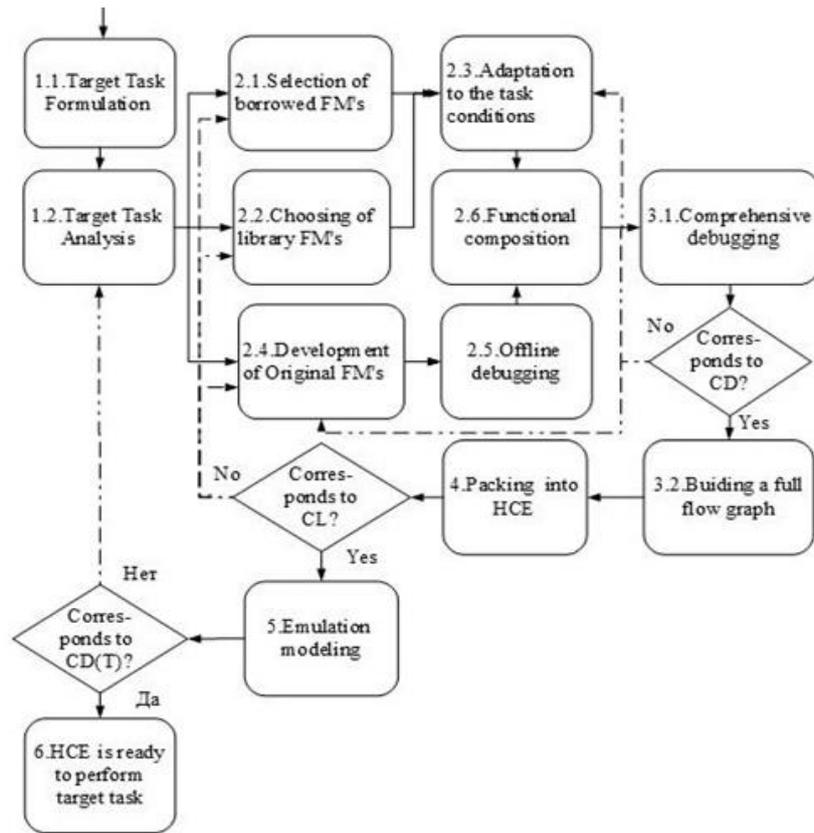


Fig. 1. Scheme of information interaction of the HCS co-design platform components

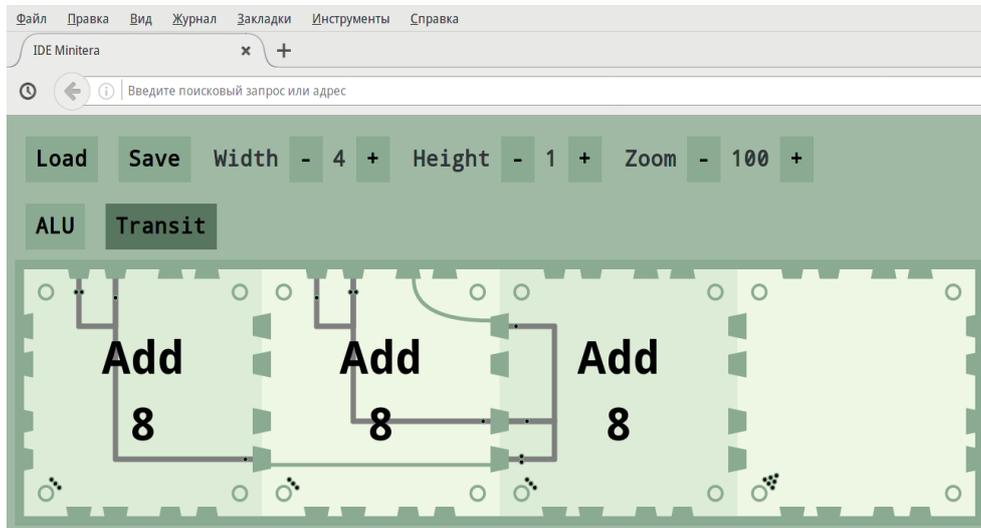


Fig. 2. Implementing the web interface of the simulation subsystem

The visibility of the symbols of the operating mode of the PE is adjusted using the CSS style sheet. On the client side, a JavaScript code is executed that passes data about the user's action to the server and receives new rules for the style sheet in response. On the server side, Perl scripts are accessed through SQL queries to the database, which stores data about the states of processor elements and user sessions. Since the modern concept of managing network components involves the use of the NETCONF or RESTCONF protocols [29] to manage device configurations, as a promising direction for the further development of the proposed approach should be the integration into the HCS co-design platform of the communication model in the YANG language [30].

Conclusion

Currently, the problem of effective HCS programming still not solved, and the main reason for this is the lack of developed and mature programming and design technologies for this computing architectures. In this paper the basic principles of construction of a technological design platform which allows to unite in a single process and automate all stages of co-design of HCS are considered. The key components of this platform are the functional programming subsystem, the information algorithm stacker on the two-dimensional PE array (the layout subsystem), and the emulator (simulation subsystem). The platform assumes the use of cloud services at all stages of development and development of HCS programs. As the first stage of the work on the creation of the technological platform for HCS programming, an intelligent graphical editor has been developed that allows verifying the description of the information flow graph and automating the procedures for laying software projects in the cellular space of the HCS.

REFERENCES

1. CompTIA. (2015) *Sizing Up the Internet of Things*. [Online] Available from: <https://www.comptia.org/resources/sizing-up-the-internet-of-things> (Accessed: 22nd May 2018).
2. Audin, G. (2015) *The Network Impact of Big Data*. [Online] Available from: <https://www.nojitter.com/post/240170228/the-network-impact-of-big-data> (Accessed: 22nd May 2018).
3. OpenFog Consortium Architecture Working Group. (2016) *White Paper.OpenFog Architecture Overview*. [Online] Available from: <https://www.openfogconsortium.org/wp-content/uploads/OpenFog-Architecture-Overview-WP-2-2016.pdf> (Accessed: 22nd May 2018).
4. Chiang, M. & Zhang, T. (2016) Fog and IoT: An Overview of Research Opportunities. *IEEE Internet of Things Journal*. 3(6). pp. 854–864. DOI: 10.1109/JIOT.2016.2584538
5. Kuhn, A. (2014) *Review of Novel Computing Architectures for Neural Applications*. [Online] Available from: <https://pdfs.semanticscholar.org/fab9/f4e8834900c0f66d10d7d0e8b786915c7271.pdf> (Accessed: 22nd May 2018).
6. Master, P. (n.d.) *A new computing architecture for Big Data and AI applications*. [Online] Available from: <http://impactvc.com/a-new-computing-architecture-for-big-data-and-ai-applications> (Accessed: 22nd May 2018).
7. Jouppi, N.J. et al. (2017) In - Datacenter Performance Analysis of a Tensor Processing Unit. *Proc. of the 44th Annual International Symposium on Computer Architecture, ISCA*. pp. 1–12. DOI: 10.1145/3140659.3080246
8. Lukin, N.A., Filimonov, A.Yu. & Trishin, V.N. (2017) Cloud computing environment of homogeneous computing systems. In: Dubrov, D.V. *Programming languages and compilers – 2017*. Rostov-on-Don: [s.n.]. pp. 181–184. (In Russian).
9. Lukin, N.A. (2014) Functionally-oriented processors are key components of built-in supercomputers for real-time systems. *Izvestiya Yuzhnogo federal'nogo universiteta. Tekhnicheskie nauki – Izvestiya SFEDU. Engineering Sciences*. 12. pp. 52–64. (In Russian).
10. Voevodin, V.V. (1986) *Mathematical Models and Methods in Parallel Processes*. Moscow: Nauka.
11. Lukin, N.A. (2010) [Functionally oriented processors with a homogeneous architecture for the implementation of algorithms of onboard control systems]. *Parallel Computing and Control Problems*. Proc. of the 5th International Conference. Moscow. pp. 1177–1184. (In Russian).
12. Atlam, H.F., Walters, R.J. & Wills, G.B. (2018) Fog Computing and the Internet of Things: A Review. *Big Data and Cognitive Computing*. 2(2). DOI: 10.3390/bdcc2020010.
13. Gamatié, A., Le Beux, S., Piel, É., Ben Atitallah, R., Etien, A., Marquet, P. & Dekeyser, J.-L. (2011) A model-driven design framework for massively parallel embedded systems. *ACM Trans. Embed. Comput.Syst.* 10(4). Article 3936. DOI: 10.1145/2043662.2043663
14. Adler, R., Schaefer, I., Schuele, T. & Vecchie, E. (2007) From Model-Based Design to Formal Verification of Adaptive Embedded Systems. *ICFEM 2007, LNCS 4789*. pp. 76–95. DOI: 10.1007/978-3-540-76650-6_6
15. Rovers, K.C., van de Burgwal, M.D., Kuper, J., Kokkeler, A.B.J. & Smit, G.J.M. (2011) Multi-domain transformational design flow for embedded systems. *International Conference on Embedded Computer Systems (SAMOS 2011)*. pp. 93–101. DOI: 10.1109/SAMOS.2011.6045449
16. Lukin, N.A. & Filimonov, A.Y. (2017) Software technologies for homogeneous computing environment. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science*. 40. pp. 61–70. (In Russian). DOI: 10.17223/19988605/40/7
17. Kovartsev, A.N. & Zhidchenko, V.V. (2011) *Methods and means of visual parallel programming. Automation of programming*. Samara: Samara State Aerospace University.
18. Devin, F., Boulet, P., Dekeyser, J.-L. & Marquet, P. (2002) GASPARD a visual parallel programming environment. *Parallel Computing in Electrical Engineering, PARELEC '02*. Proc. International Conference. DOI: 10.1109/PCEE.2002.1115225

19. Object Management Group, Inc. (n.d.) *UML Profile for MARTETM: Modeling and Analysis of Real-time Embedded Systems*. [Online] Available from: <http://www.omg.org/spec/MARTE/1.1> (Accessed: 22nd May 2018).
20. Boulet, P. (2008) Formal Semantics of Array-OL, a Domain Specific Language for Intensive Multidimensional Signal Processing. *Research Report RR-6467, INRIA*.
21. Johnston, W.M., Hanna, P. & Millar, R.J. (2004) Advances in dataflow programming languages. *ACM Computing Surveys (CSUR)*. 36(1). pp. 1–34. DOI: 10.1145/1013208.1013209
22. Böhm, A.P.W., Hammes, J., Draper, B.A., 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
23. Hammarlund, P. & Lisper, B. (1993) Data Parallel Programming: A Survey and a Proposal for a New Model. *Technical Report 93/8-SE, Department of Teleinformatics Royal Institute of Technology, Sweden*.
24. Wolfram, S. (2017) *An Elementary Introduction to the Wolfram Language*. 2nd ed. Wolfram Media.
25. Nemanich, B., Cooke, D. & Rushton, N. (2010) Sequencel: transparency and multicore parallelisms. *DAMP '10 Proceedings of the 5th ACM SIGPLAN workshop on Declarative Aspects of Multicore Programming*.
26. Cooke, D., Rushton, N., Nemanich, B. & Watson, R.G. (2008) Normalize, transpose, and distribute: An automatic approach for handling nonscalars. *ACM Transactions on Programming Languages and Systems*. 30(2). DOI: 10.1145/1330017.1330020
27. Project Jupyter. (n.d.) *Jupyter Notebook Documentation Release 5.1*. [Online] Available from: <https://media.readthedocs.org/pdf/jupyter-notebook/latest/jupyter-notebook.pdf>. (Accessed: 22nd May 2018).
28. The IPython Development Team. (n.d.) *IPython Documentation Release 6.2.0*. [Online] Available from: <http://ipython.readthedocs.io/en/latest/whatsnew/development.html>. (Accessed: 22nd May 2018).
29. Xu, H., Wang, C., Liu, W. & Chen, H. (2012) NETCONF-based Integrated Management for Internet of Things using RESTful Web Services. *International Journal of Future Generation Communication and Networking*. 5(3). pp. 73–82.
30. Scheffler, T. & Bonness, O. (2017) Manage Resource-constrained IoT Devices Through Dynamically Generated and Deployed YANG Models. *Proc. of the Applied Networking Research Workshop*. 1. pp. 42–47. DOI: 10.1145/3106328.3106331

Received: December 6, 2018

Filimonov A.Y., Trishin V.N. (2019) MODEL-ORIENTED CO-DESIGN OF HOMOGENEOUS COMPUTING SYSTEMS *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie vychislitel'naja tehnika i informatika [Tomsk State University Journal of Control and Computer Science]*. 48. pp. 67–73

DOI: 10.17223/19988605/48/8

Филимонов А.Ю., Тришин В.Н. МОДЕЛЬНО-ОРИЕНТИРОВАННЫЙ КО-ДИЗАЙН ОДНОРОДНЫХ ВЫЧИСЛИТЕЛЬНЫХ СРЕД. *Вестник Томского государственного университета. Управление, вычислительная техника и информатика*. 2019. № 48. С. 67–73

В работе предложен подход к построению технологической платформы проектирования функционально-ориентированных процессоров (ФОП), построенных на основе однородных вычислительных сред (ОВС). Особенностью предлагаемого подхода является использование ко-дизайна – одновременного и взаимосвязанного проектирования аппаратного (HW) и программного (SW) обеспечения таких ФОП. В работе приведены предложения по реализации данного подхода, основанные на использовании принципа модельно-ориентированного проектирования, который позволяет сделать процесс ко-дизайна итеративным и эффективно распределить процедуры обработки данных между HW- и SW-компонентами ФОП. На основе рассмотрения опыта реализации подобных решений в работе представлен состав функциональных подсистем технологических платформ, приведена схема информационного взаимодействия между этими подсистемами, предполагающая интенсивное применение web-технологий на всех стадиях проектирования и эксплуатации ФОП.

Ключевые слова: однородные вычислительные системы; модельно-ориентированный дизайн; параллельное программирование.

FILIMONOV Alexander Yurievich (Associate Professor of the Department of Information Technologies and Automation in the Institute of Radioelectronics and Information Technologies at Ural Federal University, Yekaterinburg, Russian Federation).
E-mail: a.filimonov@urfu.ru

TRISHIN Vasily Nikolaevich (Junior Researcher of the Institute of Engineering Science, Ural Branch of the Russian Academy of Sciences, Yekaterinburg, Russian Federation).
E-mail: trishinvn@yandex.ru

УДК 519.17

DOI: 10.17223/19988605/48/9

Д.И. Черемисинов, Л.Д. Черемисинова

**ПОИСК ПОДСХЕМ В КМОП СХЕМЕ ИЗ ТРАНЗИСТОРОВ
МЕТОДОМ GRAPH MINING**

Graph Mining – одно из направлений интеллектуального анализа данных, в котором объемные комплексные данные представлены в виде графов, а анализ ведется для того, чтобы получить новые знания. Задачей Graph Mining является обнаружение в графе типовых шаблонов. Общепринятым видом таких шаблонов являются часто встречающиеся подграфы. Рассматривается задача поиска часто встречающихся подграфов в большом графе и обсуждается применение этой задачи в преобразовании плоской КМОП (комплементарная структура металл-оксид-полупроводник) схемы из транзисторов в формате SPICE в иерархическую схему.

Ключевые слова: кластеризация графа; поиск часто встречающихся подграфов; изоморфизм подграфов; цветовое кодирование графа; КМОП схемы из транзисторов.

В последние годы наблюдается повышенный интерес к разработке алгоритмов интеллектуального анализа данных, работающих на графах. Такие графы возникают естественно в ряде различных областей, таких как обнаружение сетевых атак, семантический web, поведенческое моделирование, перепроектирование СБИС, анализ социальных сетей и классификация химических соединений. Когда данные сложны и имеют много взаимосвязей, на первый план выходит задача поиска знаний об их структуре. Важную роль в методах интеллектуального анализа для данных большого объема, моделируемых графами, играют алгоритмы поиска часто встречающихся подграфов. Задача поиска часто встречающихся подграфов состоит в нахождении в заданном графе всех тех подграфов, которые встречаются в нем с частотой большей, чем заданное значение. Идентификация часто встречающихся графов / подграфов в базе данных или в одном большом графе может использоваться для задач классификации, кластеризации графов, построения индексов при поиске подграфов.

По аналогии с программированием преобразование иерархической схемы электронного устройства в схему, состоящую исключительно из примитивных элементов, естественно назвать компиляцией. Обратный процесс, в результате которого из плоской транзисторной схемы строится иерархическая транзисторная схема, можно назвать декомпиляцией. Декомпилятор [1] как программа, восстанавливающая иерархическое описание схем, является одним из инструментов верификации лейаута, или перепроектирования (reengineering), схем [2]. Так же как при декомпиляции программ, целью декомпиляции схемы является замена представления схемы на низком (транзисторном) уровне более высокоуровневым ее представлением. В отличие от программного процесса декомпиляция схем не является языковой трансформацией, хотя декомпилятор использует определенный формат (формальный язык представления данных) схем. Языком, используемым декомпилятором схем, как правило, является формат SPICE (Simulation Program with Integrated Circuit Emphasis) для обмена электрическими схемами [3]. Формат SPICE позволяет описывать как схемы транзисторного уровня, так и иерархические.

В формате SPICE электрические схемы состоят из элементов, которые соединены друг с другом цепями. Главной частью описания схемы в формате SPICE является список транзисторов, в котором для каждого вывода транзистора указано имя цепи, соединяющей его с остальными частями схемы. Например, схема статического КМОП инвертора имеет по одному n -МОП и p -МОП транзистору, для каждого из четырех выводов (сток, затвор, исток, подложка) которых приведены имена цепей (рис. 1). Удобной моделью для поиска подсхем в исходной транзисторной схеме является помеченный неориентированный двудольный граф. Одну долю графа составляют вершины, соответствующие выводам

элементов и портам схемы (выводам всей электрической схемы), а другую – вершины, соответствующие цепям – соединениям между выводами.

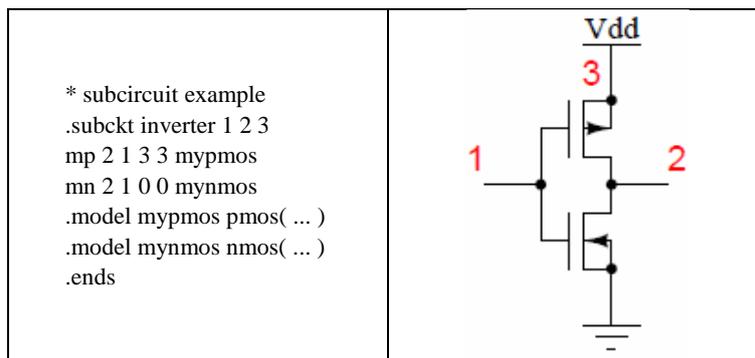


Рис. 1. Описание КМОП инвертора в формате SPICE и его электрическая схема
Fig. 1. Description of the CMOS inverter in SPICE format and its electrical circuit

На рисунках этот граф компактнее представляется в виде его реберного графа. Для графа G реберным называется граф $L(G)$, любая вершина которого представляет ребро графа G и две вершины графа $L(G)$ смежны тогда и только тогда, когда соответствующие им ребра смежны в G . Пометками (раскраской) элементов транзисторного уровня являются названия выводов транзисторов. На рис. 2 в графе КМОП схемы транзисторного уровня у транзисторов не показан вывод подложки, зеленым обозначен вывод «сток», красным – «исток», синим – «затвор».

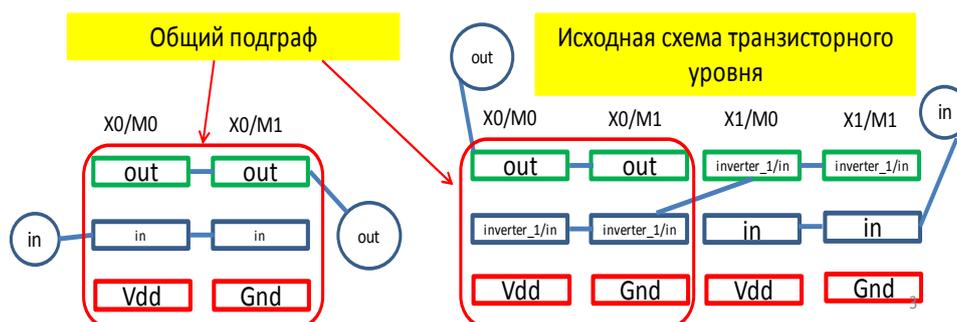


Рис. 2. Декомпиляция – поиск и извлечение подсхемы
Fig. 2. Decompiling – subcircuit recognition and extraction

Современные цифровые КМОП схемы содержат до миллиарда примитивных элементов на транзисторном уровне. КМОП – набор полупроводниковых технологий построения интегральных микросхем и соответствующая ей схемотехника микросхем. Для построения иерархического структурного описания нужно выделить наборы взаимосвязанных транзисторов в схеме в качестве отдельных компонентов, т.е. найти подсхемы в исходной плоской схеме (см. рис. 2). После замены подсхем из транзисторов элементами описание схемы становится двухуровневым.

Если искомые подсхемы библиотечных элементов заданы, т.е. известна библиотека подсхем, то образцы графов (шаблоны) для поиска формируются из описания этих библиотечных элементов. Задача декомпиляции в этом случае сводится к задаче поиска в исходном графе подграфов, изоморфных заданному [4].

В дальнейшем задача декомпиляции рассматривается только для случая, когда библиотека подсхем не известна. Если нужно распознать библиотеку подсхем, то задача в такой постановке поставлена некорректно, так как любой фрагмент, содержащий целое число транзисторов, может считаться частью библиотечной подсхемы. Нужны дополнительные критерии, позволяющие выделять подсхемы. Как и в подходе Graph Mining, далее рассматривается задача выделения в качестве подсхем фрагментов транзисторной КМОП схемы, которые являются часто встречающимися подграфами.

Алгоритм наивного поиска часто встречающихся подграфов состоит из двух операций. Первая операция разыскивает все подграфы-кандидаты для данного графа G , а вторая подсчитывает частоту встречаемости каждого найденного подграфа-кандидата. Поиск подграфов-кандидатов начинается с графов, состоящих из одной вершины. Новые вершины и ребра добавляются итеративно к уже созданным подграфам для создания новых кандидатов. Для каждого кандидата рассчитывается частота встречаемости. Подграф-кандидат считается часто встречаемым, если его частота встречаемости не меньше некоторого граничного значения. Такой алгоритм имеет огромную вычислительную сложность. Поиск подграфов-кандидатов требует перебора множества всех подмножеств (булеана) множества вершин графа G (число операций $O(2^N)$, где N – число вершин в графе G , которое для современных КМОП схем имеет порядок 10^9). Подсчет частоты встречаемости одного подграфа-кандидата S имеет еще большую сложность ($O(2^{N^2} N^n)$, где N и n – число вершин в графах G и S).

1. Разбиение графа на подграфы

Задача разбиения графа на подграфы (graph partition) [5] состоит в представлении исходного графа $G = (V, E)$ в виде множества подмножеств вершин $\text{Sep } V = \{V_1, V_2, \dots, V_n\}$, $V_i \subseteq V$ по определенным правилам. По условию задачи поиска подсхем в КМОП схеме требуется, чтобы $\bigcup_{i=1}^n V_i = V$, т.е. все вершины исходного графа должны быть распределены по подмножествам, причем все $V_i \neq \emptyset$. В зависимости от дополнительных условий, накладываемых на блоки разбиения V_i , возникает несколько задач этого типа. Например, в случае k -блочного разбиения множество вершин делят на не более чем k отдельных блоков и так, чтобы число ребер, соединяющих блоки, было минимальным. В задаче равномерного разбиения графа стремятся получить блоки примерно одинакового размера и минимизировать связи между блоками.

Задача разбиения графа на подграфы имеет множество применений. Например, эффективная реализация параллельных программ обычно требует решения проблемы разбиения графа, в котором вершины представляют вычислительные задачи, а ребра представляют обмен данными. В этом случае k -блочное разбиение графа вычислений может использоваться для назначения задач в компьютере с k процессорами. Поскольку каждый блок разбиения содержит одинаковое количество вычислительных задач, то решение задачи обеспечивает балансировку загрузки этих k процессоров, а так как при решении задачи минимизируется мощность множества разрезающих ребер, то одновременно обеспечивается и сокращение межпроцессорного обмена данными.

В последнее время проблема разбиения графов нашла новые применения в области кластеризации для обнаружения сообществ в социальных и биологических сетях. При поиске сообществ в социальных сетях делается попытка извлечь структуру из графа социальной сети, разделив его вершины на непересекающиеся подграфы (сообщества), такие что связи между вершинами подграфа включены в множество его ребер, а число соединений между подграфами мало.

Задача k -блочного разбиения графа часто решается рекурсивным делением его множества вершин пополам [5, 6]. То есть сначала получаем 2-блочное разбиение множества V , а затем разбиваем каждую из полученных частей пополам и так далее. После $\log k$ операций деления граф G разбивается на k частей. Таким образом, задача k -блочного разбиения графа сводится к последовательности 2-блочных разбиений. Эта схема широко используется из-за своей простоты, хотя она не всегда приводит к оптимальному разбиению.

Задача разбиения на подграфы может быть расширена на гиперграфы, где ребро может соединять более двух вершин [7]. Гиперребро не разрезается, если все вершины находятся в одном блоке, и разрезается ровно один раз в противном случае, независимо от того, сколько вершин находится с каждой стороны. Эта постановка задачи возникает в задаче поиска подсхем в схеме из транзисторов, возникающей в процессе проектирования СБИС.

Еще одно важное применение рекурсивного деления пополам заключается в том, чтобы найти порядок заполнения при факторизации разреженных матриц. Этот тип алгоритмов обычно называют алгоритмами упорядочения вершин путем вложенных диссекций. Вложенная диссекция рекурсивно разбивает граф на почти равные половины, пока не будет получено требуемое количество разделов. Достижимое при этом упорядочение вершин обеспечивает ускорение умножения разреженной матрицы на вектор.

На практике для решения задачи k -блочного разбиения графа используются эвристические многоуровневые алгоритмы. Эти алгоритмы состоят из трех фаз. На фазе огрубления граф сжимается, чтобы получить иерархию графов меньших размеров. В фазе разбиения применяется алгоритм 2-блочного разбиения к наименьшему графу, построенному на первой фазе. На фазе восстановления это разбиение проецируется на граф меньшего уровня иерархии, с тем чтобы получить другое разбиение графа этого уровня. На каждом уровне используется метод поиска локального минимума для улучшения разбиения, полученного на предыдущем этапе. Формально алгоритм многоуровневой бисекции графа описывается следующим образом.

Для заданного графа $G_0 = (V_0, E_0)$ алгоритм многоуровневой бисекции состоит из следующих трех этапов. Этап огрубления преобразует граф G_0 в последовательность меньших графов G_1, G_2, \dots, G_m , таких что $|V_0| > |V_1| > |V_2| > \dots > |V_m|$. На этапе разбиения производится 2-блочное разбиение R_k графа $G_k = (V_k, E_k)$, которое разбивает V_k на две части, каждая из которых содержит половину вершин графа G_k . Фаза восстановления проецирует разбиение R_k графа G_k обратно в G_0 , проходя промежуточные разбиения $R_{k-1}, R_{k-2}, \dots, R_1, R_0$.

Лучшие программы многоуровневой бисекции позволяют получать k -блочные разбиения графов с сотнями тысяч вершин за приемлемое время на персональном компьютере, когда k не превышает сотни.

Следует отметить, что описанные алгоритмы k -блочного разбиения графа не позволяют получить решение (за приемлемое время) задачи поиска часто встречающихся подграфов для графов с большим числом вершин (миллион и более). Они могут служить только разумным упрощением операции поиска подграфов-кандидатов.

2. Обзор алгоритмов поиска часто встречающихся подграфов

Целью подобных алгоритмов является поиск полного множества часто встречающихся подграфов с использованием некоторой меры частоты встречаемости. Множество часто встречающихся подграфов составляют подграфы, у которых мера частоты встречаемости (support) больше заданной границы. Для решения задачи поиска полного множества часто встречающихся подграфов используются эвристические алгоритмы, эффективная работа которых зависит от специфических характеристик исходного графа, позволяющих, например, отличить эти графы от случайных графов произвольного вида. Обычно предполагается, что в исследуемом графе большинство вершин имеет небольшую степень и небольшой процент вершин имеет большое количество соединений. Это свойство выражается формально степенным законом распределения (power law distribution) числа вершин в зависимости от их степени.

Задача поиска часто встречающихся подграфов стала популярной областью исследований в последнее десятилетие, и к настоящему времени библиография этой задачи насчитывает сотни публикаций, однако литература на русском языке по данной проблеме практически отсутствует. Было предложено много алгоритмов поиска часто встречающихся подграфов. Последние из предложенных алгоритмов превосходят ранние разработки с точки зрения требований к памяти и быстродействию на нескольких порядков. Но ни один из алгоритмов не позволяет полностью решить проблему огромной сложности вычислений. Предлагаемые методы сокращения сложности вычислений за счет использования эвристик, не гарантирующих нахождения полного множества часто встречающихся подграфов, по-прежнему позволяют решать задачи большой размерности за экспоненциальное время.

Алгоритмы поиска часто встречающихся подграфов строятся по аналогии с подходами, используемыми в хорошо известной в задаче Data mining поиска часто встречающегося множества атрибутов (frequent item set mining). В этой задаче разыскиваются множества элементов, встречающиеся в достаточно большом количестве транзакций данной базы данных. Для задачи предложены алгоритмы, трудоемкость которых меньше полиномиальной, хотя множество всех решений составляет булеан множества элементов базы данных. Трудоемкость поиска часто встречающихся графов имеет более высокий порядок из-за необходимости решать задачу изоморфизма подграфа. К алгоритму перебора подграфов-кандидатов предъявляется требование по возможности избегать генераций кандидатов, которые нечасто встречаются или изоморфны ранее найденным, с тем чтобы как можно реже выполнять тест проверки изоморфизма подграфов.

Если вершины и ребра графа имеют уникальные метки, то граф можно задать множеством меток ребер, а затем использовать существующие алгоритмы поиска часто встречающегося множества атрибутов, чтобы найти все часто встречающиеся подграфы. Так как отображение вершин и ребер на метки не однозначно, то это сведение не всегда возможно, однако переход к маркированным (раскрашенным) графам позволяет сделать тест изоморфизма подграфа более эффективным. Кроме того, для генерации часто встречающегося множества атрибутов можно использовать подход frequent item set, основанный на «аргіогі принципе», который для задачи поиска часто встречающихся подграфов формулируется следующим образом [9]: все графы, содержащие нечасто встречающийся подграф, тоже не относятся к часто встречающимся.

Маркированный (раскрашенный) граф $G = (V, E, L, l)$, представляется множеством V вершин, множеством дуг $E \subseteq V \times V$ и функцией $l: V \cup E \rightarrow L$ отображения вершин и дуг в множество меток L . Граф $S = (V_S, E_S, L_S, l)$ является подграфом графа $G = (V, E, L, l)$, если $V_S \subseteq V$, $E_S \subseteq E$ и $L_S(u) = L(u)$ для всех $u \in V_S \cup E_S$.

Отношение изоморфизма подграфов между графами g_1 и G является отношением изоморфизма между g_1 и подграфом S графа G . Граф g_1 называется шаблоном, а S – вхождением шаблона g_1 в G .

Для подграфа G_s и графа G два вхождения G_s в G называются одинаковыми, если они используют в G один и тот же набор ребер. Два вхождения G_s в G называются непересекающимися по вершинам, если они не имеют общих вершин в G . Два вхождения G_s в G называются непересекающимися по ребрам, если они не имеют общих ребер в G .

В подходе на основе «аргіогі принципа» перед генерацией подграфов-кандидатов размера $k + 1$ необходимо иметь все кандидаты размера k , где размер графа определяется числом его вершин [8]. Чтобы получить следующий подграф-кандидат, два подграфа размера k объединяются вместе, формируя граф размера $k + 1$. Чтобы выбрать два кандидата для объединения, нужно, чтобы они имели общий подграф (тест на изоморфизм подграфов). Сгенерированный граф может быть изоморфным кандидату, сгенерированному ранее (тест на изоморфизм графов), в этом случае он отбрасывается. Сгенерированный граф может быть нечасто встречающимся (массовый тест на изоморфизм подграфов [9]), в этом случае он также отбрасывается. При подсчете числа вхождений кандидата нужно учитывать, что вхождения могут перекрываться. Если это произойдет, необходимо указать, какие вхождения должны быть приняты во внимание, чтобы «аргіогі принцип» выполнялся.

В методе grow-and-store [10], позволяющем уменьшить число проверок на изоморфизм, выполняются следующие шаги.

1. Находятся вершины, у которых число вхождений больше заданной границы, и сохраняются все найденные вхождения в списке вхождений.

2. Каждое вхождение из списка расширяется на одну вершину, образуя подграф-кандидат, оценивается частота его вхождения и, если он часто встречающийся, сохраняются все его вхождения. Это действие выполняется до тех пор, пока находятся новые часто встречающиеся подграфы.

Оценка частоты встречаемости подграфа здесь не требует теста на изоморфизм. Основным «узким» местом этого подхода является создание списка для хранения всех вхождений каждого из часто встречающихся подграфов. Этот список оказывается неприемлемо большим для вычисления и хранения.

Генерирование кандидатов, исходя из часто встречающихся подграфов меньшего размера, позволяет радикально уменьшить пространство поиска часто встречающихся подграфов заданного графа. Однако проблемой остается тест на включение кандидата в множество часто встречающихся подграфов. Тривиальным решением этой проблемы является выполнение теста кандидата на изоморфизм с каждым графом множества часто встречающихся подграфов. Эта проверка выполняется более экономным способом, если каждый граф из множества часто встречающихся подграфов канонизирован [10] и хранится в таблице по хеш-адресу, вычисленному по канонической маркировке подграфа. Когда в этой таблице по хеш-адресу, вычисленному по канонической маркировке подграфа-кандидата, хранится подграф, то он изоморфен тестируемому подграфу. Наибольший выигрыш получается, если кандидаты генерируются в канонической маркировке. Канонические маркировки графов, включенных в решение, лексикографически упорядочиваются, и построение подграфа-кандидата заключается в нахождении маркировки, которая больше всех уже построенных маркировок.

3. Цветовое кодирование подграфов

Другой класс алгоритмов подсчета частоты встречаемости подграфов основан на мощной методике их цветового кодирования (color coding) [11]. Исходными данными для метода служит шаблон и раскрашенный граф $G = (V, E, L, l)$, представляемый множествами вершин V , дуг E и функций l отображения вершин и дуг в множество красок L . Идея цветового кодирования заключается в случайной раскраске графа, соответствующего шаблону, с использованием k цветов (через k обозначается размер шаблона) из множества L . Далее в исходном раскрашенном графе $G = (V, E, L, l)$ подсчитывается количество «красочных» вхождений шаблона, т.е. подграфов, в которых все вершины имеют разные цвета. Затем это количество масштабируется по вероятности того, что вхождение шаблона является красочным. Среднее из масштабированных оценок частоты вхождений шаблона после нескольких итераций с различными случайными раскрасками исходного графа дает частоту вхождений шаблона. Метод цветового кодирования позволяет приблизительно подсчитать частоту вхождений некоторого заданного шаблона в большой граф. Главным недостатком алгоритмов цветового кодирования с вычислительной точки зрения является рост их потребности в памяти, необходимой для представления результатов, пропорционально росту размера шаблона.

Теоретически самый быстрый алгоритм подсчета вхождений k -вершинного подграфа в n -вершинный граф выполняется за время $n^{\omega k/3}$, где $O(n^{\omega})$ – временная сложность матричного умножения (в настоящее время $\omega \approx 2,38$). Это, конечно, лучше временной сложности $O(2^{n^2} n^k)$ тривиального алгоритма, но все равно чрезмерно велико даже для шаблонов с размером k порядка нескольких десятков.

4. Поиск часто встречающихся подсхем в КМОП схеме из транзисторов

Топология двудольных графов, которые моделируют схемы из транзисторов, имеет некоторые специфические свойства. Например, схемы транзисторного уровня, выполненные по КМОП технологии, имеют всего два типа элементов: n -МОП и p -МОП транзисторы. Каждый из них имеет по четыре вывода, соответствующие стоку, затвору, истоку и подложке (см. рис. 1). Выводы транзисторов и порты схемы (выводы всей электрической схемы – входные и выходные полюсы) соответствуют вершинам первой доли двудольного графа. Соответственно, каждая из вершин первой доли может иметь один из восьми возможных цветов, если она соответствует выводу транзистора n -МОП и p -МОП типа. Вершины, соответствующие разным портам схемы, раскрашены в разные цвета (но не из числа упомянутых восьми цветов).

Вторая особенность схемы из транзисторов заключается в том, что каждая вершина, принадлежащая второй доле и соответствующая цепи транзисторной схемы, определяет свою компоненту связности графа. Такая вершина может характеризоваться значением ее степени. Большинство

вершин второй доли имеет небольшие степени, но выделяется несколько вершин с большими степенями; такие вершины соответствуют, например, цепям питания, земли, тактирования. Для того чтобы метод Graph Mining работал эффективно, необходимо, чтобы распределение числа вершин в зависимости от их степеней в графе выражалось степенным законом (power law distribution). Можно считать, что это условие выполняется для двудольных графов, которые представляют схемы из транзисторов.

При распознавании подсхемы внутри схемы уровня транзисторов выделяются вхождения в последнюю некоторого шаблона (графа, задающего основу подсхемы). Подсхема строится по шаблону включением в нее вершин, соответствующих выводам, упоминаемым в шаблоне, и ребер, соединяющих эти вершины с вершинами, соответствующими цепям шаблона. Для формирования подсхемы цепи шаблона должны быть разбиты на два непересекающихся подмножества: внутренние (локальные) цепи и внешние цепи. В графе подсхемы внешние цепи шаблона соединены с выводами подсхемы. При построении шаблонов разбиением графа на подграфы внешними цепями являются цепи, которые входят в разрез экземпляра шаблона. Для часто встречающихся подграфов внутренние цепи локальны во всех экземплярах подграфа, остальные цепи являются внешними цепями шаблона подсхемы.

Набор шаблонов для заданной КМОП схемы из транзисторов может строиться разбиением графа схемы на подграфы на основе классов изоморфности этих подграфов. Но так как критерием разбиения служит минимизация некоторой оценки разреза (числа межсоединений подграфов), то этот метод получает подграфы с числом локальных межсоединений, бóльшим числа внешних межсоединений. В КМОП схемах из транзисторов такие подграфы получаются только тогда, когда число транзисторов в подграфе составляет сотни и тысячи [12], в то время как схемы типовых логических элементов содержат десятки транзисторов и число локальных соединений в них меньше числа внешних соединений. Например, схема статического КМОП инвертора локальных соединений вообще не содержит (см. рис. 1). Таким образом, разбиением графа на подграфы (graph partition) шаблоны типовых логических элементов получить практически невозможно.

Для нахождения набора шаблонов подсхем путем поиска часто встречающихся подграфов (frequent subgraph mining – FSM) могут быть использованы известные алгоритмы решения задачи FSM. Но в силу чрезвычайно большой трудоемкости выполнения операции оценки частоты встречаемости подграфа применение известных практических алгоритмов FSM ограничивается поиском подграфов небольшого размера (меньше 10 вершин) – графлетов [13]. Для графов с миллионами вершин удается подсчитать оценку частоты встречаемости графлетов еще меньшего размера [14]. Графлетом является двудольный граф, задающий схему простейшего логического элемента – статического КМОП инвертора, который содержит 8 вершин в доле выводов элементов (см. рис. 1, 2) и четыре вершины в доле цепей. Для нахождения более сложных логических элементов быстродействия известных методов FSM уже недостаточно.

Заключение

Типовыми шаблонами, обнаружение которых является задачей Graph Mining, служат часто встречающиеся подграфы и кластеры – подграфы, в которых число внутренних ребер больше числа внешних ребер. Известные практические алгоритмы FSM позволяют определить частоту встречаемости графлетов, сложность которых меньше сложности простейшего логического КМОП вентиля – инвертора. Методы кластеризации графов (graph partitioning methods) выделяют шаблоны со свойством, которым КМОП вентили не обладают. Подграфы, представляющие подсхемы КМОП вентиля, имеют больше внешних ребер, чем внутренних.

Проведенное исследование известных методов Graph Mining показывает, что, несмотря на привлекательность подхода (независимость от технологии построения элементов КМОП схем), известные методы Graph Mining не позволяют построить декомпилятор КМОП схем из транзисторов.

ЛИТЕРАТУРА

1. Logic Gate Recognition in Guardian LVS // Silvano. URL: https://www.silvaco.com/content/appNotes/iccad/2-003_LogicGates.pdf (accessed: 04.01.2018).
2. Hunt V.D. Reengineering: Leveraging the Power of Integrated Product Development. Wiley, 1993. 283 p.
3. Baker R.J. CMOS Circuit Design, Layout, and Simulation. 3rd ed. Wiley-IEEE Press, 2010. 1214 p.
4. Krasilnikova L.V., Pottosin Yu.V. Partition of a transistor circuit into library modules from a given library // Proc. of the Second Intern. Conf. on Computer-Aided Design of Discrete Devices (CAD DD'97), Minsk, Republic of Belarus, November 12–14, 1997. Minsk : Natl. Acad. Sci. Belarus Inst. Eng. Cybern., 1997. V. 1. P. 94–97.
5. Karypis G., Kumar V. Multilevel Graph Partitioning Schemes // Proc. 24th Intern. Conf. Par. Proc., III. CRC Press, 1995. P. 113–122.
7. Bauer M.A., Sarrafzadeh M., Somezi F. Fundamental CAD algorithms // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2000. V. 19 (12). P. 1449–1475.
6. Schlag S., Henne V., Heuer T., Meyerhenke H., Sanders P., Schulz C. k-way Hypergraph Partitioning via n-Level Recursive Bisection // Proc. 18th Workshop on Algorithm Engineering and Experiments (ALENEX), 2016. P. 53–67.
8. Inokuchi A.; Washio T.; Motoda H. An apriori-based algorithm for mining frequent substructures from graph data // European Symposium Principle of Data Mining and Knowledge Discovery (PKDD'00), 2000. P. 13–23.
9. Sun Z., Wang H., Shao B., Li J. Efficient subgraph matching on billion node graphs // Proc. VLDB Endow. 2012. V. 5, No. 9. P. 788–799.
10. Kuramochi M.; Karypis G. Frequent subgraph discovery // Proc. Int. Conference on Data Mining (ICDM'01). San Jose, CA, 2001. P. 313–320.
11. Slota G.M., Madduri K., Fast approximate subgraph counting and enumeration // Proc. 42nd Int. Conference on Parallel Processing (ICPP). 2013. P. 210–219.
12. Trukhachev A., Ivanova N. Extracting of High-Level Structural Representation from VLSI Circuit Description Using Tangled Logic Structures // Biologically Inspired Cognitive Architectures (BICA) for Young Scientists / A.V. Samsonovich, V.V. Klimov (ed.). Springer Int. Publishing, 2018. P. 318–323.
13. Bressan M., Chierichetti F., Kumar R., Leucci S., Panconesi A. Counting Graphlets: Space vs Time // Proc. of the 10th ACM Int. Conference on Web Search and Data Mining (WSDM 2017). 2017. P. 557–566.
14. Pinar A., Seshadhri C., Vishal V. ESCAPE: Efficiently Counting All 5-Vertex Subgraphs // Proc. of the 26th Int. Conference on World Wide Web (WWW '17). Int. World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland. 2017. P. 1431–1440.

Поступила в редакцию 22 мая 2018 г.

Cheremisinov D.I., Cheremisinova L.D. (2019) SUBCIRCUITS DISCOVERY IN TRANSISTOR LEVEL CMOS CIRCUITS USING GRAPH MINING. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie vychislitel'naja tehnika i informatika* [Tomsk State University Journal of Control and Computer Science]. 48. pp. 74–82

DOI: 10.17223/19988605/48/9

In recent years, there has been an increased interest in the development of data mining algorithms that work on graphs. Such graphs occur naturally in a number of different areas, such as network attack detection, semantic web, behavioral modeling, re-designing of VLSI, analysis of social networks and classification of chemical compounds. The problem of Graph Mining is to discover typical patterns of graph data. Common feature of such patterns are frequent subgraphs. Graph Mining is one of the arms of data mining in which voluminous complex data are represented in the form of graphs and mining is done to infer knowledge from them. Frequent subgraph mining (FSM) is a sub section of graph mining domain which is extensively used for graph classification, building indices and graph clustering purposes. Identification of frequently occurring graphs / subgraphs in a database or in one large graph is a method that can be used for motif detection, social network monitoring, fraud detection, etc.

Electrical circuits consist of elements that are connected to each other by "wires", and the natural formal model of the description of the scheme is a colored undirected bipartite graph. One part consists of the elements' terminals and ports of the circuit, and the other – the connections (nets) between the terminals, i.e., networks (nets) are the "wires". The transformation of the hierarchical circuit of an electronic device into a circuit consisting solely of primitive elements is naturally called a compilation. The reverse process, in the result of which a hierarchical circuit is built up from a flat circuit, is decompilation. To build a hierarchical structural description, you need to select in the circuit as a separate component a set of interconnected transistors, i.e. find subcircuits in the original circuit. After replacing the subcircuits with elements, the description of the circuit becomes two-level. Modern digital circuits contain up to a billion primitive elements at the transistor level. Assuming that subcircuits are frequent subgraphs, you can try to build a decompiler using the Graph Mining methods.

The task FSM has become a popular area of research in the last decade, and so far, the bibliography of this task has hundreds of publications. The complexity of FSM is enormous because of the need to solve many times the problem of subgraph isomorphism. The naive search algorithm for FSM consists of two operations. The first operation searches for all candidate subgraphs for a given

graph G , and the second one calculates the frequency of occurrence of each candidate subgraph. Due to the extremely large computational complexity of the second operation, the well-known practical FSM algorithms are limited to the search for small subgraphs (less than 10 vertices) – graphlets.

Our study of the known Graph Mining methods shows that, despite the attractiveness of the approach (independence from the style of primitive elements of circuits), the known methods of Graph Mining do not allow to build a decompiler of transistor circuits.

Keywords: clustering of a graph; subgraph counting; subgraph isomorphism; graph color coding; transistor level CMOS circuits.

CHEREMISINOV Dmitry Ivanovich (Candidate of Technical Sciences, Associate Professor, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus, Minsk, Belarus)

E-mail: cher@newman.bas-net.by

CHEREMISINOVA Liudmila Dmitrievna (Doctor of Technical Sciences, Professor, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus, Minsk, Belarus)

E-mail: cld@newman.bas-net.by

REFERENCES

1. Silvaco.com. (n.d.) *Logic Gate Recognition in Guardian LVS – Silvaco*. [Online] Available from: https://www.silvaco.com/content/appNotes/iccad/2-003_LogicGates.pdf (Accessed: 4th January 2018)
2. Hunt, V.D. (1993) *Reengineering: Leveraging the Power of Integrated Product Development*. Wiley.
3. Baker, R.J. (2010) *CMOS Circuit Design, Layout, and Simulation*. 3rd ed. Wiley-IEEE Press.
4. Krasilnikova, L.V. & Pottosin, Yu.V. (1997) Partition of a transistor circuit into library modules from a given library. *Proc. of the Second International Conference on Computer-Aided Design of Discrete Devices (CAD DD'97)*. Minsk, Republic of Belarus. November 12–14, 1997. Vol. 1. pp. 94–97.
5. Karypis, G. & Kumar, V. (1995) Multilevel Graph Partitioning Schemes. *Proc. 24th Int. Conf. Par. Proc.*, III – CRC Press. pp. 113–122.
6. Schlag, S., Henne, V., Heuer, T., Meyerhenke, H., Sanders, P. & Schulz, C. (2016) K-way hypergraph partitioning via n-level recursive bisection. *Proc. 18th Workshop on Algorithm Engineering and Experiments (ALENEX)*. pp. 53–67. DOI: 10.1137/1.9781611974317.5
7. Bauer, M.A., Sarrafzadeh, M. & Somezi, F. (2000) Fundamental CAD algorithms. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 19(12). pp. 1449–1475.
8. Inokuchi, A., Washio, T. & Motoda, H. (2000) An apriori-based algorithm for mining frequent substructures from graph data. *European Symposium Principle of Data Mining and Knowledge Discovery (PKDD'00)*. pp. 13–23. DOI: 10.1007/3-540-45372-5_2
9. Sun, Z., Wang, H., Shao, B. & Li, J. (2012) Efficient subgraph matching on billion node graphs. *Proc. VLDB Endow.* 5(9). pp. 788–799. DOI: 10.14778/2311906.2311907
10. Kuramochi, M. & Karypis, G. (2001) Frequent subgraph discovery. *Proc. Int. Conference on Data Mining (ICDM'01)*. San Jose, CA. pp. 313–320. DOI: 10.1109/ICDM.2001.989534
11. Slota, G.M. & Madduri, K. (2013) Fast approximate subgraph counting and enumeration. *Proc. 42nd Int. Conference on Parallel Processing (ICPP)*. pp. 210–219. DOI: 10.1109/ICPP.2013.30
12. Trukhachev, A. & Ivanova, N. (2018) Extracting of High-Level Structural Representation from VLSI Circuit Description Using Tangled Logic Structures. In: Samsonovich, A.V. & Klimov, V.V. (eds) *Biologically Inspired Cognitive Architectures (BICA) for Young Scientists*. Springer Int. pp. 318–323.
13. Bressan, M., Chierichetti, F., Kumar, R., Leucci, S. & Panconesi, A. (2017) Counting Graphlets: Space vs Time. *Proc. of the 10th ACM Int. Conference on Web Search and Data Mining (WSDM 2017)*. pp. 557–566. DOI: 10.1145/3018661.3018732
14. Pinar, A., Seshadhri, C. & Vishal, V. (2017) ESCAPE: Efficiently Counting All 5-Vertex Subgraphs. *Proc. of the 26th International Conference on World Wide Web (WWW '17)*. Int. World Wide Web Conferences Steering Committee. Republic and Canton of Geneva, Switzerland. pp. 1431–1440. DOI: 10.1145/3038912.3052597

ДИСКРЕТНЫЕ ФУНКЦИИ И АВТОМАТЫ

УДК 519.713

DOI: 10.17223/19988605/48/10

Е.В. Широкова, С.А. Прокопенко, Н.В. Шабалдина**К ПОСТРОЕНИЮ ПАРАЛЛЕЛЬНОЙ КОМПОЗИЦИИ
РАСШИРЕННЫХ АВТОМАТОВ***Работа выполнена при поддержке Российского научного фонда, проект № 16-49-03012.*

Рассматривается задача построения параллельной композиции двух взаимодействующих компонент, к одной из которых нет внешнего доступа. Поведение каждой компоненты описывается расширенным автоматом. Исследуются условия, при которых поведение взаимодействующих компонент можно описать расширенным автоматом без перехода к эквивалентным конечным автоматам.

Ключевые слова: расширенный автомат; конечный автомат; параллельная композиция расширенных автоматов.

Сложные системы, например веб-сервисы, обычно являются многоэлементными, т.е. представляются в виде композиции взаимодействующих компонент. Адекватной моделью, позволяющей описать поведение компонент веб-сервиса, является расширенный автомат [1]. Предполагается, что компоненты взаимодействуют в режиме диалога, т.е. в каждый момент времени активной является только одна компонента, а также соблюдается условие так называемой «медленной внешней среды», при котором очередной входной символ поступает на систему только после того, как она произвела внешний выходной символ в ответ на предыдущий внешний входной символ.

Для описания совместной работы взаимодействующих компонент обычно используется операция композиции расширенных автоматов, моделирующая поведение компонент. Однако задача построения композиции расширенных автоматов недостаточно изучена. Как правило, в случае, если области определения контекстных переменных и входных / выходных параметров расширенного автомата конечны, переходят к решению задачи композиции расширенных автоматов путем моделирования поведения последних. В этом случае поведение каждой компоненты описывается эквивалентным конечным автоматом [2], задача композиции которых хорошо изучена [3]. Однако если области определения контекстных переменных и входных / выходных параметров бесконечны либо слишком велики, то по заданному расширенному автомату нет возможности построить эквивалентный конечный автомат. В этом случае можно произвести упрощение исходного расширенного автомата, построив его срез. При построении срезов сохраняются некоторые нужные нам свойства расширенных автоматов.

Существуют различные виды срезов, например так называемый $SliceR(M)$ [4] – срез, который сохраняет свойства достижимости исходного расширенного автомата. Для построения такого среза из исходного расширенного автомата удаляются все контекстные переменные, от которых не зависят предикаты переходов автомата. Уменьшение числа контекстных переменных дает возможность строить более простые конечные автоматы при моделировании поведения расширенного автомата на параметризованных входных последовательностях.

Срез расширенного автомата $SliceFSM(M)$ (FSM, или конечноавтоматный срез) [4] строится для решения проблемы различимости состояний в расширенном автомате и не содержит контекстных переменных и выходных параметров. Для построения такого среза необходимо удалить переходы,

на которых предикат зависит от контекстных переменных. Необходимо отметить, что в общем случае FSM-срез не является конечным автоматом, поскольку некоторые переходы в нем могут зависеть от предикатов, которые, в свою очередь, зависят от входных параметров. Таким образом, такой срез является расширенным автоматом с пустым множеством контекстных переменных и выходных параметров.

Также для упрощения расширенного автомата можно построить так называемый l -эквивалент [5] путем моделирования поведения расширенного автомата на входных последовательностях длины не больше заданного числа l . Построенный l -эквивалент является конечным (в общем случае частичным) автоматом.

Методы построения композиции конечных автоматов хорошо изучены [3, 6]. Поэтому, построив для каждой из компонент конечно-автоматную абстракцию одним из вышеописанных способов, мы можем найти совместное поведение компонент в виде конечного автомата. Однако вопрос, каким образом вернуться от конечного автомата композиции к исходной модели, т.е. к расширенному автомату, остается открытым. Это, в частности, связано с тем, что непросто установить обратную связь между переходами в конечном автомате (например, l -эквиваленте) и соответствующими им переходами в расширенном автомате. В связи с этим хотелось бы рассмотреть возможность описания взаимодействия компонент композиции без перехода к эквивалентным конечным автоматам.

В работе [7] приводится формальное описание композиции расширенных автоматов в алфавитах I_1, U, O_2 , которая является частным случаем композиции, представленной на рис. 1. Также в [7] сформулированы ограничения, при которых такое описание возможно.

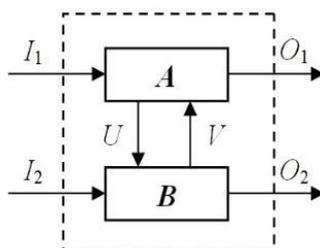


Рис. 1. Структура композиции
Fig. 1. Structure of the composition

Аналогичным образом хотелось бы построить расширенный автомат для параллельной композиции в алфавитах I_1, O_1, U, V и сформулировать условия существования такой композиции. В отличие от предыдущего вида композиции, где компоненты следуют одна за другой и отсутствует внутренний диалог между компонентами, в данном случае такой диалог возможен и компоненты взаимодействуют в режиме «медленной внешней среды» [3]. Такой режим работы композиции хорошо подходит для описания взаимодействия компонент веб-сервиса: клиентского и серверного приложений, так как взаимодействие между клиентским приложением и сервером осуществляется через последовательный обмен запросами и ответами. При этом доступ к серверам, предоставляемым сторонними разработчиками, может быть получен только посредством запросов от клиентского приложения, поэтому сервер рассматривается как встроенная компонента (не имеющая внешних входных и выходных символов).

1. Основные понятия и определения

Конечным автоматом [3] называется пятерка $S = (S, I, O, T_S, s_0)$, где S – непустое конечное множество состояний с выделенным начальным состоянием s_0 , I – непустое множество входных символов, называемое входным алфавитом, O – непустое множество выходных символов, называемое выходным алфавитом, $T_S \in I \times S \times S \times O$ – отношение переходов.

Автомат S называется полностью определенным, если для каждой пары $(i, s) \in I \times S$ существует по крайней мере одна пара $(o, s') \in O \times S$, такая что $(i, s, s', o) \in T_S$. В противном случае автомат S называется частичным.

Автомат S называется детерминированным, если для любой пары $(i, s) \in I \times S$ существует не более одной пары $(o, s') \in O \times S$, такой что $(i, s, s', o) \in T_S$, в противном случае автомат называется недетерминированным.

Расширенный автомат [1] представляет собой пятерку $M = (S, I, O, V, T)$, где S – непустое конечное множество состояний автомата, I – непустое конечное множество входных символов, называемое входным алфавитом, O – непустое конечное множество выходных символов, называемое выходным алфавитом, V – конечное, возможно пустое, множество контекстных переменных, T – отношение переходов. Каждый переход t из T – это семерка (s, i, P, op, o, up, s') , где $s, s' \in S$ являются начальным и конечным состояниями перехода; $i \in I$ есть входной символ и D_{inp-i} обозначает множество векторов, компонентами которых являются значения параметров, соответствующих входному символу i (далее входные параметры); $o \in O$ – выходной символ и D_{out-o} обозначает множество векторов, компонентами которых являются значения параметров, соответствующих выходному символу o (далее выходные параметры); P, op и up – функции, определенные над входными параметрами и контекстными переменными из V :

$P: D_{inp-i} \times D_V \rightarrow \{\text{истина, ложь}\}$ – предикат, где D_V – множество контекстных векторов;

$op: D_{inp-i} \times D_V \rightarrow D_{out-o}$ – функция вычисления значений выходных параметров;

$up: D_{inp-i} \times D_V \rightarrow D_V$ – функция вычисления значений контекстных переменных.

Параметризованным входным (выходным) символом называется пара «входной символ i , вектор a из D_{inp-i} », т.е. пара (i, a) (пара «выходной символ o , вектор b из D_{out-o} », т.е. пара (o, b)).

Конфигурацией расширенного автомата M называется пара «состояние s , контекстный вектор v », т.е. (s, v) .

Расширенный автомат называется полностью определенным, если в каждом состоянии s под действием каждого входного символа существует хотя бы один переход. Расширенный автомат M называется непротиворечивым, если из каждого состояния s для любого параметризованного входного символа и каждого значения контекстного вектора v существует не более одного перехода, предикат которого принимает значение «истина». Расширенный автомат M называется детерминированным, если в каждом состоянии s существует не более одного выполнимого перехода по любому параметризованному входному символу.

Рассмотрим композицию расширенных автоматов A и B на рис. 1, в которой:

– автомат A имеет входной алфавит $I_1 \cup V$, выходной алфавит $O_1 \cup U$ и отношение переходов T_A ;

– автомат B имеет входной алфавит $I_2 \cup U$, выходной алфавит $O_2 \cup V$ и отношение переходов T_B ;

– алфавиты I_1, I_2, O_1, O_2, V и U попарно не пересекаются.

Параллельная композиция расширенных автоматов A и B имеет:

– входной алфавит $I = I_1 \cup I_2$;

– выходной алфавит $O = O_1 \cup O_2$;

– под действием входного символа $i \in I_1 \cup I_2$ композиция вырабатывает внешний выходной символ $o \in O_1 \cup O_2$ или внутренний выходной символ, который является входным символом для другой компоненты;

– следующий входной символ может быть подан на композицию только после того, как компоненты закончили внутренний диалог.

2. Параллельная композиция расширенных автоматов

В настоящей работе для построения параллельной композиции расширенных автоматов хотелось бы использовать подходы к композиции классических конечных автоматов. Существует два

распространенных подхода: на основе перехода к полуавтоматам и путем построения дерева достижимости.

В работе [8] для построения композиции двух конечных автоматов предлагается для каждого автомата построить соответствующий полуавтомат, расширить алфавиты каждой компоненты до алфавитов другой компоненты и пересечь полученные полуавтоматы. Таким образом, получаем так называемый глобальный полуавтомат, который описывает совместное поведение взаимодействующих компонент. Далее для построения параллельной композиции необходимо ограничить глобальный полуавтомат на внешние алфавиты. Для того чтобы вернуться к модели конечного автомата, полученный полуавтомат должен удовлетворять следующему свойству: язык L этого полуавтомата должен содержаться в $(IO)^*$, где $I = I_1 \cup I_2$, $O = O_1 \cup O_2$. В случае, когда данное свойство не выполняется, необходимо сначала пересечь полученный полуавтомат с полуавтоматом, соответствующим языку $(IO)^*$, а затем вернуться к модели конечного автомата.

В работе [9] для получения параллельной композиции конечных автоматов предлагается строить дерево достижимости, которое описывает совместное поведение автоматов-компонент при подаче на вход композиции внешних входных символов из множества $I_1 \cup I_2$. Автомат композиции строится по дереву достижимости путем удаления всех внутренних переходов. Таким образом, каждый переход полученного автомата композиции помечен входным символом $i \in I_1 \cup I_2$ и внешним выходным символом $o \in O_1 \cup O_2$.

Первый подход (на основе перехода к полуавтоматам) затруднительно применить напрямую к расширенным автоматам, поскольку неясно, как выполнять расщепление перехода расширенного автомата на два перехода соответствующего полуавтомата, из-за наличия предикатов и функций обновления значений выходных параметров и контекстных переменных. Поэтому мы адаптируем второй подход (на основе дерева достижимости) для композиции расширенных автоматов и формулируем условия применимости данного подхода.

Мы рассматриваем частный случай параллельной композиции расширенных автоматов, а именно построение расширенного автомата для композиции в алфавитах I_1, O_1, U, V (т.е. компонента B является встроенной).

Пусть A и B – некоторые расширенные автоматы. Если параллельную композицию данных автоматов можно построить, не выполняя перехода к эквивалентным конечным автоматам, то необходимо построить дерево достижимости, которое описывает совместное поведение расширенных автоматов-компонент при подаче на вход композиции внешнего входного символа из алфавита I_1 . Расширенный автомат композиции строится по дереву достижимости путем удаления всех внутренних переходов. Таким образом, каждый переход полученного расширенного автомата композиции помечен входным символом $i \in I_1$, значениями параметров, соответствующих входному символу i , предикатом, значениями контекстных переменных, внешним выходным символом $o \in O_1$, значениями параметров, соответствующих выходному символу o .

Пример 1. Рассмотрим пример использования модели расширенного автомата и параллельной композиции таких автоматов для описания взаимодействия клиентской и серверной частей протокола POP3. POP3 (Post Office Protocol Version 3) [10] – протокол, с помощью которого клиенты электронной почты могут получать почту с удаленных почтовых серверов по TCP соединению. Стандартный порт POP3 сервера – 110. Изначально сервер прослушивает порт 110, ожидая TCP соединения. Когда клиент желает воспользоваться сервисом POP3, он должен установить соединение с сервером. После установки соединения сервер посылает клиенту приветствие. Далее должна быть выполнена авторизация пользователя. При успешной авторизации клиент и сервер начинают обмениваться командами и ответами до тех пор, пока соединение не будет закрыто или прервано. После команды клиента QUIT и ответа от сервера +OK сервер освобождает все ресурсы, задействованные в состоянии передачи данных. Ответ сервера всегда начинается с одного из двух идентификаторов: +OK, что означает успешное выполнение команды, поступившей от клиента, или –ERR, что означает, что команда не была выполнена.

Для того чтобы при построении параллельной композиции расширенных автоматов избежать возникновения цепочек внутренних переходов с разными предикатами и функциями вычисления значений переменных, предикаты переходов расширенного автомата, описывающего поведение сервера, были перенесены в расширенный автомат композиции. После авторизации сервер должен передать клиенту число писем n в почтовом ящике, где n – контекстная переменная для всех расширенных автоматов в примере. Массив $message$ длины n является также контекстной переменной расширенных автоматов, описывающих поведение клиентского приложения и взаимодействие клиента и сервера. Изначально все элементы массива $message$ равны 1. При удалении i -го письма элемент $message(i) = 0$. При восстановлении писем $\forall i \leq n \ message(i) = 1$.

На рис. 2 представлен фрагмент расширенного автомата, описывающего поведение клиентской части в состоянии транзакции (передачи данных). Переходы данного автомата перечислены ниже:

- 1) «Запрос письма (i)» ($i \leq n$ и $message(i) \neq 0$) / RETR (i);
- 2) +OK ($weight$) octets The (i) message / «Письмо (i), ($weight$) октетов»;
- 3) «Запрос письма (i)» ($i > n$ или $message(i) = 0$) / «Такого письма нет»;
- 4) «Удалить письмо (i)» ($i \leq n$) / DELE (i);
- 5) +OK message (i) deleted / $message(i) = 0$, «Письмо (i) удалено»;
- 6) «Удалить письмо (i)» ($i > n$) / «Такого письма нет»;
- 7) «Восстановить письма» / RSET;

8) +OK maildrop has n messages (($weight$) octets) / Для всех $i \leq n \ message(i) = 1$, «Письма восстановлены, в почтовом ящике n писем, ($weight$) октетов».

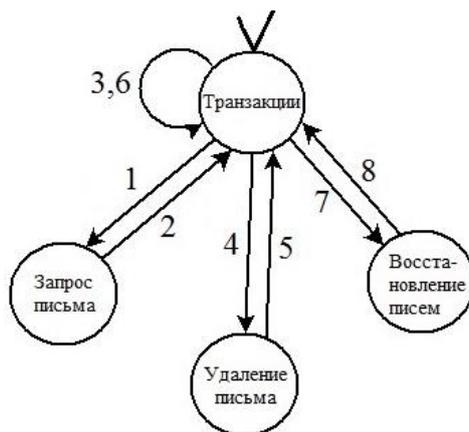


Рис. 2. Расширенный автомат, описывающий поведение клиентского приложения в состоянии обмена командами и ответами с сервером

Fig. 2. An extended automaton describing the behavior of the client application in the state of exchanging commands and responses with the server

Для простоты мы рассматриваем расширенный автомат, описывающий поведение серверной части протокола POP3 только в состоянии транзакции. В этом состоянии рассматриваются только следующие входные действия: получить i -е письмо – команда RETR (i), удалить i -е письмо – команда DELE (i), восстановить все письма, помеченные как удаленные, – команда RSET. Переходы автомата имеют вид:

- 1) RETR (i) / +OK ($weight$) octets The (i) message;
- 2) DELE (i) / +OK message (i) deleted;
- 3) RSET / +OK maildrop has n messages (($weight$) octets).

Автомат композиции клиентской и серверной частей имеет одно состояние и следующие переходы:

- 1) «Запрос письма (i)» ($i \leq n$ и $message(i) \neq 0$) / «Письмо (i), ($weight$) октетов»;
- 2) «Запрос письма (i)» ($i > n$ или $message(i) = 0$) / «Такого письма нет»;

- 3) «Удалить письмо (i)» ($i \leq n$) / $message(i) = 0$, «Письмо (i) удалено»;
- 4) «Удалить письмо (i)» ($i > n$) / «Такого письма нет»;
- 5) «Восстановить письма» / Для всех $i \leq n$ $message(i) = 1$, «Письма восстановлены, в почтовом ящике n писем, ($weight$) октетов».

Данный пример иллюстрирует, что в частном случае, когда возможен перенос предикатов компоненты в расширенный автомат композиции, можно формально описать параллельную композицию расширенных автоматов без использования моделирования.

Пример 2. Пусть A и B – расширенные автоматы (рис. 3, a , b). Предикаты переходов компоненты A зависят от выходных параметров компоненты B ; предикаты переходов компоненты B зависят от выходных параметров компоненты A .

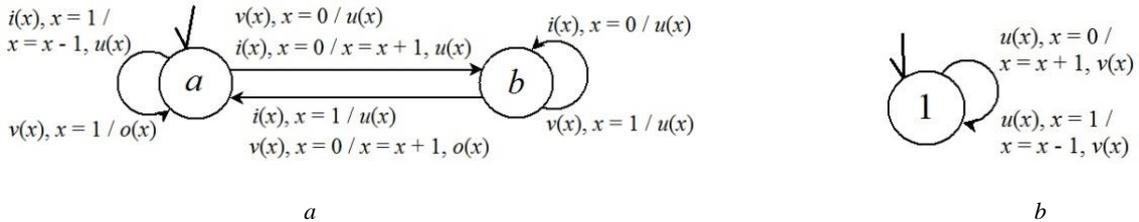


Рис. 3. Расширенный автомат A (a); Расширенный автомат B (b)
 Fig. 3. Advanced automaton A (a); Advanced machine B (b)

В данном случае, при построении параллельной композиции расширенных автоматов A и B , возникают цепочки внутренних переходов с разными предикатами и функциями вычисления значений переменной x . В настоящий момент нам не известно, каким образом описать переход между устойчивыми состояниями в композиции, сформировав по цепочке внутренних переходов предикат и функцию вычисления значений переменной x , т.е. описать параллельную композицию расширенных автоматов формально.

Построим дерево достижимости. Начальное состояние системы $a1$. При подаче входного символа $i(x)$ со значением параметра $x = 0$ система переходит из состояния $a1$ в состояние $b1$ с выдачей выходного символа $u(1)$. При подаче входного символа $i(x)$ со значением параметра $x = 1$ система остается в состоянии $a1$ с выдачей выходного символа $u(0)$ и т.д. Строим дерево достижимости до тех пор, пока в каждой ветви не получим внешний выходной символ. Дерево достижимости для расширенных автоматов A и B представлено на рис/ 4.

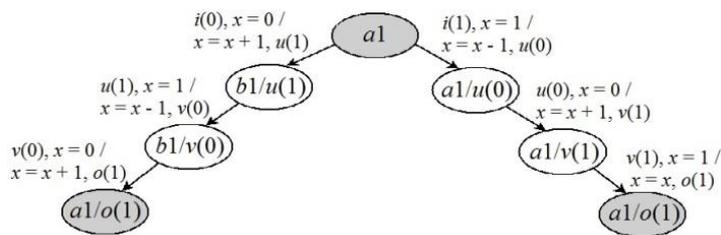


Рис. 4. Дерево достижимости
 Fig. 4. Reach tree

Полученное дерево достижимости показывает, что при построении параллельной композиции расширенных автоматов A и B возникают цепочки внутренних переходов с разными предикатами и разными функциями вычисления значения переменной x . Если мы хотим перейти к расширенному автомату композиции во внешних алфавитах, т.е. «свернуть» внутренние цепочки переходов, то возникает вопрос, как сформировать предикат перехода и функции вычисления значений параметров.

Таким образом, можно сформулировать следующие утверждения.

Утверждение 1.1. Пусть A и B – некоторые расширенные автоматы. Параллельную композицию данных автоматов можно построить, не выполняя перехода к эквивалентным конечным автоматам,

если на любой внутренней входной символ компонента A вырабатывает внешний выходной символ и предикаты компонента A не зависят от выходных параметров компоненты B .

Доказательство. Поскольку на всякий внутренний входной символ компонента A вырабатывает внешний выходной символ, то внутренний диалог состоит только из обмена одной входо-выходной парой. Так как предикат перехода автомата A не зависит от выходных параметров компоненты B , то не возникает конфликта при вычислении значения предиката перехода при таком внутреннем диалоге. Таким образом, для данного случая можно построить параллельную композицию расширенных автоматов A и B , не выполняя перехода к эквивалентным конечным автоматам.

Утверждение 1.2. Пусть A и B – некоторые расширенные автоматы. Параллельную композицию данных автоматов можно построить, не выполняя перехода к эквивалентным конечным автоматам, если во внутреннем диалоге компонент содержится не более одного перехода, помеченного предикатом.

Доказательство. Если во внутреннем диалоге автоматов A и B нет переходов, помеченных предикатами, то при построении дерева достижимости не возникает цепочек внутренних переходов с разными функциями вычисления значений параметров и контекстных переменных. Таким образом, можно построить параллельную композицию расширенных автоматов A и B , не выполняя перехода к эквивалентным конечным автоматам.

Если во внутреннем диалоге данных автоматов содержится только один переход, помеченный предикатом, то при построении дерева достижимости для автоматов A и B возникает один внутренний переход с разными функциями вычисления значений параметров и контекстных переменных. В этом случае возможен перенос предиката в переход расширенного автомата композиции, так как мы можем отследить по полученному дереву достижимости, какое именно внешнее входное воздействие с каким параметром привело к возникновению такого внутреннего перехода и выдаче соответствующего внешнего выходного символа. То есть для данного случая можно построить параллельную композицию расширенных автоматов A и B , не выполняя перехода к эквивалентным конечным автоматам.

Заключение

Таким образом, в данной работе мы рассмотрели возможность переноса методов теории конечных автоматов для построения параллельной композиции на случай взаимодействующих расширенных автоматов. Сформулированы условия, при которых данный перенос возможен, т.е. можно построить композицию расширенных автоматов без их моделирования. В дальнейшем будет рассмотрена композиция, в которой обе компоненты взаимодействуют с внешней средой.

ЛИТЕРАТУРА

1. Petrenko A., Boroday S., Groz R. Confirming Configurations in EFSM Testing // IEEE Trans. Software Eng. 2004. V. 30 (1). P. 29–42.
2. Гилл А. Введение в теорию конечных автоматов. М. : Наука, 1966. 272 с.
3. Евтушенко Н.В., Рекун М.В., Тихомирова С.В. Недетерминированные автоматы: анализ и синтез : учеб. пособие. Томск : Том. гос. ун-т, 2009. Ч. 2: Решение автоматных уравнений. 111 с.
4. Коломеец А.В. Алгоритмы синтеза проверяющих тестов для управляющих систем на основе расширенных автоматов : дис. ... канд. техн. наук. Томск, 2010. 129 с.
5. Карибский В.В., Пархоменко П.П., Согомоян Е.С., Халчев В.Ф. Основы технической диагностики. М. : Энергия, 1976. 464 с.
6. Villa T., Yevtushenko N., Mishchenko A., Brayton R.K., Petrenko A., Sangiovanni-Vincentelli A. The unknown component problem: theory and applications. Berlin : Springer, 2012. 311 p.
7. Prokopenko S. Locating a faulty component of an EFSM composition // The Proc. of ISP RAS. 2014. V. 26, is. 6. P. 47–55.
8. Castagnetti G., Piccolo M., Villa T., Yevtushenko N., Mishchenko A., Brayton R.K. Solving Parallel Equations with BALM-II. Technical Report No. UCB/EECS2012-181 / Electrical Engineering and Computer Sciences University of California at Berkeley. Berkeley, CA, 2012. URL: <http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-181.pdf> (accessed: 27.05.2018).

9. El-Fakih Kh., Trenkaev V., Spitsyna N., Yevtushenko N. FSM Based Interoperability Testing Methods // Lecture notes in Computer Science. 2004. V. 2978. P. 60–75.
10. RFC 1939 – Протокол POP3. URL: <http://www.rfc2.ru/1939.rfc> (дата обращения: 27.05.2018).

Поступила в редакцию 20 июня 2018 г.

Shirokova E.V., Prokopenko S.A., Shabaldina N.V. (2019) ON DERIVING THE PARALLEL COMPOSITION OF EXTENDED FINITE STATE MACHINES. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie vychislitel'naja tehnika i informatika* [Tomsk State University Journal of Control and Computer Science]. 48. pp. 83–91

DOI: 10.17223/19988605/48/10

In this paper we consider a problem of communicating components of telecommunication systems. The components interact not only with each other but also with an environment. The interaction is in a dialog mode and the so-called “slow environment” condition takes place. As usual, each component of the system is a software. An Extended Finite State Machine (EFSM) allows to describe the software. In order to verify and test the communicating components we need a formal description of the system. It is well known how to get such description for the case when each component is described as a classical Finite State Machine (FSM). However, it is not always possible to derive an equivalent FSM for the appropriate EFSM. The first reason is that domains of input / output parameters and context variables can be infinite. The second one is that even domains are finite we can face to so-called state explosion problem. So in this paper we discuss approaches how to describe the composition of EFSMs in case when one of components is embedded. There are two approaches to derive the composition of FSMs. One of them is based on transformation of each FSM into corresponding automaton. These automata are composed and then the composition is transformed into an FSM. Unfortunately, we can't use this approach because we don't know how to transform the EFSM into automaton. The second approach is based on derivation of reachability tree. In this case, the composition at a stable state accepts an external input, moves through intermediate states (an internal dialog between components), reaches new stable state and produces an external output. The composition has only stable states and transitions between these states are marked by external inputs and outputs. And we adopted this approach for the case of EFSM's composition. We have formulated two conditions when this approach is valid for EFSMs. If for each internal input the Context produces an external output and predicates of Context don't depend on output parameters of Embedded component, then it's possible to construct a composition of EFSMs without derivation equivalent FSMs. The second condition is that the approach is valid if there is not more than one transition with predicate in internal dialog between components.

Keywords: extended finite state machine; finite state machine; parallel composition of extended finite state machines.

SHIROKOVA Ekaterina Vladimirovna (Post-graduate Student, National Research Tomsk State University, Tomsk, Russian Federation).
E-mail: darusenkova@gmail.com

PROKOPENKO Svetlana Anatolievna (Candidate of Technical Sciences, Associate Professor, National Research Tomsk State University, Tomsk, Russian Federation).
E-mail: s.prokopenko@sibmail.com

SHABALDINA Natalia Vladimirovna (Candidate of Technical Sciences, Associate Professor, National Research Tomsk State University, Tomsk, Russian Federation).
E-mail: nataliamailbox@mail.ru

REFERENCES

1. Petrenko, A., Boroday, S. & Groz, R. (2004) Confirming Configurations in EFSM Testing. *IEEE Trans. Software Eng.* 30(1). pp. 29–42.
2. Gill, A. (1966) *Vvedenie v teoriyu konechnykh avtomatov* [Introduction to the theory of finite state machines]. Translated from English. Moscow: Nauka.
3. Evtushenko, N.V., Rekun, M.V. & Tikhomirova, S.V. (2009) *Nedeterminirovannye avtomaty: analiz i sintez* [Nondeterministic finite state machines: analysis and synthesis]. Part 2. Tomsk: Tomsk State University.
4. Kolomeets, A.V. (2010) *Algoritmy sinteza proverayushchikh testov dlya upravlyayushchikh sistem na osnove rasshirenykh avtomatov* [Diagnostic test derivation methods for telecommunication systems based on an EFSM model]. Engineering Cand. Diss. Tomsk.
5. Karibskiy, V., Parkhomenko, P., Sogomonyan, E. & Khalchev, V. (1976) *Osnovy tekhnicheskoy diagnostiki* [Basics of technical diagnostics]. Moscow: Energiya.
6. Villa, T., Yevtushenko, N., Mishenko, A., Brayton, R.K., Petrenko, A. & Sangiovanni-Vincentelli, A. (2012) *The Unknown Component Problem: Theory and Applications*. Berlin: Springer.

7. Prokopenko, S. (2014) Locating a faulty component of an EFSM composition. *The Proc. of ISP RAS*. 26(6). pp. 47–55. DOI: 10.1007/s 10009-014-0357-7
8. Castagnetti, G., Piccolo, M., Villa, T., Yevtushenko, N., Mishchenko, A. & Brayton R.K. (2018) *Solving Parallel Equations with BALM-II*. Technical Report No. UCB/EECS2012-181. Electrical Engineering and Computer Sciences University of California at Berkeley. [Online] Available from: <http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-181.pdf>. (Accessed: 27th May 2018).
9. El-Fakih, Kh., Trenkaev, V., Spitsyna, N. & Yevtushenko, N. (2004) FSM Based Interoperability Testing Methods. *Lecture Notes in Computer Science*. 2978. pp. 60–75. DOI: 10.1007/978-3-540-24704-3_5
10. RFC. (1939) *POP3 Protocol*. [Online] Available from: <http://www.rfc2.ru/1939.rfc>. (Accessed: 27th May 2018).

ПРОЕКТИРОВАНИЕ И ДИАГНОСТИКА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

УДК 004.312

DOI: 10.17223/19988605/48/11

Н.Б. Буторина, Е.Г. Пахомова

О ДЛИНАХ ПУТЕЙ САМОТЕСТИРУЕМЫХ ДЕТЕКТОРОВ, ПОСТРОЕННЫХ В БАЗЕ ПЛБ

Исследование выполнено при поддержке гранта Российского научного фонда, проект № 14-19-00218.

При проектировании самопроверяемых схем используются самотестируемые детекторы кодов, в частности детекторы равновесных кодов ((m, n) -кодов). Для представления всевозможных кодовых слов (m, n -кодов) используется специальная формула разложения. В работе сравниваются два самотестируемых детектора (m, n -кодов), построенные на многократном использовании этой формулы разложения и реализации полученной в результате формулы программируемыми логическими блоками (ПЛБ). Конкретный вид формулы определяется множеством представляемых ею кодовых слов (параметрами m, n), числом k входов в ПЛБ и способом разбиения переменных на подмножества. В работе анализируется влияние способа разбиения переменных на подмножества на максимальную длину пути в схеме детектора и на количество ПЛБ в детекторе.

Ключевые слова: самотестируемость; детектор; программируемые логические блоки; равновесный код.

В самопроверяемых схемах используются детекторы кодов. Как правило, детектор строится на той же элементной базе, что и схема. Метод построения детектора обычно состоит в подсчете веса входного кодового слова. Для этой цели используются схемы, основанные либо на пороговых элементах, либо на параллельных счетчиках [1–7]. В данной работе предлагается синтез детектора на ПЛБ. В [8] был предложен метод проектирования самопроверяемого конечного автомата. В случае исправного функционирования схемы на выходе комбинационной составляющей реализуются кодовые слова некоторого неупорядоченного кода, например равновесного. Там же предлагается метод синтеза самотестируемого детектора равновесного кода (детектора (m, n) -кодов), основанный на многократном использовании специальной формулы разложения множества кодовых слов равновесного кода и реализации полученной в результате формулы программируемыми логическими блоками в предположении, что среди программируемых блоков обязательно присутствуют блоки с двумя выходами. К реализации формулы предъявляется специальное требование: соответствующая реализации схема детектора (m, n) -кодов должна быть самотестируемой для заданного множества V неисправностей. Предлагаемый для сравнения подход [9] основан на многократном использовании той же формулы разложения, применяемой к иному, по сравнению с работой [8], разбиению подмножества переменных кодовых слов. Это разбиение ориентировано на сокращение максимальной длины путей в схеме, являющейся реализацией полученной в результате формулы, и минимизацию разброса длин путей. В предложенном в работе [9] методе синтеза самотестируемых детекторов наличие двух выходных ПЛБ не обязательно. Это значит, что данный метод позволяет строить самотестируемые детекторы, используя любые существующие на текущий момент программируемые логические матрицы (Field Programmable Gate Arrays (FPGA), производители Xilinx, Altera, Achronix, Actel, Atmel, Lattice semiconductor и др.), в то время как метод, предложенный в [8], ориентирован только на FPGA фирмы Xilinx.

Рассматривается множество V неисправностей детектора, которое включает в себя все кратные константные неисправности на входах и выходах ПЛБ. При этом в схеме детектора неисправным может быть только один ПЛБ. Предполагается, что в системе, состоящей из самопроверяемой схемы и детектора (m, n) -кодов, неисправными могут быть либо схема, либо детектор, но не оба вместе.

К самотестируемому детектору предъявляются следующие требования:

1) при появлении на выходе схемы (или на входе детектора) в некоторый момент времени t не-кодированного слова детектор должен выдать соответствующий сигнал;

2) в самом детекторе может произойти неисправность из рассматриваемого множества неисправностей V , которая должна быть обнаружима в рабочей области функционирования детектора, т.е. на множестве всех его кодовых слов. Это означает, что должен существовать кодовый набор (кодированное слово (m, n) -кода), на котором эта неисправность проявляется на выходах детектора.

Самотестируемый детектор имеет два выхода, причем комбинации значений сигналов имеют следующие интерпретации:

а) (01) или (10) означает, что входной набор является кодовым словом $((m, n)$ -кода) и детектор исправен;

б) (00) или (11) означает, что либо входной набор не является кодовым словом, либо детектор неисправен.

Детектор строится по формуле, представляющей множество всех кодовых слов (m, n) -кода, являясь реализацией этой формулы.

1. Метод разложения множества кодовых слов (m, n) -кода

Заметим, что число всевозможных кодовых слов (m, n) -кода равно C_n^m , т.е. числу сочетаний из n по m . Кодовые слова могут быть представлены дизъюнкцией элементарных конъюнкций (ДНФ) ранга n . Обозначим эту дизъюнкцию $D_n^m(X)$, где $X = \{x_1, \dots, x_n\}$ – множество переменных. Уже при $n = 10, m = 5$ ДНФ $D_{10}^5(X)$ состоит из $C_{10}^5 = 252$ конъюнкций ранга 10 и содержит 2 520 букв. Поскольку любые две конъюнкции из $D_n^m(X)$ ортогональны по крайней мере по двум переменным, то ДНФ $D_n^m(X)$ является совершенной и сокращенной одновременно и, следовательно, не может быть сокращена в результате минимизации ДНФ.

Для представления всевозможных (m, n) -кодов в [8] предложена специальная формула, включающая скобки, символы \wedge, \vee и ДНФ $D_p^q(X^r)$. Выражение $D_p^q(X^r)$ – это дизъюнкция конъюнкций, соответствующих всем (q, p) -кодированным словам, $p \leq n, q \leq p, X^r \subset X, X = \{x_1, \dots, x_n\}$. Построим формулу следующим образом.

Разделим множество X на два подмножества X^1, X^2 , где $X^1 = \{x_1, \dots, x_k\}, X^2 = \{x_{k+1}, \dots, x_n\}$.

Все множество кодовых слов равновесного кода может быть представлено формулой

$$D_n^m(X) = \sum_{i=0}^m D_k^i(X^1) D_{n-k}^{m-i}(X^2), \quad (1)$$

где символ \wedge между $D_k^i(X^1)$ и $D_{n-k}^{m-i}(X^2)$ опущен.

Назовем k основой разложения (k – число входов в ПЛБ), D_k^i, D_{n-k}^{m-i} – функциями разложения. Если $n - k > k$, то формула (1) снова используется для соответствующей функции разложения D_{n-k}^{m-i} , $i = \overline{0, m}$, и т.д.

В результате многократного применения формулы (1) получаем представление для всех кодовых слов (m, n) -кода в виде формулы, содержащей скобки, операции \vee и \wedge и $D_p^q(X^r)$. В этой формуле для любой ДНФ $D_p^q(X^r)$ выполняется условие $p \leq k$. Представления такого вида в дальнейшем будем называть *формулами А*.

Например, получим формулу A для D_8^4 , $k = 2$ при помощи формулы (1).

В соответствии с приведенным алгоритмом на первом шаге: $X^1 = \{x_1, x_2\}$, $X^2 = \{x_3, x_4, x_5, x_6, x_7, x_8\}$.

$$D_8^4(X) = D_2^0(X^1)D_6^4(X^2) \vee D_2^1(X^1)D_6^3(X^2) \vee D_2^2(X^1)D_6^2(X^2).$$

Далее, $X^{21} = \{x_3, x_4\}$, $X^{22} = \{x_5, x_6, x_7, x_8\}$. Выполнив следующий шаг разложения по формуле (1) для $D_6^2(X^2)$, $D_6^3(X^2)$, $D_6^4(X^2)$ получим

$$D_6^2(X^2) = D_2^0(X^{21})D_4^2(X^{22}) \vee D_2^1(X^{21})D_4^1(X^{22}) \vee D_2^2(X^{21})D_4^0(X^{22}),$$

$$D_6^3(X^2) = D_2^0(X^{21})D_4^3(X^{22}) \vee D_2^1(X^{21})D_4^2(X^{22}) \vee D_2^2(X^{21})D_4^1(X^{22}),$$

$$D_6^4(X^2) = D_2^0(X^{21})D_4^4(X^{22}) \vee D_2^1(X^{21})D_4^3(X^{22}) \vee D_2^2(X^{21})D_4^2(X^{22}).$$

Далее, $X^{221} = \{x_5, x_6\}$, $X^{222} = \{x_7, x_8\}$. Выполнив следующий шаг разложения по формуле (1) для $D_4^0(X^{22})$, $D_4^1(X^{22})$, $D_4^2(X^{22})$, $D_4^3(X^{22})$, $D_4^4(X^{22})$ получим

$$D_4^0(X^{22}) = D_2^0(X^{221})D_2^0(X^{222}),$$

$$D_4^1(X^{22}) = D_2^0(X^{221})D_2^1(X^{222}) \vee D_2^1(X^{221})D_2^0(X^{222}),$$

$$D_4^2(X^{22}) = D_2^0(X^{221})D_2^2(X^{222}) \vee D_2^1(X^{221})D_2^1(X^{222}) \vee D_2^2(X^{221})D_2^0(X^{222}),$$

$$D_4^3(X^{22}) = D_2^1(X^{221})D_2^2(X^{222}) \vee D_2^2(X^{221})D_2^1(X^{222}),$$

$$D_4^4(X^{22}) = D_2^2(X^{221})D_2^2(X^{222}).$$

Подставляя полученные выражения в формулу (1) получаем формулу A . Структура этой формулы может быть представлена деревом (рис. 1).

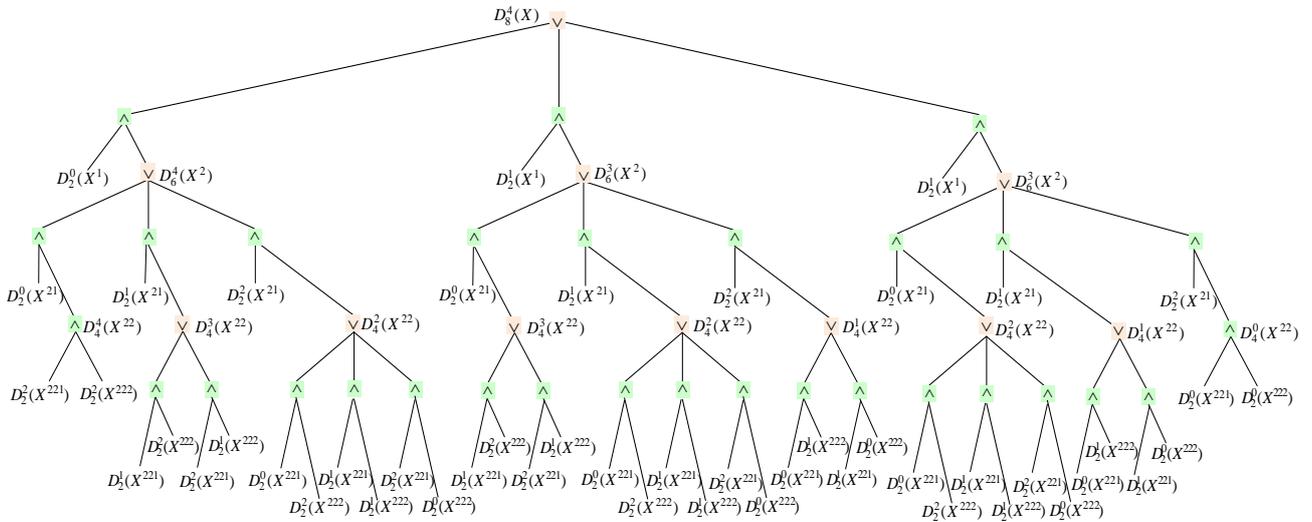


Рис. 1. Дерево разложения для D_8^4 , $k = 2$ (способ 1)

Fig. 1. Tree decomposition for D_8^4 , $k = 2$ (technique 1)

В [9] был предложен другой способ разбиения множества переменных на подмножества.

Разделим множество X на два подмножества X^1, X^2 , где $X^1 = \{x_1, \dots, x_g\}$, $X^2 = \{x_{g+1}, \dots, x_n\}$. Все множество кодовых слов равновесного кода может быть представлено формулой

$$D_n^m(X) = \sum_{i=0}^m D_g^i(X^1)D_{n-g}^{m-i}(X^2), \quad (2)$$

где символ \wedge между $D_k^i(X^1)$ и $D_{n-k}^{m-i}(X^2)$ опущен.

Здесь $D_g^i(X^1)$, $D_{n-g}^{m-i}(X^2)$ – функции разложения, основа разложения k – максимальная мощность подмножества переменных при последнем применении формулы (2).

В статье [9] предлагается следующий способ выбора g : g – наименьшее целое, большее или равное числу $n/2$ или, в другой записи, $g = \lceil n/2 \rceil$. Если $g > k$ и / или $n - g > k$, то формула (2) снова используется для каждой функции разложения $D_g^i(X^1)$, $D_{n-g}^{m-i}(X^2)$, $i = \overline{0, m}$, и т.д. В результате мы имеем формулу для всех кодовых слов (m, n) -кода, в которой для любой $D_p^q(X^r)$ выполняется условие $p \leq k$. Это значит, что во втором способе многократное разложение применяется как к первому, так и ко второму множителю формулы (2).

Например, получим формулу A для D_8^4 , $k = 2$ вторым способом.

На первом шаге $X^1 = \{x_1, x_2, x_3, x_4\}$, $X^2 = \{x_5, x_6, x_7, x_8\}$. Результат разложения представляется в виде:
 $D_8^4(X) = D_4^0(X^1)D_4^4(X^2) \vee D_4^1(X^1)D_4^3(X^2) \vee D_4^2(X^1)D_4^2(X^2) \vee D_4^3(X^1)D_4^1(X^2) \vee D_4^4(X^1)D_4^0(X^2)$.

На втором шаге имеем подмножества $X^{11} = \{x_1, x_2\}$, $X^{12} = \{x_3, x_4\}$, $X^{21} = \{x_5, x_6\}$, $X^{22} = \{x_7, x_8\}$. Применим формулу (2) еще раз ко всем функция разложения:

$$\begin{aligned} D_4^0(X^1) &= D_2^0(X^{11})D_2^0(X^{12}), \\ D_4^1(X^1) &= D_2^0(X^{11})D_2^1(X^{12}) \vee D_2^1(X^{11})D_2^0(X^{12}), \\ D_4^2(X^1) &= D_2^0(X^{11})D_2^2(X^{12}) \vee D_2^1(X^{11})D_2^1(X^{12}) \vee D_2^2(X^{11})D_2^0(X^{12}), \\ D_4^3(X^1) &= D_2^1(X^{11})D_2^2(X^{12}) \vee D_2^2(X^{11})D_2^1(X^{12}), \\ D_4^4(X^1) &= D_2^2(X^{11})D_2^2(X^{12}), \\ D_4^0(X^2) &= D_2^0(X^{21})D_2^0(X^{22}), \\ D_4^1(X^2) &= D_2^0(X^{21})D_2^1(X^{22}) \vee D_2^1(X^{21})D_2^0(X^{22}), \\ D_4^2(X^2) &= D_2^0(X^{21})D_2^2(X^{22}) \vee D_2^1(X^{21})D_2^1(X^{22}) \vee D_2^2(X^{21})D_2^0(X^{22}), \\ D_4^3(X^2) &= D_2^1(X^{21})D_2^2(X^{22}) \vee D_2^2(X^{21})D_2^1(X^{22}), \\ D_4^4(X^2) &= D_2^2(X^{21})D_2^2(X^{22}). \end{aligned}$$

Подставляя полученные выражения в формулу (2) получаем формулу A .

Можно сказать, что формула (1) есть частный случай формулы (2) при $g = k$.

Структура формулы (2) может быть представлена деревом (рис. 2).

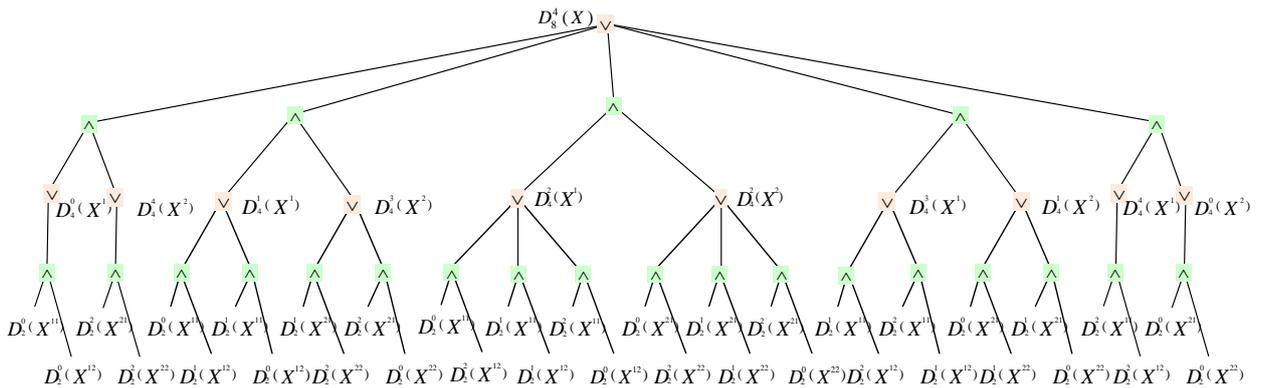


Рис. 2. Дерево разложения для D_8^4 , $k = 2$ (способ 2)

Fig. 2. Tree decomposition for D_8^4 , $k = 2$ (technique 2)

Дадим определение шага разложения.

1. При первом применении формулы (2) множество переменных X разбивается на два подмножества: X^1 и X^2 . При этом мощность одного из этих подмножеств или обоих подмножеств может оказаться больше k . Это первый шаг разложения.

2. На втором шаге разложения формула (2) применяется ко всем функциям разложения предыдущего шага, определенным на подмножестве переменных, мощность которого больше k . В результате применения формулы (2) подмножество X^1 и / или X^2 разбивается на два подмножества, мощность каждого из которых меньше мощности X^1 или X^2 соответственно, но при этом снова может быть больше k .

3. Таким образом, i -й шаг разложения – это применение формулы (2) ко всем функциям разложения предыдущего ($i - 1$)-го шага, определенным на подмножестве переменных, мощность которого больше k .

Итак, конкретный вид формулы определяется множеством представляемых ею кодовых слов (параметрами m, n), значением k и способом разбиения переменных на подмножества.

В дальнейшем будем сравнивать рассмотренные выше способы разбиения множеств переменных на два подмножества. По сути, формула (1) есть частный случай формулы (2). Действительно, в первом способе разложения $g = k$ на каждом шаге разложения, и формула (1) используется только для функции разложения D_{n-k}^{m-i} , $i = \overline{0, m}$, и т.д.

Во втором способе выбирается g , кратное k . При этом возможны следующие две ситуации.

1. Пусть n кратно k , т.е. $n = kq = k(q^1 + q^2)$, где kq^1 – ближайшее к $\lceil n/2 \rceil$ число, кратное k , $q^2 = q - q^1$. Итак, на первом шаге множество X , $|X| = n$, разбивается на два подмножества X^1 и X^2 , где $|X^1| = kq^1 = g$, $|X^2| = kq^2 = n - g$.

Если $|X^1| > k$ и (или) $|X^2| > k$, то применяем формулу (2) к соответствующему сомножителю $D_g^i(X^1)$, $D_{n-g}^{m-i}(X^2)$, выполняя разбиение множеств X^1, X^2 в соответствии с правилами пункта 1, и т.д.

2. Пусть n не кратно k . Тогда при использовании формулы (2) на первом шаге разложения предлагается выбирать g – ближайшее к $\lceil n/2 \rceil$ число, кратное k . В этом случае $n - g$ не кратно k . Итак, на первом шаге множество X , $|X| = n$, разбивается на два подмножества X^1 и X^2 , причем $|X^1| = kq^1 = g$, $|X^2| = kq^2 + r = n - g$, где $r < k$.

Если $|X^1| > k$ и (или) $|X^2| > k$, то применяем формулу (2) к соответствующему сомножителю $D_g^i(X^1)$, $D_{n-g}^{m-i}(X^2)$, выполняя разбиение множеств X^1, X^2 в соответствии с правилами пункта 2, и т.д.

Приведем пример нахождения конкретных значений g и $n - g$ при $k = 6$ для $n = 18, 21, 30$ двумя способами (рис. 3, 4).

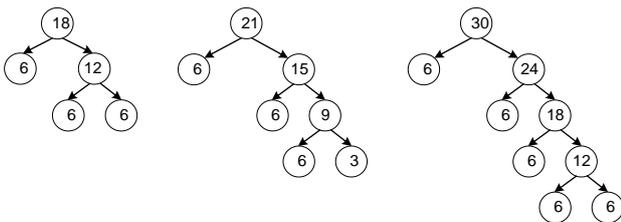


Рис.3. Деревья выбора при $k = 6$ для $n = 18, 21, 30$ (способ 1)
Fig. 3. Trees of choice with $k = 6$ for $n = 18, 21, 30$ (technique 1)

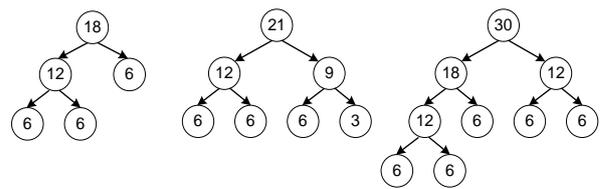


Рис.4. Деревья выбора при $k = 6$ для $n = 18, 21, 30$ (способ 2)
Fig. 4. Trees of choice with $k = 6$ for $n = 18, 21, 30$ (technique 2)

В обоих способах конечным вершинам дерева разложения сопоставляются ДНФ вида D_k^i , где $0 \leq i \leq k$.

Под синтезом детектора будем понимать реализацию формулы A композицией программируемых логических блоков, основанную на анализе дерева разложения формулы A . Подход к синтезу самотестируемого детектора для обоих способов остается одинаковым: вершины дерева разложения формулы A покрываются программируемыми логическими блоками снизу вверх [8].

Заметим, что кодовые слова, представляемые формулой A , при синтезе детектора необходимо разбивать на два подмножества: каждое из подмножеств сопоставляется одному из выходов детектора.

2. Сравнение двух детекторов

Проведем оценку детекторов по двум параметрам: количеству ПЛБ и длине пути.

Последовательность ПЛБ, такую что выход предыдущего блока является входом в последующий блок, назовем *путем* в детекторе. Входы первого блока являются входами детектора, а выход последнего блока – один из выходов детектора. *Длиной пути* назовем число ПЛБ, образующих путь. Нас будут интересовать пути максимальной длины, поскольку они влияют на быстродействие самопроверяемой системы, состоящей из самопроверяемой схемы и контролирующего ее самотестируемого детектора.

Договоримся детектор, построенный по формуле (1) называть *детектор 1*, а детектор, построенный по формуле (2) – *детектор 2*.

При $n \leq 3k$ значения g и $n - g$ для обоих способов разложения одинаковы, следовательно, одинаковы формулы A и представляющие их деревья разложения. Различия появляются при $n > 3k$.

Проведем оценку детекторов по максимальной длине пути L_{\max} . Она зависит от числа μ шагов разложения и максимальной длины пути λ в подсхемах, реализующих формулу (1) или (2) на очередном шаге разложения.

Число μ шагов разложения зависит от выбранного способа разбиения переменных на подмножества.

Максимальная длина пути λ в подсхеме, реализующей формулу на очередном шаге разложения, зависит от количества слагаемых в этой формуле. Покажем это на примере.

Применим формулу (2) к D_{14}^7 при $k = 7$ с целью получения реализующей D_{14}^7 подсхемы:

$$D_{14}^7 = D_7^0(X^1)D_7^7(X^2) \vee D_7^1(X^1)D_7^6(X^2) \vee D_7^2(X^1)D_7^5(X^2) \vee D_7^3(X^1)D_7^4(X^2) \vee D_7^4(X^1)D_7^3(X^2) \vee D_7^5(X^1)D_7^2(X^2) \vee D_7^6(X^1)D_7^1(X^2) \vee D_7^7(X^1)D_7^0(X^2). \quad (3)$$

Обозначения выходов ПЛБ приведены в табл. 1.

Таблица 1

Обозначения выходов ПЛБ, реализующих множители формулы (3)

Реализуемая функция	Обозначение выхода	Реализуемая функция	Обозначение выхода
$D_7^0(X^1)$	y_1	$D_7^0(X^2)$	y_{16}
$D_7^1(X^1)$	y_3	$D_7^1(X^2)$	y_{14}
$D_7^2(X^1)$	y_5	$D_7^2(X^2)$	y_{12}
$D_7^3(X^1)$	y_7	$D_7^3(X^2)$	y_{10}
$D_7^4(X^1)$	y_9	$D_7^4(X^2)$	y_8
$D_7^5(X^1)$	y_{11}	$D_7^5(X^2)$	y_6
$D_7^6(X^1)$	y_{13}	$D_7^6(X^2)$	y_4
$D_7^7(X^1)$	y_{15}	$D_7^7(X^2)$	y_2

Имеем 8 слагаемых, состоящих из двух множителей. Каждый множитель сопоставляется входу ПЛБ, который реализует дизъюнкцию нескольких слагаемых. Так как в данном примере количество входов ПЛБ равно 7, то для реализации всех слагаемых потребуется $\lceil 2 \cdot 8 / 7 \rceil = 3$ ПЛБ. Обозначим выходы этих ПЛБ через z_1, z_2, z_3 .

Таким образом, ПЛБ1 реализует функцию $y_1y_2 \vee y_3y_4 \vee y_6y_7$, выход этого ПЛБ обозначен через z_1 , ПЛБ2 реализует функцию $y_7y_8 \vee y_9y_{10} \vee y_{11}y_{12}$, выход этого ПЛБ обозначен через z_2 , ПЛБ3 реализует функцию $y_{13}y_{14} \vee y_{15}y_{16}$, выход этого ПЛБ обозначен через z_3 .

Пусть выходы подсхемы, реализующей формулу (3), являются выходами детектора (их два). Выходы z_1 и z_2 делаем входами ПЛБ, реализующий функцию D_2^1 . Выход этого ПЛБ является одним из выходов детектора, второй выход – z_3 . Максимальная длина пути λ_{\max} в подсхеме, реализующей формулу (3), равна 2. Эта подсхема приведена на рис. 5.

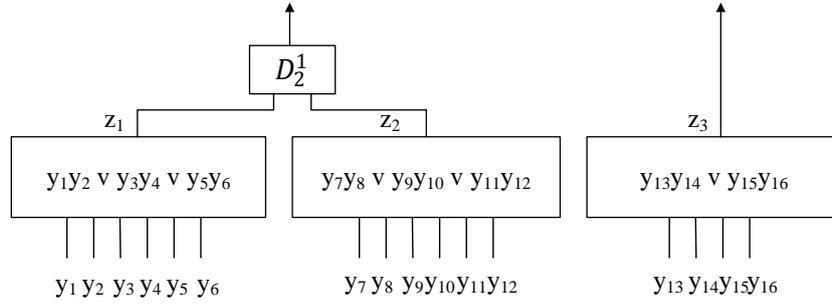


Рис. 5. Подсхема, реализующая формулу (3)
Fig. 5. A subcircuit realizing formula (3)

В общем случае, если на реализацию формулы (1) или (2) требуется два и более ПЛБ и выходы этих ПЛБ не являются выходами детектора, то максимальная длина пути λ_{\max} в подсхеме, реализующей формулу, больше 1.

Пусть L_{\max} – максимальная длина пути в схеме, реализующей детектор, λ_{\max}^i – максимальная длина пути среди всех подсхем, реализующих i -й шаг разложения с использованием формулы (1) или (2). Тогда по построению $L_{\max} = \sum_{i=1}^{\mu} \lambda_{\max}^i + 1$, т.е. L_{\max} – сумма всех максимальных длин путей на каждом шаге разложения + 1 (для реализации листьев дерева разложения).

Чтобы найти длину пути λ в подсхеме, реализующей формулу (2) или (1) на очередном шаге разложения, необходимо знать количество слагаемых sum в этой формуле.

Утверждение 1. Количество слагаемых sum в формуле (2) зависит от $m, n - m, g$ и $n - g$, а именно $sum = \min\{m + 1, g + 1, n - g + 1, n - m + 1\}$.

Доказательство. Формула (2) имеет вид $D_n^m(X) = \sum_{i=0}^m D_g^i(X^1) D_{n-g}^{m-i}(X^2)$. Разберем всевозможные случаи и приведем примеры.

1. Пусть $m > g, m < n - g$. Так как в разложении D_g^i индекс i начинается с 0 и не может быть больше g , то получаем в формуле (2) $(g + 1)$ -е слагаемое. Рассматривая связь чисел m, g и $n - g$, получаем, что в этом случае

$$g + 1 = \min\{g + 1, m + 1, n - g + 1, n - m + 1\}.$$

Действительно, условия $m > g$ и $m < n - g$ можно переписать в виде двойного неравенства $g < m < n - g$. Откуда находим

$$g + 1 < m + 1 < n - g + 1.$$

Иначе говоря,

$$g + 1 = \min\{g + 1, m + 1, n - g + 1\}.$$

Кроме того, из $m < n - g$ следует, что $n - m > g$, следовательно, $n - m + 1 > g + 1$. Таким образом,

$$g + 1 = \min\{g + 1, m + 1, n - g + 1, n - m + 1\}.$$

Например, пусть $g = 4, m = 5$ и $n = 10$. Тогда формула (2) имеет вид:

$$D_{10}^5 = D_4^0(X^1)D_6^5(X^2) \vee D_4^1(X^1)D_6^4(X^2) \vee D_4^2(X^1)D_6^3(X^2) \vee D_4^3(X^1)D_6^2(X^2) \vee D_4^4(X^1)D_6^1(X^2)$$

То есть количество слагаемых $sum = 5 = g + 1$.

2. Пусть $m < g, m > n - g$. В этом случае слагаемых $n - g + 1$.

Поскольку из неравенств $m < g, m > n - g$ следует, что $n - g + 1 < m + 1 < g + 1$, то $n - g + 1 = \min\{n - g + 1, m + 1, g + 1\}$. Кроме того, из $m < g$ находим: $-m > -g$, следовательно

$$n - m + 1 > n - g + 1.$$

Таким образом, $n - g + 1 = \min\{g + 1, m + 1, n - g + 1, n - m + 1\}$.

Например, пусть $g = 6, m = 4$ и $n = 9$. Тогда формула (2) имеет вид:

$$D_9^4 = D_6^1(X^1)D_3^3(X^2) \vee D_6^2(X^1)D_3^2(X^2) \vee D_6^3(X^1)D_3^1(X^2) \vee D_6^4(X^1)D_3^0(X^2).$$

То есть количество слагаемых $sum = 4 = n - g + 1$.

3. Пусть $m < g$, $m < n - g$. В этом случае слагаемых $m + 1 = \min\{m + 1, g + 1, n - g + 1\}$. Кроме того, из $m < g$, $m < n - g$, получаем: $m < g < n - m \Rightarrow m < n - m$, а значит $m + 1 < n - m + 1$.

Следовательно, $m + 1 = \min\{m + 1, g + 1, n - g + 1, n - m + 1\}$.

Например, пусть $g = 4$, $m = 2$ и $n = 8$. Тогда формула (2) имеет вид:

$$D_8^2 = D_4^0(X^1)D_4^2(X^2) \vee D_4^1(X^1)D_4^1(X^2) \vee D_4^2(X^1)D_4^0(X^2).$$

То есть количество слагаемых $sum = 3 = m + 1$.

4. Пусть $m > g$, $m > n - g$. В этом случае i в формуле (2) должно удовлетворять условиям $i \leq g$ и $m - i \leq n - g$. Откуда находим, что $m - n + g \leq i \leq g$. Это неравенство имеет $g - (m - n + g) + 1 = n - m + 1$ целых решений.

При этом из $m > g$, $m > n - g$, получаем: $n - m < g < m$, из чего следует $n - m + 1 < g + 1 < m + 1$. Иначе говоря, $n - m + 1 = \min\{m + 1, g + 1, n - m + 1\}$.

Кроме того, из условия $m > g$ получаем: $-m < -g \Rightarrow n - m + 1 < n - g + 1$. Следовательно,

$$n - m + 1 = \min\{m + 1, g + 1, n - g + 1, n - m + 1\}.$$

Например, пусть $g = 4$, $m = 6$ и $n = 8$. Тогда формула (2) имеет вид

$$D_8^6 = D_4^2(X^1)D_4^4(X^2) \vee D_4^3(X^1)D_4^3(X^2) \vee D_4^4(X^1)D_4^2(X^2).$$

То есть количество слагаемых $sum = 3 = n - m + 1$.

Итак, во всех случаях получаем, что $sum = \min\{m + 1, g + 1, n - g + 1, n - m + 1\}$. Утверждение доказано.

Следствие 1. Количество слагаемых sum в формуле (1) зависит от m , $n - m$, k и $n - k$, а именно $sum = \min\{m + 1, k + 1, n - k + 1, n - m + 1\}$.

Доказательство. Так как формула (1) является частным случаем формулы (2), а именно получается из формулы (2) при $g = k$, то $c = \min\{m + 1, k + 1, n - k + 1, n - m + 1\}$. Следствие доказано.

Каждое из слагаемых текущего шага разложения реализуется некоторой подсхемой. Выясним, какая из них даст наибольшую длину пути. В силу того, что C_n^m достигает максимума при $m = n/2$, таким слагаемым будет слагаемое, у которого i наиболее близкое к $g/2$ или $m - i$, наиболее близкое к $(n - g)/2$.

Приведем алгоритм нахождения максимальной длины пути L в детекторе 1 для D_n^m , где $0^\circ < m^\circ \leq n$, k – число входов в ПЛБ. Как было сказано выше, для нахождения максимальной длины пути в подсхеме на текущем шаге разложения следует выбирать из всех множителей такое D_p^q , у которого $p = \max(k, n - k)$, а q – число, наиболее близкое к $p/2$.

Найдем число μ шагов разложения. Так как количество подмножеств переменных мощности k (одно из подмножеств может иметь мощность меньше k), на которые разбивается множество переменных мощности n равно $\lceil n/k \rceil$, то число шагов разложения $\mu = \lceil n/k \rceil - 1$.

По следствию 1, число слагаемых sum в формуле разложения для различных D_p^q находится по формуле $sum = \min(k + 1, p - k + 1, q + 1, p - q + 1)$.

Теперь найдем $CountCLB$ – количество ПЛБ, необходимых для реализации sum слагаемых. В каждом слагаемом два множителя и каждый множитель – это отдельный вход в некоторый ПЛБ. Так как число входов в ПЛБ равно k , то для реализации всех sum слагаемых потребуется $CountCLB = \lceil 2sum/k \rceil$ ПЛБ.

Так как $sum = \min(k + 1, p - k + 1, q + 1, p - q + 1)$, то $sum \leq k + 1$. Следовательно, $CountCLB = \lceil 2sum/k \rceil \leq \lceil (2k + 2)/k \rceil \leq 3$. Поэтому максимальная длина пути λ_{\max} в подсхеме для реализации формулы разложения на каждом шаге не может быть больше 2. В итоге получаем, если $CountCLB \geq 2$, то реализация формулы потребует длины пути в подсхеме $\lambda = 2$, иначе $\lambda = 1$. Особая ситуация возникает на первом шаге разложения, т.е. при $\mu = 1$:

- а) если $CountCLB = 2$, то выходы этих ПЛБ и есть выходы детектора, т.е. $\lambda = 1$;
- б) если $CountCLB = 3$, то, как и в общем случае, $\lambda = 2$.

Алгоритм нахождения L для детектора 1

1. Положим $q = m, p = n$.
2. Вычислим число шагов разложения $\mu = \lceil p/k \rceil - 1, L = 1$.
3. Вычислим число слагаемых $sum = \min(k + 1, p - k + 1, q + 1, p - q + 1)$.
4. Вычислим количество ПЛБ $CountCLB$, необходимых для реализации sum слагаемых $CountCLB = \lceil 2sum/k \rceil$.
5. Определяем λ :
 - а) если имеем первый уровень разложения (т.е. $\mu = 1$) и количество $CountCLB$, необходимых для реализации формулы ПЛБ, равно 3, то $\lambda = 2$, в противном случае $\lambda = 1$;
 - б) если не первый уровень разложения (т.е. $\mu \neq 1$) и количество $CountCLB$, необходимых для реализации формулы ПЛБ, ≥ 2 , то $\lambda = 2$, в противном случае $\lambda = 1$.
6. $L = L + \lambda, \mu = \mu - 1$.
7. Если $\mu = 0$, то нахождение L закончено. Иначе, полагаем $p = \max(p, p - k), q -$ число, наиболее близкое к $p/2$, и переходим на шаг 3.

Приведем алгоритм нахождения максимальной длины пути L в детекторе 2 для D_n^m , где $0^\circ < m^\circ < n, k -$ число входов в ПЛБ.

В детекторе 2 наибольшее число слагаемых sum_{max} на каждом шаге разложения может сильно отличаться от sum_{max} предыдущего шага разложения (наибольшее число на первом шаге разложения; на каждом последующем шаге число слагаемых не изменяется или становится меньше). Это обусловлено тем, что для данного n C_n^m достигает максимального значения при $m = n/2$ и с уменьшением n C_n^m уменьшается. Увеличение числа слагаемых в формуле разложения может увеличить число ПЛБ в детекторе, что способно привести к увеличению максимальной длины пути в детекторе.

Разложение по формуле (2) заканчивается, если мощность каждого из подмножеств переменных становится меньше либо равной k (последний шаг разложения). По утверждению 1, число слагаемых sum в формуле разложения для различных D_n^m находится по формуле $sum = \min\{m + 1, g^\circ + 1, n - g + 1, n - m + 1\}$. Теперь найдем $CountCLB -$ количество ПЛБ, необходимых для реализации sum слагаемых. В каждом слагаемом два множителя и каждый множитель – это отдельный вход в некоторый ПЛБ. Так как число входов в ПЛБ равно k , то для реализации всех x слагаемых потребуется $CountCLB = \lceil 2sum/k \rceil$ ПЛБ. Но если $CountCLB > 1$ (не первый шаг разложения) или если $CountCLB^\circ > 2$ (первый шаг разложения), то требуются дополнительные ПЛБ для реализации полученного числа $CountCLB$ ПЛБ. Это увеличивает максимальную длину пути λ_{max} в подсхеме, реализующей формулу разложения. Максимальная длина пути λ_{max} увеличивается до тех пор, пока количество дополнительных ПЛБ, необходимых для реализации вновь полученного числа $CountCLB$ ПЛБ, не станет равным 1 (2).

Алгоритм нахождения L для детектора 2

1. Положим $q = m, p = n$.
2. $L = 1$.
3. Найдем такие g и $p - g$, что $p = g + (p - g) = kq^1 + kq^2 + r$, где $k -$ число входов в ПЛБ, $r < k$, а $kq^1 -$ ближайшее к $\lceil p/2 \rceil$ число, кратное k .
4. Вычислим $sum = \min(g + 1, p - g + 1, q + 1, p - q + 1)$ и $CountCLB = \lceil 2sum/k \rceil, \lambda = 1$.
5. Если не первый шаг разложения и $CountCLB > 1$, то до тех пор, пока количество необходимых для реализации формулы ПЛБ $CountCLB = \lceil CountCLB/k \rceil$ больше 1, увеличиваем λ .

6. Если первый шаг разложения и $CountCLB > 2$, то до тех пор, пока количество необходимых для реализации формулы ПЛБ $CountCLB = \lceil CountCLB/k \rceil$ больше 2, увеличиваем λ .

7. $L = L + \lambda$.

8. Если $g \leq k$ и $p - g \leq k$, то нахождение L закончено. Иначе $p = g$, q – число, наиболее близкое к $p/2$ и идем на шаг 3.

Сравним сложность детектора 1 и детектора 2. Будем считать, что при четных k используются двухвыходные ПЛБ (табл. 2).

Таблица 2

Сравнение сложности детектора 1 и детектора 2

m	n	k	Оценка максимальной длины пути		Количество ПЛБ	
			Детектор 1	Детектор 2	Детектор 1	Детектор 2
6	12	4	5	5	31	31
8	16	4	7	5	55	60
3	16	4	6	4	32	32
10	20	4	9	7	85	82
3	20	4	8	6	43	43
3	64	4	30	8	164	164
10	64	4	31	11	448	471
20	64	4	31	12	703	728
32	64	4	31	12	811	840

Количество ПЛБ, сопоставляемых листьям дерева разложения (все ПЛБ реализуют функции вида D_k^q , где $0 \leq q \leq k$) одинаково при любом из рассматриваемых способах разбиения (так как в обоих случаях при разбиении идет ориентация на число k входов в ПЛБ). Разница в количестве ПЛБ начинается при многократном применении формул (1) и (2) при реализации D_p^q , где $p > k$.

Заключение

Таким образом, при $n > 3k$ оценки максимальных длин путей детекторов, синтезированных методом, предлагаемым в [9], оказываются меньше, чем оценки длин путей детекторов, синтезированных методом, предлагаемым в [8]. Количество ПЛБ, используемых при синтезе детектора рассматриваемыми методами, существенно не отличается.

ЛИТЕРАТУРА

1. Parag K.L. Self-Checking and Fault-Tolerant Digital Design. Morgan Kaufmann Pub., 2000. 400 p.
2. Anderson D.A., Metze G. Design of totally self-checking check circuits for m-out-of-n codes // IEEE Trans. Computers C-22. 1973. P. 263–269.
3. Marouf M.A., Friedman A.D. Efficient design of sel-checking checker for any m-out-of-n code // IEEE Trans. Computers C-27. 1978. P. 482–90.
4. Ubar R., Raik J., Vierhaus H.-T. Design and Test Technology for Dependable Systems-on-Chip. New York : IGI Global, 2011. 578 p.
5. Blanke M., Kinnaert M., Lunze J., Staroswiecki M. Diagnosis and Fault-Tolerant Control. 2nd ed. Berlin : Springer-Verlag Berlin Heidelberg, 2006. 672 p.
6. Fujiwara E. Code Design for Dependable Systems: Theory and Practical Applications. John Wiley & Sons, Inc., 2006. 720 p.
7. Göessel M., Ocheretny V., Sogomonyan E., Marienfeld D. New Methods of Concurrent Checking. Ed. 1. Dordrecht : Springer Science + Business Media B. V., 2008. 182 p.
8. Матросова А.Ю., Никитин К.В. Синтез самотестируемого детектора (m, n) -кодов на программируемых логических блоках // Вестник Томского государственного университета. Приложение. 2003. № 6. С. 124–136.
9. Буторина Н.Б., Цидендоржиева С.Р. Построение самотестируемого детектора равновесных кодовых слов // Новые информационные технологии в исследовании сложных структур : 7-я рос. конф. с междунар. участием. Томск : Том. гос. ун-т, 2008. С. 44.

Поступила в редакцию 7 сентября 2018 г.

Butorina N.B., Pakhomova E.G. (2019) ON THE LENGTHS OF THE PATHS IN SELF-TESTING CHECKERS BASED ON CLB. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie vychislitel'naja tehnika i informatika* [Tomsk State University Journal of Control and Computer Science]. 48. pp. 92–102

DOI: 10.17223/19988605/48/11

Self-testing checkers, in particular, (m, n) -code checkers are widely used in self-checking circuits design. There is a special formula $D_n^m(X) = \sum_{i=0}^m D_g^i(X^1) D_{n-g}^{m-i}(X^2)$ representing all (m, n) -code words. This paper compares two self-testing (m, n) -code checkers designed on the base of configurable logical blocks (CLB) by using a multiple decomposition of the formula. The form of the formula obtained by decomposition depends on the set of represented code words (i.e., values of m and n), the number k of the CLB inputs and the way of partitioning variables into subsets. In the first approach, $g = k$. If $n - g > k$ then the decomposition formula is applied again for the obtained function $D_{n-g}^{m-i}(X^2)$, etc. In the second approach, g is the least integer which is greater than or equal to $n/2$. If $g > k$ and/or $n - g > k$, then the decomposition formula is applied again for all obtained functions $D_g^i(X^1)$, $D_{n-g}^{m-i}(X^2)$, $i = \overline{0, m}$, etc. In the paper, we analyze how the choice of g (or the way of partitioning variables into subsets) influence on the maximum path length in the designed checker and on the number of CLBs in the checker.

Keywords: selftesting; checker; configurable logical blocks (CLB); (m, n) -code.

BUTORINA Nataly Borisovna (National Research Tomsk State University, Tomsk, Russian Federation).

E-mail: nnatta07@mail.ru

PAKHOMOVA Elena Grigorievna (Candidate of Physics and Mathematics, Associate Professor, National Research Tomsk State University, Tomsk, Russian Federation).

E-mail: peg@tpu.ru

REFERENCES

1. Parag, K.L. (2000) *Self-Checking and Fault-Tolerant Digital Design*. Morgan Kaufmann.
2. Anderson, D.A. & Metze, G. (1973) Design of totally self-checking check circuits for m-out-of-n codes. *IEEE Trans. Computers* C-22. pp. 263–69. DOI: 10.1109/T-C.1973.223705
3. Marouf, M.A. & Friedman, A.D. (1978) Efficient design of sel-checking checker for any m-out-of-n code. *IEEE Trans. Computers* C-27. pp. 482–90. DOI: 10.1109/TC.1978.1675138
4. Ubar, R., Raik, J. & Vierhaus, H-T. (2011) *Design and Test Technology for Dependable Systems-on-Chip*. New York: IGI Global.
5. Blanke, M., Kinnaert, M., Lunze, J. & Staroswiecki, M. (2006) *Diagnosis and Fault-Tolerant Control*. 2nd ed. Berlin: Springer-Verlag Berlin Heidelberg.
6. Fujiwara, E. (2006) *Code Design for Dependable Systems: Theory and Practical Applications*. John Wiley & Sons.
7. Göessel, M., Ocheretny, V., Sogomonyan, E. & Marienfeld, D. (2008) *New Methods of Concurrent Checking*. Dordrecht: Springer Science+Business Media B. V.
8. Matrosova, A.Yu. & Nikitin, K.V. (2003) Sintez samotestiruемого детектора (m, n) -кодov на программруемых логических блоках [Design of a self-testing (m,n) -code checkers using configurable logic blocks]. *Vestnik Tomskogo gosudarstvennogo universiteta. Prilozhenie*. 6. pp. 124–136.
9. Butorina, N.B. & Tsidendorzheva, S.R. (2008) [Design of a self-testing (m,n) -code checkers]. *Novye informatsionnye tekhnologii v issledovanii slozhnykh struktur* [Computer-aided technologies in applied mathematics. ICAM 2008]. Proc. of the 7th Conference. Tomsk: Tomsk State University. pp. 44. (In Russian).

УДК 004.054 (004.048)

DOI: 10.17223/19988605/48/12

С.Г. Мосин

**ОПТИМИЗАЦИЯ ПОСТРОЕНИЯ НЕЙРОМОРФНОГО СПРАВОЧНИКА
НЕИСПРАВНОСТЕЙ ДЛЯ ТЕСТИРОВАНИЯ И ДИАГНОСТИКИ
АНАЛОГОВЫХ ИНТЕГРАЛЬНЫХ СХЕМ**

Работа выполнена за счет средств субсидии, выделенной в рамках государственной поддержки Казанского (Приволжского) федерального университета в целях повышения его конкурентоспособности среди ведущих мировых научно-образовательных центров.

Статья посвящена вопросам снижения сложности построения справочников неисправностей для аналоговых интегральных схем на основе искусственной нейронной сети (НС). Представлены преимущества нейроморфных справочников неисправностей (НСН) – ассоциативный режим работы и слабое влияние количества рассматриваемых неисправностей на архитектуру НС. Обсуждаются варианты построения НСН в аспекте больших данных. Предложен подход к выбору существенных характеристик контролируемых параметров для тестирования и диагностики неисправностей, обеспечивающий уменьшение размерности обучающего множества. Метод главных компонент (МГК) и критерий, основанный на объясненной дисперсии остатков, применяются для уменьшения количества коэффициентов, используемых для обучения НС. Представлен сквозной маршрут формирования НСН. Экспериментальные результаты демонстрируют эффективность предложенного подхода в виде сокращения времени и вычислительных затрат на формирование НСН, обеспечивающего высокий уровень покрываемости неисправностей до 100%.

Ключевые слова: метод главных компонент; нейросетевой справочник неисправностей; тестирование; диагностика; аналоговые интегральные схемы.

Диагностика неисправностей интегральных схем (ИС) – важный этап производственного цикла, обеспечивающий выявление места возникновения неисправностей в ходе тестирования и гарантирующий высокое качество партии ИС на выходе. Тестирование и диагностика аналоговых ИС представляют существенную функциональную сложность по сравнению с цифровыми ИС [1]. Во многом этому способствуют непрерывный характер обрабатываемых сигналов, сложные нелинейные зависимости входных и выходных сигналов, допуски на параметры внутренних компонентов, которые случайным образом влияют на отклонения параметров и передаточной функции от номинальных значений, чувствительность выходных характеристик к отклонениям внутренних и внешних параметров, возможность возникновения не только катастрофических, но и параметрических неисправностей, отсутствие эффективных моделей аналоговых неисправностей и др.

Цель предлагаемой работы – поиск и исследование метода выбора существенных характеристик выходного отклика проверяемой схемы на приложенный тестовый сигнал, которые обеспечивают снижение вычислительных и временных затрат на обучение нейроморфного справочника неисправностей без снижения покрываемости рассматриваемых неисправностей. Для достижения поставленной цели в работе определены и решены следующие задачи: представление непрерывного аналогового сигнала во временной области дискретными значениями в частотной области на основе вейвлет-преобразования; сокращение размерности матрицы коэффициентов вейвлет-преобразования, используемых для обучения нейронной сети, с применением метода главных компонент; программная реализация предложенного метода выбора существенных характеристик выходных откликов; экспериментальное исследование показателей обучения НСН с применением входных векторов различной длины; анализ полученных результатов.

1. Методы функционального тестирования и диагностики неисправностей аналоговых схем

Существует два подхода к функциональному тестированию аналоговых ИС: методы на основе соответствия спецификации и на основе моделирования неисправностей [2–3]. В ходе первого подхода контролируют соответствие выходных характеристик проверяемой схемы значениям спецификации. Контроль осуществляют как в рабочем режиме, так и в режиме тестирования, когда на вход схемы подают специальные тестовые сигналы. Второй подход получил большее распространение, поскольку обеспечивает решение задачи не только тестирования, но и диагностики неисправностей [4]. Оценка влияния возможных неисправностей на работу схемы – важный подготовительный этап, в ходе которого формируют справочник неисправностей, а затем последовательно внедряют неисправности одну за другой в исправную схему и выполняют моделирование ее поведения. Выходные отклики, измеряемые в тестовых узлах схемы, для исправного состояния и для каждой рассмотренной неисправности сохраняют в справочнике неисправностей. Для аналоговых схем различают катастрофические и параметрические неисправности. Катастрофические связывают с возникновением в схеме эффектов обрыва цепи или короткого замыкания. При параметрических неисправностях наблюдается отклонение параметров внутренних компонентов за пределы допусков. Для аналоговой схемы множество катастрофических неисправностей конечно, а множество параметрических – бесконечно. Наличие допусков на параметры внутренних компонентов схемы, в свою очередь, определяет случайность поведения как исправной, так и неисправной схемы. Таким образом, для достоверного представления поведения схемы возникает необходимость использования метода статистических испытаний при моделировании каждого состояния устройства. Как следствие, оценка поведения исправной схемы и с учетом рассматриваемых неисправностей – сложная вычислительная задача, требующая существенных временных затрат. Использование современных высокопроизводительных вычислительных систем для моделирования неисправностей в аналоговых схемах позволяет получить результат за приемлемое время за счет распараллеливания задачи на нескольких вычислительных узлах [5].

Размерность справочника неисправностей (СН) и механизм последовательного поиска соответствий в структуре СН – слабые стороны данного метода. Развитие инструментов машинного обучения открыло возможность перехода к НСН, функционирующим в ассоциативном режиме, архитектура которых нечувствительна к количеству рассматриваемых неисправностей. Методы машинного обучения и анализа данных позволяют преодолеть проблему больших данных о поведении аналоговой схемы, полученных в ходе моделирования, при построении эффективного НСН, обеспечивающего тестирование и диагностику неисправностей. При этом возникают сложности, связанные с выбором архитектуры нейронной сети, а также структуры и количества обучающих наборов [6–8]. Выбор параметров, которые будут использованы для обучения нейронной сети, – важная задача, результат которой оказывает существенное влияние на сходимость процесса обучения, его длительность и качество покрытия неисправностей при распознавании откликов тестируемой схемы.

2. Декомпозиция маршрута построения нейроморфного справочника неисправностей для аналоговых схем

Формирование условий и механизмов тестирования и диагностики разрабатываемой схемы осуществляют в рамках тестопригодного проектирования, учитывая структуру схемы и ее особенности уже на ранних этапах проектирования. Маршрут формирования справочника неисправностей на основе нейронной сети представлен на рис. 1.

Исходными данными являются список соединений (*netlist*) оригинальной схемы; список возможных неисправностей $\mathbf{FL} = [f_j]$, $j = 1..N$, который может включать как катастрофические, так и

параметрические неисправности; модели неисправностей. Для каждого i -го состояния схемы $c_i \in C$ формируется соответствующее описание схемы, которое поступает на моделирование,

$$C = C_{ff} \cup C_{cf} \cup C_{pf}; \quad (1)$$

$$C_{ff} = c_1; \quad (2)$$

$$C_{cf} = [c_k], k = 2..(N_{cf} + 1); \quad (3)$$

$$C_{pf} = [c_l], l = (N_{cf} + 2)..(N_{cf} + 1 + N_{pf}), \quad (4)$$

где C_{ff} – состояние исправной схемы; C_{cf} – множество состояний схемы с катастрофическими неисправностями; C_{pf} – множество состояний схемы с параметрическими неисправностями; N_{cf} – количество рассмотренных катастрофических неисправностей; N_{pf} – количество рассмотренных параметрических неисправностей.

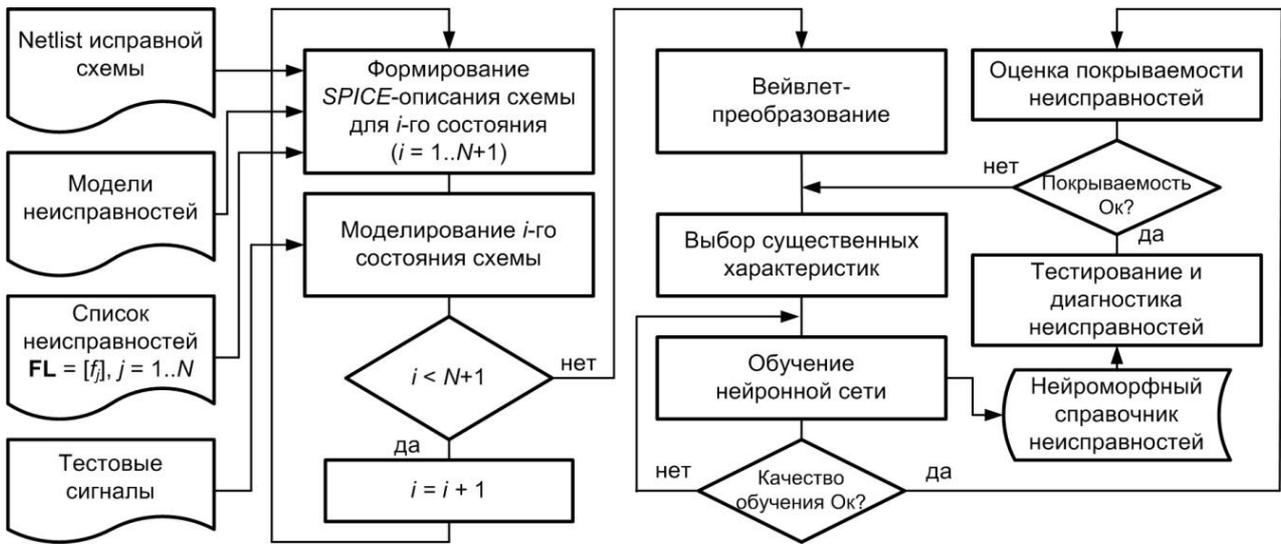


Рис. 1. Маршрут формирования нейроморфного справочника неисправностей
Fig. 1. The route of formation of the neuromorphic fault dictionary

Моделирование. Для учета динамических и нелинейных свойств аналоговой схемы моделирование осуществляется во временной области с использованием анализа переходного режима. Для учета влияния допусков в ходе моделирования использован метод Монте-Карло, который обеспечивает вычисление наиболее реалистичных значений выходных откликов схемы с учетом статистической функции распределения параметров внутренних компонентов.

Значение параметра для каждого внутреннего компонента схемы p_i на каждом шаге моделирования рассчитывают по формуле

$$P_i = P_{nom_i} (1 + \Delta_i \xi_i), \quad (5)$$

где P_{nom_i} – номинальное значение параметра i -го компонента из списка соединений; Δ_i – относительный разброс параметра p_i ; ξ_i – центрированная случайная величина, принимающая значение на отрезке $(-1, +1)$ по нормальному закону с нулевым математическим ожиданием и среднеквадратическим отклонением $\sigma = 0,25$. В ходе моделирования аналоговой схемы методом Монте-Карло можно задавать закон распределения, соответствующий типовым разбросам параметров внутренних компонентов для реальной технологической линии.

Многочисленное повторение расчета выходной функции при случайных отклонениях параметров внутренних компонентов позволяет сформировать множество выходных откликов схемы

$$R = [R_i], R_i = [r_i^m], R \in \mathfrak{R}_{NR \times t}, \quad (6)$$

где r_i^m – вектор временных отсчетов выходного отклика схемы длины t , $i = 1..N+1$ – количество состояний схемы, m – количество итераций метода Монте-Карло, $NR = mN + m$, t – количество временных точек.

Вейвлет-преобразование. В силу влияния шумов, нелинейных искажений, фазового сдвига и тому подобного непосредственное использование выходных откликов для тестирования и диагностики нежелательно без предварительной обработки или выделения из них существенных характеристик. В маршруте предложено использовать вейвлет-преобразование (ВП) для представления непрерывного аналогового сигнала во временной области дискретными значениями в частотной области

В ходе вейвлет-преобразования одномерного сигнала реализуется его представление в виде обобщенного ряда или интеграла Фурье (7) по системе базисных функций (8), сконструированных из материнского вейвлета $\psi(t)$, за счет операций сдвига во времени (b) и изменения временного масштаба (a). Множитель $1/\sqrt{a}$ обеспечивает независимость нормы этих функций от масштабирующего числа a :

$$W(a, b) = \int_{-\infty}^{\infty} f(t) \psi_{a,b}^*(t) dt, \quad (7)$$

$$\psi_{ab} = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right). \quad (8)$$

В результате вейвлет-преобразования

$$WT : \mathbf{R} \rightarrow \mathbf{X} \quad (9)$$

непрерывный выходной отклик представляют кортежем коэффициентов масштабирования a и сдвига b : $x = [a; b]$, а множество рассматриваемых откликов – матрицей коэффициентов вейвлет-преобразования $\mathbf{X} = [x_i], i = 1..N, \mathbf{X} \in \mathfrak{R}_{N \times M}$.

Выбор существенных характеристик. Не все коэффициенты ВП одинаково полезны для определения различий откликов одних состояний схемы от откликов других состояний. Использование большого числа коэффициентов на входе НС, в том числе незначущих, приводит к росту сложности и вычислительным затратам на обучение. Кроме того, оно может спровоцировать проблемы сходимости или переобучения НС. В этой связи снижение размерности вектора коэффициентов ВП на входе НС за счет выбора из них наиболее значимых – важная задача. Для ее решения предложено использовать метод главных компонент, суть которого:

$$PCA : \mathbf{X} \rightarrow \hat{\mathbf{X}}, \quad \mathbf{X} \in \mathfrak{R}_{N \times M}, \quad \hat{\mathbf{X}} \in \mathfrak{R}_{N \times \hat{M}}, \quad M > \hat{M}. \quad (10)$$

Обучение нейронной сети. Для построения нейроморфного СН выбрана трехслойная нейронная сеть. Множество коэффициентов ВП $\hat{\mathbf{X}}$ представляем двумя подмножествами – обучающим ($\hat{\mathbf{X}}_{train}$) и тестовым ($\hat{\mathbf{X}}_{test}$). Обучение НС осуществляется с применением строк матрицы $\hat{\mathbf{X}}_{train}$ и ассоциированных с ними выходных векторов \mathbf{y}_i . Длина \mathbf{y}_i соответствует количеству рассматриваемых состояний схемы и определяет количество нейронов в выходном слое НС. Для представления состояния использована прямая схема, когда все разряды вектора принимают значение 0 за исключением разряда, соответствующего номеру текущего рассматриваемого состояния, где назначена -1 . Обучение НС завершается, если достигнут допустимый уровень ошибки обучения или превышен предел числа итераций обучения. При невыполнении первого или второго условий возможно повторное обучение НС.

Оценка покрываемости неисправностей. Качество обучения НС для решения задачи диагностики оценивают общим показателем покрываемости неисправностей в схеме

$$FC = \frac{N_{detected}}{N_{total}}, \quad (11)$$

где $N_{detected}$ – количество обнаруженных неисправностей, N_{total} – общее количество неисправностей, а также покрываемостью отдельно взятой неисправности

$$FC_i = \frac{N_{detected}^i}{N_{total}^i}, \quad (12)$$

где $N_{detected}^i$ – количество обнаруженных i -х неисправностей, N_{total}^i – общее количество i -х неисправностей.

Если покрываемость неисправностей ниже требуемого уровня, то необходимо пересмотреть множество используемых коэффициентов $\hat{\mathbf{X}}$ и повторить обучение НС.

3. Выбор существенных характеристик с использованием метода главных компонент

Метод главных компонент (МГК) обеспечивает проецирование данных одной размерности на подпространство меньшей размерности. Возникающие при этом остатки \mathbf{E} принято считать шумом, не содержащим значащей информации. Пусть матрица данных \mathbf{X} имеет I строк, которые называют образцами, и J столбцов, представляющих независимые переменные. Главными компонентами называют формальные переменные t_a , $a = 1..A$, которые являются линейной комбинацией исходных переменных x_j , $j = 1..J$,

$$\mathbf{t}_a = \mathbf{p}_{a1}\mathbf{x}_1 + \mathbf{p}_{a2}\mathbf{x}_2 + \dots + \mathbf{p}_{aJ}\mathbf{x}_J, \quad (13)$$

A – число главных компонент, причем $A < I$, $A < J$.

Используя (13), матрицу \mathbf{X} представляют произведением

$$\mathbf{X} = \mathbf{TP}^t + \mathbf{E} = \sum_{a=1}^A \mathbf{t}_a \mathbf{p}_a^t + \mathbf{E}, \quad (14)$$

где $\mathbf{T} \in \mathfrak{R}_{I \times A}$ – матрица счетов, $\mathbf{P} \in \mathfrak{R}_{J \times A}$ – матрица нагрузок и $\mathbf{E} \in \mathfrak{R}_{I \times J}$ – матрица остатков. Матрица счетов \mathbf{T} представляет проекции исходных образцов – J -мерных векторов \mathbf{x}_i – на A -мерное подпространство главных компонент. Строки матрицы \mathbf{T} содержат координаты образцов в новой системе координат, а столбцы представляют проекции всех образцов на одну новую координатную ось и являются ортогональными. Матрица нагрузок \mathbf{P} является матрицей перехода из исходного I -мерного пространства переменных \mathbf{x}_j в A -мерное подпространство главных компонент. Строки матрицы \mathbf{P} содержат коэффициенты, связывающие переменные t и x согласно (13).

МГК реализуется итерационной процедурой, в которой новые главные компоненты (PC – principal components) добавляются последовательно друг за другом [9]. Важно определить, когда остановить данный процесс, и выбрать достаточное число главных компонент. При малом количестве PC совокупная выборка данных будет неполной, при большом – возникает переоценка.

Для выбора достаточного количества PC предложено использовать следующий критерий:

$$\arg \min_{PC} (ERV_{PC} \geq ERV_{TH}), \quad (15)$$

где ERV_{PC} – объясненная дисперсия; ERV_{TH} – пороговое значение.

Результат моделирования методом главных компонент – матрица $\hat{\mathbf{X}}$, содержащая оценки, найденные по модели, построенной на обучающем наборе \mathbf{X} . Отклонение оценки от проверочного значения вычисляют как матрицу остатков

$$\mathbf{E} = \mathbf{X} - \hat{\mathbf{X}}, \quad (16)$$

элементы которой используют для расчета объясненной дисперсии остатков в (15)

$$ERV = 1 - \frac{\sum_{i=1}^I \sum_{j=1}^J e_{ij}^2}{\sum_{i=1}^I \sum_{j=1}^J x_{ij}^2}, \quad e_{ij} \in \mathbf{E}, \quad x_{ij} \in \mathbf{X}. \quad (17)$$

Применение метода главных компонент обеспечивает сокращение числа используемых коэффициентов для обучения нейронной сети ($\hat{M} < M$).

4. Экспериментальные исследования

Исследование предложенного подхода выполнено на контрольной схеме аналогового фильтра Саллена–Ки (рис. 2). В качестве тестового сигнала использован синусоидальный сигнал с амплитудой 1 В и частотой 72 Гц. Допуски на параметры внутренних компонентов схемы – для резисторов и конденсаторов – определены величиной 10%. Список рассмотренных одиночных неисправностей включает 14 катастрофических и 14 параметрических неисправностей. Катастрофические неисправности представлены эффектами обрыва цепи (ОЦ) и короткого замыкания (КЗ) для каждого компонента схемы. Параметрические неисправности задаются отклонениями $\pm 50\%$ от номинального значения для каждого компонента.

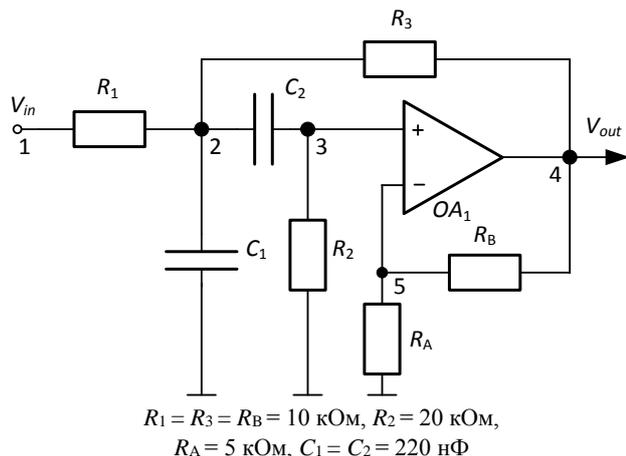


Рис. 2. Схема аналогового фильтра Саллена–Ки
Fig. 2. The circuit of analog Sallen–Key filter

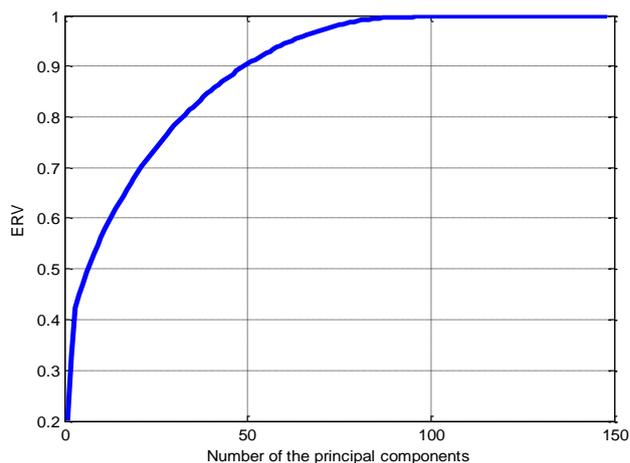


Рис. 3. График зависимости объясненной дисперсии остатков от количества главных компонент
Fig. 3. A graph of the explained variance of residuals versus the number of principal components

Моделирование схемы фильтра для 29 состояний (одно исправное и 28 неисправных) выполнено во временной области с использованием метода Монте-Карло. Для моделирования схем использован пакет PSpice CAPE CADENCE. Количество итераций для исправного состояния составляет 20 000, а для каждого неисправного состояния – 5 000. Выходные ответы измеряются в течение одного периода относительно тестового узла 4 с задержкой 80 мс с момента запуска теста.

Материнский вейвлет Добеши четвертого порядка (db4) использован для преобразования выходных откликов схемы из временной области в частотную область. В результате для каждого сигнала формируется вектор коэффициентов вейвлет преобразования длиной 148 элементов (по 74 для коэффициентов a и b). Обучающее множество включает 600 откликов из 20 000 для исправного состояния и по 200 откликов из 5 000 для каждого неисправного состояния. В результате была сформирована матрица коэффициентов ВП \mathbf{X}_{train} , включающая 6 200 строк по 148 элементов. Отклики, не включенные в это обучающее множество, образуют тестовое множество \mathbf{X}_{test} .

Матрицы счетов и нагрузок, а также объясненная остаточная дисперсия были рассчитаны с использованием МГК для множества \mathbf{X}_{train} . График для ERV представлен на рис. 3. Требуемое минимальное количество главных компонент определяется порогом $ERV_{TH} \geq 0,95$, поэтому в результирующем обучающем множестве $\hat{\mathbf{X}}_{train}$ будут использованы только 62 коэффициента для каждого отклика. При уменьшении объема значимой информации на 5% удалось уменьшить размер обучающего множества почти в 2,4 раза.

Трехслойная нейронная сеть использована для реализации НСН. Входной слой состоит из 62 нейронов в соответствии с количеством используемых коэффициентов в обучающем наборе для каждого отклика. Выходной слой состоит из 29 нейронов для представления рассматриваемых состояний схемы относительно соответствующего отдельного выходного нейрона.

Время, затраченное на обучение НСН для разных исходных данных и оцененное в ходе экспериментов, представлено в табл. 1.

Таблица 1

Характеристики обучения НСН

Номер НСН	1	2
Архитектура НС ¹	62 : 20 : 29	148 : 20 : 29
Использованные методы ²	ВП + МГК	ВП
Количество эпох обучения	109	119
Точность обучения	0,0106	0,0096
Величина ERV	0,9505	–
Время, с	24 147	271 800

¹ Показывает количество нейронов в первом : втором : третьем уровнях нейронной сети.

² ВП: вейвлет-преобразование временных откликов, ВП + МГК – преобразование матрицы с коэффициентами ВП временных откликов схемы методом главных компонент.

Построение и обучение нейронной сети осуществлялось с использованием набора инструментов Neural Network Toolbox пакета MATLAB на вычислительной системе с процессором Intel® Core™ i7-4770 @ 3,40 ГГц и оперативной памятью 8 ГБ.

Результаты покрытия отдельных неисправностей (FC), полученные для тестового множества, показаны в табл. 2. Анализ значений FC для режимов тестирования (FC_{test}) и диагностики (FC_{diag}), оцененных для отдельных неисправностей с использованием НСН, демонстрирует адекватность перехода от архитектуры 2 к архитектуре 1.

Таблица 2

Покрываемость отдельных неисправностей при тестировании и диагностике

Номер НСН	1		2		Номер НСН	1		2		
	Состояние схемы	FC_{test}	FC_{diag}	FC_{test}		FC_{diag}	Состояние схемы	FC_{test}	FC_{diag}	FC_{test}
Исправное	0,9059	–	0,8974	–	–	–	–	–	–	–
Катастрофические неисправности	R1_K3	1	1	1	1	R1_+ 50%	0,8274	0,7328	0,8372	0,7352
	R1_ОЦ	1	0,9888	0,9996	0,9866	R1_- 50%	0,9552	0,9198	0,9602	0,9296
	R2_K3	1	1	1	1	R2_+ 50%	0,9808	0,9020	0,9750	0,8970
	R2_ОЦ	1	1	1	1	R2_- 50%	1	0,9964	1	0,9054
	R3_K3	0,9978	0,8992	0,9978	0,8898	R3_+ 50%	0,9324	0,9002	0,9424	0,9064
	R3_ОЦ	1	0,9958	1	0,9962	R3_- 50%	0,9996	0,8123	0,9980	0,8038
	RA_K3	1	0,9900	1	0,9912	RA_+ 50%	0,9720	0,8944	0,9716	0,9020
	RA_ОЦ	1	1	1	1	RA_- 50%	1	0,8421	0,9996	0,8378
	RB_K3	1	1	1	1	RB_+ 50%	0,9734	0,8198	0,9726	0,8208
	RB_ОЦ	1	0,9874	1	0,9916	RB_- 50%	1	0,9412	1	0,9522
	C1_K3	1	1	1	1	C1_+ 50%	0,9982	0,8834	0,9990	0,8722
	C1_ОЦ	0,9822	0,4734	0,9976	0,4094	C1_- 50%	0,9998	0,9082	1	0,8754
	C2_K3	1	0,9964	1	0,9892	C2_+ 50%	0,9616	0,8370	0,9550	0,8452
	C2_ОЦ	1	1	1	1	C2_- 50%	1	0,9886	1	0,7912

Обнаружение исправного состояния фильтра на уровне 89,7% связано с подбором поведения исправной схемы и схемы с несколькими типами параметрических неисправностей. Увеличение количества откликов для исправной схемы в обучающем множестве позволит улучшить качество диагностики данного состояния, но в то же время приведет к увеличению ошибки второго рода, когда для соответствующих параметрических неисправностей схема будет признана исправной.

Таблица 3

Покрываемость отдельных неисправностей при тестировании и диагностике

Номер НСН	Тестирование				Диагностика		
	α	FC_{all}	FC_{cat}	FC_{par}	FC_{all}	FC_{cat}	FC_{par}
1	0,094	0,985	0,999	0,972	0,901	0,953	0,848
2	0,103	0,986	0,999	0,972	0,905	0,947	0,862

Примечание. α – значение ошибки первого рода для исправного состояния; FC_{all} – покрываемость для всех типов рассмотренных неисправностей; FC_{cat} – покрываемость для всех рассмотренных катастрофических неисправностей; FC_{par} – покрываемость для всех рассмотренных параметрических неисправностей.

Общие значения покрываемости неисправностей для режимов тестирования и диагностики, обеспеченные НСН, представлены в табл. 3. Ограничение количества главных компонент по уровню $ERV_{TH} \geq 0,95$ приводит к уменьшению размерности обучающего множества в 2,39 раза и времени обучения в 11,26 раза, обеспечивая при этом высокий уровень покрываемости неисправностей.

Заключение

Выбор существенных характеристик с использованием МГК позволил уменьшить размер обучающего множества почти в 3 раза и снизить затраты времени на обучение нейронной сети более чем в 11 раз. Полученный НСН обеспечивает решение задач как тестирования, так и диагностики неисправностей в ассоциативном режиме. Результаты экспериментальных исследований демонстрируют высокую покрываемость неисправностей: при тестировании для катастрофических неисправностей – до 99,9%, для параметрических неисправностей – до 97,2%, а при диагностике для катастрофических неисправностей – до 95,3% и параметрических – до 84,8%. Описание процесса формирования НСН в виде сквозного маршрута позволяет автоматизировать этот этап и использовать его в рамках тестопригодного проектирования аналоговых ИС.

ЛИТЕРАТУРА

1. Мосин С.Г. Анализ методов тестопригодного проектирования аналоговых и смешанных ИС // Известия высших учебных заведений. Электроника. 2007. № 1. С. 59–64.
2. Variyam P.N., Chatterjee A. Specification-driven test design for analog circuits // Proc. 1998 IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (Cat. No. 98EX223). 1998. P. 335–340. DOI: 10.1109/DFTVS.1998.732183.
3. Zhirabok A., Baranov A. Fault Diagnosis in Analog Electrical Circuits: Data-Driven Method // Proc. International Conference on Process Control. 2013. P. 1–6. DOI: 10.1109/PC.2013.6581389.
4. Мосин С.Г. Тестирование аналоговых схем с использованием нейросетевого сигнатурного анализатора // Вестник компьютерных и информационных технологий. 2012. № 10. С. 3–8.
5. Mosin S. Automated simulation of faults in analog circuits based on parallel paradigm // Proc. IEEE East-West Design & Test Symposium, Novi Sad, 2017. P. 1–6. DOI: 10.1109/EWDTS.2017.8110133.
6. Aminian M., Aminian F. Neural-network based analog circuit fault diagnosis using wavelet transform as preprocessor // IEEE Trans. CAS II. 2000. V. 47, No. 2. P. 151–156. DOI: 10.1109/82.823545.
7. Yuan L., He Y., Huang J. and Sun Y. A new neural-network-based fault diagnosis approach for analog circuits by using kurtosis and entropy as a preprocessor // IEEE Transactions on Instrumentation and Measurement. 2010. V. 59, No. 3. P. 586–595. DOI: 10.1109/TIM.2009.2025068.
8. Xiong J., Tian S. and Yang C. Fault Diagnosis for Analog Circuits by Using EEMD, Relative Entropy, and ELM // Computational Intelligence and Neuroscience. 2016. Article ID 7657054. 9 p. DOI: 10.1155/2016/7657054.
9. Jolliffe I.T. Principal Component Analysis. Springer, 2002.

Статья поступила 25 августа 2018 г.

Mosin S.G. (2019) OPTIMIZATION OF CONSTRUCTING THE NEUROMORPHIC FAULT DICTIONARY FOR TESTING AND DIAGNOSTICS OF ANALOG ICs. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naja tehnik i informatika* [Tomsk State University Journal of Control and Computer Science]. 48. pp. 103–111

DOI: 10.17223/19988605/48/12

Fault diagnostics of integrated circuits (IC) is an important stage of the production cycle, providing the detection of the faults location during testing and ensuring the high quality of the ICs batch at the output. Testing and diagnostics of analog ICs represent significant functional complexity compared to digital ICs. In many respects this is facilitated by the continuous nature of the signals being processed, the complex nonlinear dependences of the input and output signals, the tolerances on the parameters of internal components that randomly affect the deviations of the parameters and the transfer function from the nominal values, the sensitivity of the output characteristics to the deviations of internal and external parameters, possibility of appearance not only catastrophic, but also parametric faults, the lack of effective models for analog faults, etc.

There are two approaches to functional testing of analog ICs: specification-driven methods based on compliance with the specification and data-driven methods based on the fault simulation. The second approach is widely used because it provides the solution to the task of not only testing, but also faults diagnostics. The dimension of the fault dictionary (FD) and the mechanism for sequential matching in the FD structure are the weaknesses of this method. The development of machine learning tools has opened the possibility

of transition to neuromorphic FD, operating in an associative mode, the architecture of which is not sensitive to the number of considered faults. The choice of the parameters that will be used to train the neural network is an important task, the result of which has a significant influence on the convergence of the training process, its duration and the quality of the fault coverage. The scopes of the proposed work is to search for and study the method of selecting the essential characteristics of the output response for the circuit under test, which reduce the computational and time costs for training the neuromorphic FD (NFD) without reducing the coverage of the considered faults. To achieve this goal, the following tasks are defined and solved: representation of a continuous analog signal in the time domain by discrete values in the frequency domain based on the wavelet transform (WT); reducing the dimension of the matrix of WT-coefficients applied for training the neural network, using the principal components analysis; software implementation of the proposed method for selecting the essential characteristics of output responses; experimental study of indices of training the NFD with the use of input vectors of different lengths; analysis of the obtained results.

In the result of the wavelet transform the continuous output response is represented by a tuple of the scaling factors a and the shifts b : $x = [a; b]$, and the set of considered responses is a matrix of WT-coefficients $\mathbf{X} = [x_i]$, $i = 1..N$, $\mathbf{X} \in \mathfrak{R}_{N \times M}$. The application of the principal component analysis (PCA) provides a reduction in the number of coefficients used for training the neural network

$$PCA: \mathbf{X} \rightarrow \hat{\mathbf{X}}, \mathbf{X} \in \mathfrak{R}_{N \times M}, \hat{\mathbf{X}} \in \mathfrak{R}_{N \times \hat{M}}, M > \hat{M}.$$

PCA is implemented by an iterative procedure in which new principal components (PC) are added sequentially one after another. It is important to determine when to stop this process, and to take a sufficient number of principal components. With a small number of PCs, the aggregate sample of data will be incomplete, at large number an overvaluation arises. The criterion for selecting a sufficient number of PCs is defined:

$$\arg \min_{PC} (ERV_{PC} \geq ERV_{TH}),$$

where ERV_{PC} is the explained residual variance, ERV_{TH} is the threshold value.

The experimental studies were carried out for the analog filter circuit on operational amplifier using the developed software. The obtained results have demonstrated the effectiveness of the proposed method of selecting the essential characteristics of output responses, which is reflected in the reduction of time and computational costs for training. The resultant NFD provides the fault coverage up to 100%.

The work is performed according to the Russian Government Program of Competitive Growth of Kazan Federal University.

Keywords: principal component analysis; neuromorphic fault dictionary; testing; diagnostics; analog integrated circuits.

MOSIN Sergey Gennadievich (Candidate of Technical Sciences, Associate Professor, Kazan (Volga region) Federal University, Kazan, Russian Federation).

E-mail: smosin@ieee.org

REFERENCES

1. Mosin, S.G. (2007) Analysis of design-for-testability methods for analog and mixed-signal IC. *Elektronika*. 1. pp. 59–64. (In Russian).
2. Variyam, P.N. & Chatterjee, A. (1998) Specification-driven test design for analog circuits. *Proceedings IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*. pp. 335–340. DOI: 10.1109/43.875320
3. Zhirabok, A. & Baranov, A. (2013) Fault Diagnosis in Analog Electrical Circuits: Data-Driven Method. *Proc. International Conference on Process Control*. pp. 1–6. DOI: 10.1109/PC.2013.6581389
4. Mosin, S.G. (2012) A testing of analog circuits using neural network signature analyzer. *Vestnik komp'yuternykh i informatsionnykh tekhnologiy – Herald of Computer and Information Technologies*. 10. pp. 3–8. (In Russian).
5. Mosin, S. (2017) Automated simulation of faults in analog circuits based on parallel paradigm. *Proc. IEEE East-West Design & Test Symposium*. pp. 1–6. DOI: 10.1109/EWDTS.2017.8110133
6. Aminian, M. & Aminian, F. (2000) Neural-network based analog circuit fault diagnosis using wavelet transform as preprocessor. *IEEE Trans. CAS II*. 47(2). pp. 151–156. DOI: 10.1109/82.823545
7. Yuan, L., He, Y., Huang, J. & Sun, Y. (2010) A new neural-network-based fault diagnosis approach for analog circuits by using kurtosis and entropy as a preprocessor. *IEEE Transactions on Instrumentation and Measurement*. 59(3). pp. 586–595. DOI: 10.1109/TIM.2009.2025068
8. Xiong, J., Tian, S. & Yang, C. (2016) Fault Diagnosis for Analog Circuits by Using EEMD, Relative Entropy, and ELM. *Computational Intelligence and Neuroscience*. Article ID 7657054. DOI: 10.1155/2016/7657054
9. Jolliffe, I.T. (2002) *Principal Component Analysis*. Springer.

МЕМУАРЫ, ПАМЯТНЫЕ ДАТЫ, ПЕРСОНАЛИИ

75 лет профессору Анатолию Григорьевичу Дмитренко

3 августа 2019 года исполнилось 75 лет известному ученому в области распространения и рассеяния электромагнитных волн доктору физико-математических наук, профессору, Почетному работнику Высшего профессионального образования РФ **Анатолию Григорьевичу Дмитренко**.

А.Г. Дмитренко в 1961 г. поступил на радиофизический факультет Томского государственного университета. Окончил его в 1966 г. по специальности «Радиофизика и электроника», квалификация – «физик-радиоэлектроник». Сразу же после окончания университета проходил обучение в аспирантуре при кафедре радиофизики ТГУ. После окончания аспирантуры в январе 1970 г. был принят в лабораторию радиофизики Сибирского физико-технического института при Томском госуниверситете на должность младшего научного сотрудника. С 1973 г. А.Г. Дмитренко – старший научный сотрудник, а с 1987 г. – ведущий научный сотрудник той же лаборатории радиофизики. В этой должности Анатолий Григорьевич работал до июня 1996 г. В июне 1996 г. он поступил в очную докторантуру при кафедре радиофизики ТГУ, которую окончил в июне 1999 г. с представлением докторской диссертации. После окончания докторантуры принят на должность доцента кафедры исследования операций факультета прикладной математики и кибернетики ТГУ. С 2000 по 2018 г. профессор кафедры исследования операций. В настоящее время А.Г. Дмитренко является профессором кафедры прикладной математики, образованной путем объединения кафедр исследования операций и прикладной математики. Анатолий Григорьевич читал или читает такие курсы, как «Общая физика», «Технические средства защиты информации», «Электроника и схемотехника», «Системы и сети передачи информации», «Методы решения некорректных задач», руководит научной работой студентов и аспирантов.

Кандидатскую диссертацию на тему «Излучение электромагнитных диполей, расположенных на неоднородности типа “береговой линии”» защитил в 1973 г. по специальности 01.04.03 – радиофизика. Докторскую диссертацию на тему «Численное решение задач электромагнитного рассеяния на неосесимметричных телах методом дискретных источников» защитил в 1999 г. по той же специальности. Имеет ученое звание «старший научный сотрудник» по специальности «Радиофизика», которое присвоено ВАК в 1991 г., а также ученое звание «профессор» по кафедре защиты информации и криптографии, которое присвоено Федеральной службой по надзору в сфере образования и науки в 2007 г.

Основное направление научных исследований А.Г. Дмитренко – разработка математических моделей и численных методов решения задач электромагнитного рассеяния на объектах различной природы, реализация их в виде пакетов прикладных программ и использование последних для про-



гноза явлений рассеяния. На основе единой концепции им разработана группа численных методов для решения задач электромагнитного рассеяния на идеально проводящем теле, импедансном теле с переменным вдоль поверхности импедансом, магнитодиэлектрическом теле, киральном теле, идеально проводящем теле с магнитодиэлектрическим и киральным покрытиями, а также на структурах, образованных близко расположенными идеально проводящими, импедансными и диэлектрическими телами. Преимуществом разработанных методов по сравнению с известными является то, что они не накладывают ограничений на геометрию поверхности; последняя может быть задана даже в табличной форме.

Все разработанные методы реализованы в виде пакетов прикладных программ, которые позволяют рассчитывать рассеянные поля в произвольной точке пространства и исследовать зависимость амплитуды этих полей от параметров рассеивающего объекта. Результаты анализа полей и вытекающие из них технические рекомендации внедрены в большое число НИР, которые выполнялись по постановлениям Правительства, и использованы для разработки малоотражающих покрытий, создания алгоритмов пеленгования и устройств пеленгации с повышенной точностью пеленгования, создания специальных систем связи.

Анатолий Григорьевич Дмитренко опубликовал свыше 120 научных работ, в том числе 3 монографии и 3 учебно-методических пособия; им получено 4 авторских свидетельства на изобретения; участвовал в работе многих международных, всесоюзных и всероссийских конференций; является членом совета Д 212.267.12 по присуждению ученой степени доктора наук по специальности 05.13.01 – системный анализ, управление и обработка информации, членом совета Д 212.267.04 по присуждению ученой степени доктора наук по специальности 01.04.03 – радиофизика, а также членом совета Д 212.267.22 по присуждению ученой степени доктора наук по специальности 05.13.19 – методы и системы защиты информации, информационная безопасность.

А.Г. Дмитренко награжден медалью «За заслуги перед Томским университетом». Отмечен нагрудным знаком «Изобретатель СССР».

**Дорогой Анатолий Григорьевич!
Поздравляем Вас с юбилеем и желаем здоровья на многие годы!
Творческих Вам успехов!**

*Институт прикладной математики и компьютерных наук ТГУ
Редакционная коллегия журнала «Вестник ТГУ.
Управление, вычислительная техника и информатика»
Кафедра прикладной математики*

СВЕДЕНИЯ ОБ АВТОРАХ

БУТОРИНА Наталья Борисовна – старший преподаватель кафедры программирования Института прикладной математики и компьютерных наук Национального исследовательского Томского государственного университета. E-mail: nnatta07@mail.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

ЗАМЯТИН Александр Владимирович – доктор технических наук, заведующий кафедрой теоретических основ информатики, директор Института прикладной математики и компьютерных наук Национального исследовательского Томского государственного университета. E-mail: zamyatin@mail.tsu.ru

ЛАПАТИН Иван Леонидович – кандидат физико-математических наук, доцент кафедры прикладной информатики Института прикладной математики и компьютерных наук Национального исследовательского Томского государственного университета. E-mail: ilapatin@mail.ru

ЛИЗЮРА Ольга Дмитриевна – магистрант кафедры теории вероятностей и математической статистики Института прикладной математики и компьютерных наук Национального исследовательского Томского государственного университета. E-mail: oliztsu@mail.ru

МОСИН Сергей Геннадьевич – доцент, кандидат технических наук, директор института вычислительной математики и информационных технологий Казанского (Приволжского) федерального университета. E-mail: smosin@ieee.org

НАЗАРОВ Анатолий Андреевич – доктор технических наук, профессор, заведующий кафедрой теории вероятностей и математической статистики Института прикладной математики и компьютерных наук Национального исследовательского Томского государственного университета. E-mail: nazarov.tsu@gmail.com

НЕЖЕЛЬСКАЯ Людмила Алексеевна – доцент, доктор физико-математических наук, профессор кафедры прикладной математики Института прикладной математики и компьютерных наук Национального исследовательского Томского государственного университета. E-mail: ludne@mail.ru

ОСИПОВА Марина Анатольевна – кандидат физико-математических наук, доцент кафедры алгебры, геометрии и анализа Дальневосточного федерального университета, научный сотрудник Института прикладной математики ДВО РАН (г. Владивосток). E-mail: mao1975@list.ru

ПАУЛЬ Светлана Владимировна – кандидат физико-математических наук, доцент кафедры теории вероятностей и математической статистики Института прикладной математики и компьютерных наук Национального исследовательского Томского государственного университета. E-mail: paulsv82@mail.ru

ПАХОМОВА Елена Григорьевна – доцент, кандидат физико-математических наук, доцент кафедры программирования Института прикладной математики и компьютерных наук Национального исследовательского Томского государственного университета. E-mail: reg@tpu.ru

ПАШИНСКАЯ 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

ПРИСТУПА Андрей Викторович – кандидат технических наук, доцент кафедры теоретических основ информатики Института прикладной математики и компьютерных наук Национального исследовательского Томского государственного университета. E-mail: pristupa@sibmail.com

ПРОКОПЕНКО Светлана Анатольевна – кандидат технических наук, доцент кафедры информационных технологий в исследовании дискретных структур Национального исследовательского Томского государственного университета. E-mail: s.prokopenko@sibmail.com

РУБАН Анатолий Иванович – профессор, доктор технических наук, профессор кафедры информатики Института космических и информационных технологий Сибирского Федерального университета (г. Красноярск). E-mail: ai-rouban@mail.ru

РУДЬКО Игорь Михайлович – кандидат технических наук, старший научный сотрудник Института проблем управления им. В.А. Трапезникова РАН (г. Москва). E-mail: igor-rudko@mail.ru

СИДОРОВА Екатерина Филипповна – аспирант кафедры прикладной математики Института прикладной математики и компьютерных наук Национального исследовательского Томского государственного университета. E-mail: katusha_sidorova@mail.ru

ТРИШИН Василий Николаевич – младший научный сотрудник Института машиноведения Уральского отделения Российской академии наук (г. Екатеринбург). E-mail: trishinvn@yandex.ru

ФИЛИМОНОВ Александр Юрьевич – доцент Департамента информационных технологий и автоматизации Института радиоэлектроники и информационных технологий Уральского федерального университета (г. Екатеринбург). E-mail: a.filimonov@urfu.ru

ШАБАЛДИНА Наталия Владимировна – кандидат технических наук, доцент кафедры информационных технологий в исследовании дискретных структур Национального исследовательского Томского государственного университета. E-mail: nataliamailbox@mail.ru

ШИРОКОВА Екатерина Владимировна – аспирант кафедры информационных технологий в исследовании дискретных структур Национального исследовательского Томского государственного университета. E-mail: darusenkova@gmail.com

ЧЕРЕМИСИНОВ Дмитрий Иванович – доцент, кандидат технических наук, ведущий научный сотрудник лаборатории логического проектирования Объединенного института проблем информатики Национальной академии наук Беларуси (г. Минск, Беларусь). E-mail: cher@newman.bas-net.by

ЧЕРЕМИСИНОВА Людмила Дмитриевна – профессор, доктор технических наук, главный научный сотрудник лаборатории логического проектирования Объединенного института проблем информатики Национальной академии наук Беларуси (г. Минск, Беларусь). E-mail: cld@newman.bas-net.by

ЦИЦИАШВИЛИ Гурами Шалвович – профессор, доктор физико-математических наук, профессор кафедры алгебры, геометрии и анализа Дальневосточного федерального университета, главный научный сотрудник Института прикладной математики ДВО РАН (г. Владивосток). E-mail: guram@iam.dvo.ru

Научный журнал

**ВЕСТНИК
ТОМСКОГО
ГОСУДАРСТВЕННОГО
УНИВЕРСИТЕТА**

**УПРАВЛЕНИЕ,
ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА
И ИНФОРМАТИКА**

**TOMSK STATE UNIVERSITY
JOURNAL OF CONTROL AND COMPUTER SCIENCE**

2019. № 48

Редактор Е.Г. Шумская
Оригинал-макет Е.Г. Шумской
Редакторы-переводчики: Г.М. Кошкин; В.Н. Горенинцева
Дизайн обложки Л.Д. Кривцовой

Подписано к печати 26.09.2019 г. Формат 60x84¹/₈.
Гарнитура Times. Усл. печ. л. 13,72.
Тираж 250 экз. Заказ № 3958. Цена свободная.

Дата выхода в свет 15.10.2019 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)–52-96-75
Сайт: <http://publish.tsu.ru>; E-mail: rio.tsu@mail.ru