

ВЕСТНИК ТОМСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА

УПРАВЛЕНИЕ,
ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА
И ИНФОРМАТИКА

TOMSK STATE UNIVERSITY
JOURNAL OF CONTROL AND COMPUTER SCIENCE

Научный журнал

2014

№ 3 (28)

Свидетельство о регистрации: ПИ № ФС 77-29497 от 27 сентября 2007 г.



EDITORIAL BOARD

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

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

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

Sergey Vorobeychikov – Doctor of Science, Prof. of the Higher Mathematics and Mathematical Modeling Department

Gennady Koshkin – Doctor of Science, Prof. of the Theoretical Cybernetics Department

Yury Kostyuk – Doctor of Science, Prof. of the Theoretical Informatics Department

Anjela Matrosova – Doctor of Science, Prof., Head of the Programming Department

Anatoly Nazarov – Doctor of Science, Prof., Head of the Probability Theory and Mathematical Statistics Department

Sergey Sushchenko – Doctor of Science, Prof., Head of the Applied of Information Department, Dean of the Faculty of Informatics

EDITORIAL COUNCIL

Ana Rosa Cavalli
PhD, Prof.
University VII
Paris, France

Vladimir Dombrovskii
Doctor of Science, Prof.
Tomsk State University
Russia

Alexander Dudin
Doctor of Science, Prof.
Belarusian State University
Minsk, Republic Belorussia

Enco Orzinger
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 Science, Prof.
University of Technology
Tallinn, Estonia

Nina Yevtushenko
Doctor of Science, Prof.
Tomsk State University
Russia

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

РЕДАКЦИОННАЯ КОЛЛЕГИЯ

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

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

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

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

Коскин Геннадий Михайлович – проф., д-р физ.-мат. наук, проф. каф. теоретической кибернетики ТГУ

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

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

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

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

РЕДАКЦИОННЫЙ СОВЕТ

Ана Роза Ковали
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 техн. наук, проф.
Технологический университет,
Таллинн, Эстония

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

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

JORNAL INFO

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

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

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

The topics of the journal are the following:

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

Rules of registration articles are given in a site:

<http://journals.tsu.ru/informatics/>
ISSN 2311-2085 (Online), ISSN 1998-8605 (Print).

О ЖУРНАЛЕ

Журнал «Вестник Томского государственного университета. Управление, вычислительная техника и информатика» выходит ежеквартально и распространяется по подписке, его подписной индекс 44031 в объединённом каталоге «Пресса России». 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 «Вестник Томского государственного университета. Управление, вычислительная техника и информатика» включен в Перечень изданий для публикации основных результатов кандидатских и докторских диссертаций, а также входит в систему Российского Индекса Научного Цитирования (РИНЦ). Правила оформления статей приведены на сайте:

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

СОДЕРЖАНИЕ

УПРАВЛЕНИЕ ДИНАМИЧЕСКИМИ СИСТЕМАМИ

Параев Ю.И. Оптимальное управление двухсекторной экономикой	4
Шилин А.А., Букреев В.Г. Исследование оптимального и скользящего режимов управления с релейным элементом, охваченным обратной связью	12

ОБРАБОТКА ИНФОРМАЦИИ

Беккерман Е.Н., Горцев А.М. Эвристический алгоритм оценки числа состояний асинхронного МС-потока событий	20
Келлер Ю.А. Применение кластеризации данных на основе самоорганизующихся карт Кохонена при подборе скважин-кандидатов для методов увеличения нефтеотдачи	32
Китурко О.М., Маталыцкий М.А. Асимптотический анализ доходов в замкнутой НМ-сети с переменным числом приоритетных и бесприоритетных заявок	38

ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ

Бабанов А.М. Два современных подхода к семантическому моделированию – ORM и ERMM	46
Моисеев А.Н., Политов А.М., Рахимов В.В., Хомич М.О. Базовая архитектура системы совместного редактирования диаграмм с синхронизацией в режиме реального времени	57

ДИСКРЕТНЫЕ ФУНКЦИИ И АВТОМАТЫ

Бурдонов И.Б., Косачев А.А. Исследование графа взаимодействующими автоматами	67
---	----

ПРОЕКТИРОВАНИЕ И ДИАГНОСТИКА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Сапожников В.В., Сапожников В.В., Ефанов Д.В. Об использовании свойств кодов с суммированием по обнаружению монотонных ошибок в системах функционального контроля комбинационных схем	76
Черемисинова Л.Д. Логический синтез комбинационных КМОП схем с учетом рассеивания мощности	89
Ubar R. Fault Effect Reasoning in Digital Systems by Topological View on Low- and High-level Decision Diagrams	99
Сведения об авторах	114

CONTENTS

CONTROL OF DYNAMICAL SYSTEMS

Paraev Yu.I. Optimal control two-sector economy	4
Shilin A.A., Bukreev V.G. Research and optimal sliding mode control switching element covered feedback	12

DATA PROCESSING

Bekkerman E.A., Gortsev A.M. Heuristic algorithm for estimating the number of states of asynchronous MC-flow events	20
Keller Yu.A. The application of data clustering on the basis of Kohonen self-organizing maps in the process of selecting candidate wells for enhanced oil recovery methods	32
Kiturko O.M., Matalytski M.A. Asymptotic analysis incomes in a closed HM-network with a variable number of priority and no priority messages	38

INFORMATICS AND PROGRAMMING

Babanov A.M. Two modern approaches to semantic modeling – ORM and ERMM	46
Moiseev A.N., Politov A.M., Rakhimov V.V., Khomich M.O. Real-time collaborative diagram editor unified element presentation concept	57

DISCRETE FUNCTION AND AUTOMATION

Burdonov I.B., Kossatchev A.S. Graph learning by interacting automata	67
--	----

DESIGNING AND DIAGNOSTICS OF COMPUTER SYSTEMS

Sapozhnikov V.V., Sapozhnikov V.V., Efanov D.V. Implementation of sum code characteristics for unidirectional error detection in concurrent error detection (CED) systems of combinational circuits	76
Cheremisinova L.D. Lower-power logic synthesis of combinational CMOS circuits	89
Ubar R. Fault Effect Reasoning in Digital Systems by Topological View on Low- and High-level Decision Diagrams	99
Brief information about the authors	114

УПРАВЛЕНИЕ ДИНАМИЧЕСКИМИ СИСТЕМАМИ

УДК 658.512

Ю.И. Параев

ОПТИМАЛЬНОЕ УПРАВЛЕНИЕ ДВУХСЕКТОРНОЙ ЭКОНОМИКОЙ

Получено аналитическое решение динамической задачи оптимального управления двухсекторной экономикой на конечном интервале времени. Управление заключается в распределении произведённого продукта на накопление (инвестирование) и непроеизводственное потребление. Задача состоит в выборе такого управления, при котором обеспечивается максимум суммарного потребления за планируемый конечный интервал времени. Решение получено с помощью принципа максимума Понтрягина. Найдена магистраль – участок сбалансированного равновесного состояния экономики. Решены задачи оптимального выхода на магистраль и оптимального схода с магистрали. Во всех случаях управления оказываются кусочно-постоянными.

Ключевые слова: двухсекторная экономика; оптимальное управление; магистраль; принцип максимума Понтрягина; производственная функция; непроеизводственное потребление.

В модели двухсекторной экономики рассматриваются два сектора с различными технологиями производства продукции [1–4]. Обычно в одном из секторов производятся средства производства, а в другом – предметы потребления. Управление экономикой состоит в распределении произведенного продукта на инвестирование секторов экономики и на непроеизводственное накопление. Задача заключается в выборе такого управления, при котором обеспечивается максимум непроеизводственного накопления за планируемый конечный интервал времени. С помощью принципа максимума Понтрягина получено аналитическое решение задачи. Найдена магистраль – участок сбалансированного равновесного состояния экономики. Решены задачи оптимального выхода на магистраль и оптимального схода с магистрали. На всем интервале времени управления оказываются кусочно-постоянными.

1. Постановка задачи

Оба сектора экономики характеризуются следующими величинами: k_i – основной капитал, l_i – трудовые ресурсы и $Y_i(k_i, l_i)$ – производственные функции ($i = 1, 2$). Значение $Y_i = Y_i(k_i, l_i)$ есть валовой продукт, произведенный i -м сектором в единицу времени, т.е. $Y_i \Delta t$ есть валовой продукт, произведенный за время Δt . В результате получаем систему уравнений, описывающую поведение двухсекторной экономики:

$$\begin{aligned} \dot{k}_1 &= -\mu_1 k_1 + W_1, \quad k_1(0) = k_{10} \geq 0, \\ \dot{k}_2 &= -\mu_2 k_2 + W_2, \quad k_2(0) = k_{20} \geq 0, \\ \dot{C} &= \delta C + W_3, \quad C(0) = 0. \end{aligned} \quad (1)$$

Здесь $C(t)$ – непроеизводственное потребление, μ_i (>0) – коэффициенты амортизации, δ (>0) – норма дисконтирования, W_1, W_2, W_3 – доли произведенного валового продукта, направленные на инвестирование секторов экономики и на увеличение непроеизводственного потребления. Если считать, что все трудовые ресурсы постоянны и равны l_0 , то $l_1 = \vartheta l_0$ и $l_2 = (1 - \vartheta)l_0$, где ϑ – коэффициент перераспределения.

Управление двухсекторной экономикой заключается в выборе величин W_1 , W_2 , W_3 и ϑ .
Возможны два варианта распределения произведенного валового продукта:

1. W_1 , W_2 , W_3 – доли суммарного валового продукта, произведенного обоими секторами, т.е.

$$W_i = u_i Y \quad (i = 1, 2, 3),$$

где

$$Y = Y_1 + Y_2, \quad u_1 + u_2 + u_3 = 1, \quad 0 \leq u_i \leq 1, \quad u_3 = 1 - u_1 - u_2. \quad (2)$$

2. Валовой продукт, произведенный каждым сектором, распределяется раздельно, т.е.

$$W_i = u_{i1} Y_1 + u_{i2} Y_2 \quad (i = 1, 2, 3),$$

где

$$u_{1k} + u_{2k} + u_{3k} = 1, \quad 0 \leq u_{ik} \leq 1, \quad u_{3k} = 1 - u_{1k} - u_{2k}, \quad i = 1, 2, 3, \quad k = 1, 2.$$

Например, в модели, приписываемой К. Марксу, продукт, произведенный в 1-м секторе, идет на инвестирование 1-го и 2-го секторов. Продукт, произведенный во 2-м секторе, идет на инвестирование 2-го сектора и на увеличение непроизводственного потребления. В этом случае $u_{31} = u_{12} = 0$.

Далее будем рассматривать только 1-й вариант. При этом для простоты будем предполагать, что трудовые ресурсы постоянны и не перераспределяются между секторами.

В этом случае непроизводственное потребление на интервале времени $[0, T]$ равно

$$J = \int_0^T e^{\delta(T-t)} W_3(t) dt = \int_0^T e^{\delta(T-t)} (1 - u_1 - u_2) Y(t) dt. \quad (3)$$

Основная задача: в течение интервала времени $[0, T]$ найти такие управления u_i ($i = 1, 2, 3$) с учетом (1) и (2), при которых функционал (3) максимален.

Производственные функции выбираются в форме Кобба–Дугласа $Y_i(k_i, l_i) = A_i k_i^{\alpha_i} l_i^{\beta_i}$ ($i = 1, 2$).
Здесь A_i – масштаб темпа производства ($A_i > 0$), α_i – коэффициент эластичности по основным фондам, β_i – коэффициент эластичности по трудовым ресурсам, причем $\alpha_i + \beta_i = 1$, $\alpha_i, \beta_i > 0$, $\beta_i = 1 - \alpha_i$. Поскольку трудовые ресурсы предполагаются постоянными, то $Y_i(k_i, l_i) = A_i k_i^{\alpha_i}$ на всем интервале времени.

2. Применение принципа максимума Понтрягина

При применении принципа максимума Понтрягина сначала на основании (1) и (3) составляется функция Гамильтона

$$\begin{aligned} H(k_1, k_2, p_1, p_2, u_1, u_2) = \\ = p_1(-\mu_1 k_1 + u_1 Y) + p_2(-\mu_2 k_2 + u_2 Y) - e^{\delta(T-t)} (1 - u_1 - u_2) Y, \end{aligned} \quad (4)$$

где $p_1(t)$ и $p_2(t)$ – вспомогательные переменные, которые удовлетворяют уравнениям

$$\dot{p}_i = -\frac{\partial H}{\partial k_i} = \mu_i p_i - Y_i' [u_1 p_1 + u_2 p_2 - e^{\delta(T-t)} (1 - u_1 - u_2)], \quad p_i(T) = 0. \quad (5)$$

Здесь $Y_i' = \frac{\partial Y_i}{\partial k_i} = \alpha_i A_i k_i^{\alpha_i - 1}$, $i = 1, 2$.

Удобно ввести новые вспомогательные переменные $q_i(t) = p_i(t) e^{-\delta(T-t)}$, $i = 1, 2$. Тогда вместо (4) получаем

$$\begin{aligned} H(k_1, k_2, p_1, p_2, u_1, u_2) = e^{-\delta(T-t)} H(k_1, k_2, q_1, q_2, u_1, u_2) = \\ = -q_1 \mu_1 k_1 - q_2 \mu_2 k_2 + Y [u_1 (q_1 + 1) + u_2 (q_2 + 1) - 1], \end{aligned} \quad (6)$$

где согласно (5) переменные $q_i(t)$ удовлетворяют уравнениям

$$\dot{q}_i = \lambda_i q_i - Y_i' [u_1 (q_1 + 1) + u_2 (q_2 + 1) - 1], \quad q_i(T) = 0. \quad (7)$$

Здесь $\lambda_i = \delta + \mu_i$, $i = 1, 2$.

Функция Гамильтона (6) линейна относительно u_1 и u_2 . Поэтому при $q_1(t)+1 \neq 0$ и $q_2(t)+1 \neq 0$ минимум этой функции достигается в угловых точках симплекса (2):

$$\begin{aligned} u_1(t) = u_2(t) = 0, u_3(t) = 1 & \quad \text{при } q_1(t) > -1 \text{ и } q_2(t) > -1, \\ u_1(t) = 1, u_2(t) = u_3(t) = 0 & \quad \text{при } q_1(t) < -1 \text{ и } q_2(t) > -1, \\ u_1(t) = 0, u_2(t) = 1, u_3(t) = 0 & \quad \text{при } q_1(t) > -1 \text{ и } q_2(t) < -1. \end{aligned}$$

3. Магистраль

Согласно общей теории в случае, когда функция Гамильтона линейна относительно управлений, возможно особое управление u_1 и u_2 . Особое управление [5] существует на некотором интервале времени $[t_1, t_2]$, если на всем этом интервале коэффициент при этом управлении в H тождественно равен нулю.

Согласно (6) особые управления $u_{1\text{ос}}$ и $u_{2\text{ос}}$ существуют на некотором интервале времени $[t_1, t_2]$, если $q_1(t) = q_2(t) \equiv -1$, т.е. функция H не зависит от управлений. Для этого необходимо равенство нулю первой и второй производных функций $q_1(t)$ и $q_2(t)$. Равенство нулю первых производных функций $q_1(t)$ и $q_2(t)$ при $q_1(t) = q_2(t) \equiv -1$ приводит к условиям

$$Y_i' = \lambda_i \quad \text{или} \quad k_{i\text{ос}}^{\beta_i} = \frac{\alpha_i A_i}{\lambda_i}, \quad i = 1, 2. \quad (8)$$

Равенство нулю вторых производных функций $q_1(t)$ и $q_2(t)$ при $q_1(t) = q_2(t) \equiv -1$ и $\dot{q}_1 = \dot{q}_2 = 0$ приводит к условиям $\dot{k}_1 = \dot{k}_2 = 0$. Отсюда

$$u_{i\text{ос}} = \frac{\mu_i k_{i\text{ос}}}{Y}, \quad i = 1, 2.$$

С учетом (8) можно показать, что $u_{1\text{ос}} + u_{2\text{ос}} < 1$, т.е. особое управление всегда удовлетворяет условию (2). Интервал времени $[t_1, t_2]$, в течение которого имеет место особое управление, соответствует участку сбалансированного равновесного состояния экономики, который называется *магистралью*. На этом интервале переменные k_1 и k_2 и управления u_1 и u_2 постоянны.

Непроизводственное потребление на интервале времени $[t_1, t_2]$ равно

$$J = \int_{t_1}^{t_2} e^{\delta(T-t)} (1 - u_{1\text{ос}} - u_{2\text{ос}}) Y(t) dt = (1 - u_{1\text{ос}} - u_{2\text{ос}}) Y \frac{e^{\delta(T-t_1)} - e^{\delta(T-t_2)}}{\delta},$$

где $Y = Y_1 + Y_2$ и согласно (8)

$$Y_i = A_i \left(\frac{\alpha_i A_i}{\lambda_i} \right)^{\frac{\alpha_i}{\beta_i}}, \quad i = 1, 2.$$

Возможен еще вариант, когда на некотором интервале $[t', t'']$ одно управление особое (частная магистраль), а другое – неособое. Из (2) следует, что в этом случае неособое управление обязательно равно нулю. Если на интервале $[t', t'']$ управление u_1 особое, то $u_2 = 0$. При этом $q_1(t) \equiv -1$, $q_2(t) > -1$ и переменная k_2 убывает. Если на интервале $[t', t'']$ управление u_2 особое, то $u_1 = 0$. При этом $q_2(t) \equiv -1$, $q_1(t) > -1$ и переменная k_1 убывает.

Предположим, что на интервале $[t_1, t_2]$ существует магистраль, т.е. оба управления особые. Тогда остается выбрать управления на интервале $[0, t_1]$ – выход на магистраль, и на интервале $[t_2, T]$ – сход с магистрали.

4. Выход на магистраль

На интервале времени $[0, t_1]$ получается краевая задача для уравнений (1) с граничными условиями

$$k_1(0) = k_{10}, k_2(0) = k_{20}, \quad (9)$$

$$k_1(t_1) = k_{1\text{oc}}, k_2(t_1) = k_{2\text{oc}}. \quad (10)$$

При этом длина интервала $[0, t_1]$ неизвестна. Она зависит от условий (9), (10) и параметров задачи.

Напомним, что при $u_i = 1$ функция $k_i(t)$ растет, а при $u_i = 0$ убывает. Среди всех решений уравнений (1) можно выделить такие, которые заканчиваются в конечных точках (10). Более того, можно выделить решения, которые заканчиваются в (10) при управлениях $u_1 = 1, u_2 = 0$ или $u_1 = 0, u_2 = 1$ на всем интервале $[0, t_1]$. Эти решения можно получить в результате интегрирования уравнений (1) в обратном времени с граничными условиями (10). В результате получаются нижние и верхние траектории: $N_1(t)$ и $B_2(t)$ в случае, когда $u_1 = 1, u_2 = 0$, и $N_2(t)$ и $B_1(t)$ в случае, когда $u_1 = 0, u_2 = 1$. При этом все другие решения лежат между этими кривыми, т.е.

$$N_i(t) \leq k_i(t) \leq B_i(t) = k_{i0c} e^{\mu_i(t_1-t)}, \quad i = 1, 2.$$

По аналогии с задачей оптимального быстродействия эти кривые назовем *линиями переключения*.

Очевидно, что в конце интервала $[0, t_1]$ система должна двигаться по этим линиям, чтобы попасть в точку (10). Поэтому получается следующее решение. Интервал $[0, t_1]$ разбивается на две части $[0, t']$ и $[t', t_1]$. Далее возможен два варианта.

А. Управления выбираются в виде

$$u_1 = \begin{cases} 1 & \text{при } 0 < t < t', \\ 0 & \text{при } t' < t < t_1 \end{cases} \quad \text{и} \quad u_2 = \begin{cases} 0 & \text{при } 0 < t < t', \\ 1 & \text{при } t' < t < t_1. \end{cases} \quad (11)$$

В результате получаем решение, представленное на рис. 1, а. При этом

$$k_1(t') = X_{11}(0, t'; k_{10}, k_{20}), k_2(t') = X_{20}(0, t'; k_{20}) = k_{20} e^{-\mu_2 t'},$$

$$k_1(t_1) = X_{10}(t', t_1; k_1(t')) = k_1(t') e^{-\mu_1(t_1-t')} = k_{1\text{oc}},$$

$$k_2(t_1) = X_{21}(t', t_1; k_1(t'), k_2(t')) = k_{2\text{oc}}.$$

Здесь $X_{i1}(s, t; k_1(s), k_2(s))$ – решение i -го уравнения (1) на интервале времени (s, t) при $u_i = 1$; $X_{i0}(s, t; k_i(s)) = k_i(s) e^{-\mu_i(t-s)}$ – решение i -го уравнения (1) на интервале времени (s, t) при $u_i = 0$.

Исключая из этих уравнений $k_1(t')$ и $k_2(t')$, получаем два уравнения для нахождения t' и t_1 .

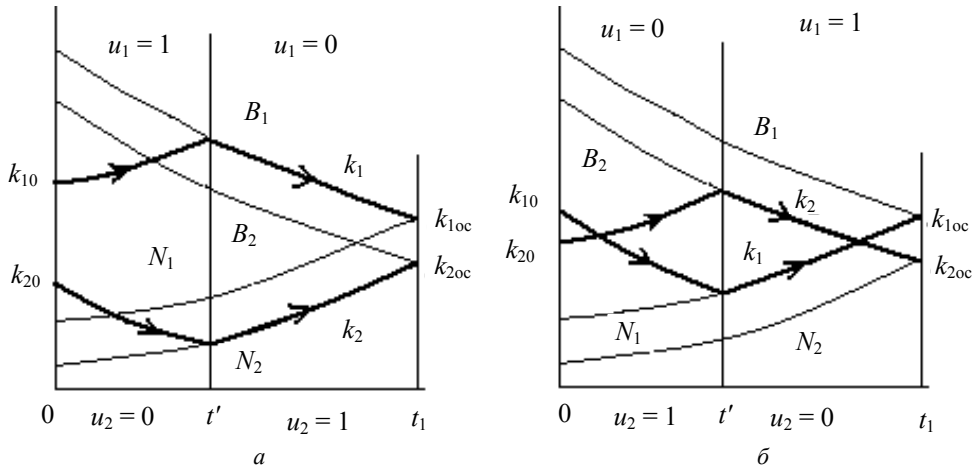


Рис. 1. Выход на магистраль

Б. Управления выбираются в виде

$$u_1 = \begin{cases} 0 & \text{при } 0 < t < t', \\ 1 & \text{при } t' < t < t_1 \end{cases} \quad \text{и} \quad u_2 = \begin{cases} 1 & \text{при } 0 < t < t', \\ 0 & \text{при } t' < t < t_1. \end{cases} \quad (12)$$

В результате получаем решение, представленное на рис. 1, б. При этом

$$\begin{aligned} k_1(t') &= X_{11}(0, t'; k_{10}, k_{20}), \quad k_2(t') = X_{20}(0, t'; k_{20}) = k_{20}e^{-\mu_2 t'}, \\ k_1(t_1) &= X_{10}(t', t_1; k_1(t')) = k_1(t')e^{-\mu_1(t_1-t')} = k_{10c}, \\ k_2(t_1) &= X_{21}(t', t_1; k_1(t'), k_2(t')) = k_{20c}. \end{aligned}$$

Исключая из этих уравнений $k_1(t')$ и $k_2(t')$, получаем два уравнения для нахождения t' и t_1 .

Таким образом, выход на магистраль осуществляется за счет релейного управления с одним моментом переключения. При управлениях (11) или (12) $u_3 = 0$, т.е. накоплений нет и нет оптимизационной задачи. Поэтому переменные $q_1(t)$ и $q_2(t)$ нас не интересуют. Из решений А и Б следует выбрать то, при котором значение t_1 наименьшее. Это связано с тем, что на интервале $[0, t_1]$ нет накоплений и его следует взять максимально коротким.

5. Сход с общей магистральной

На интервале времени $[t_2, T]$ получается краевая задача для уравнений (1) и (7) с граничными условиями

$$q_1(t_2) = q_2(t_2) = -1, \quad k_1(t_2) = k_{10c}, \quad k_2(t_2) = k_{20c}, \quad (13)$$

$$q_1(T) = q_2(T) = 0. \quad (14)$$

При этом длина интервала $[t_2, T]$ неизвестна.

Пусть $u_1 = u_2 = 0$. Тогда из (1) получаем, что на интервале $[t_2, T]$

$$k_i(t) = k_{i0c}e^{-\mu_i(t-t_2)}, \quad i = 1, 2,$$

и с учетом (8) $Y_i' = \alpha_i A_i k_i^{\alpha_i-1} = \lambda_i e^{\beta_i \mu_i(t-t_2)}$. Подставляя последнее в (7), получаем

$$\dot{q}_i = \lambda_i q_i + \lambda_i e^{\beta_i \mu_i(t-t_2)} = \lambda_i (q_i + e^{\beta_i \mu_i(t-t_2)}), \quad i = 1, 2.$$

Поскольку на интервале $[t_2, T]$ $-1 \leq q_i \leq 0$, а $e^{\beta_i \mu_i(t-t_2)} > 1$, то $\dot{q}_i > 0$ и эти функции растут. Решение этих уравнений с учетом (13) следующее:

$$\begin{aligned} q_i(t) &= -e^{\lambda_i(t-t_2)} + \lambda_i \int_{t_2}^t e^{\lambda_i(t-\tau)} e^{\beta_i \mu_i(\tau-t_2)} d\tau = \\ &= -e^{\lambda_i(t-t_2)} + \frac{\delta + \mu_i}{\delta + \alpha_i \mu_i} \left(e^{\lambda_i(t-t_2)} - e^{\beta_i \mu_i(t-t_2)} \right), \end{aligned} \quad (15)$$

$i = 1, 2$. Поскольку параметры задачи разные, то функции $q_1(t)$ и $q_2(t)$ растут по разному и достигают значения 0 в разные моменты времени. Поэтому значения t_2 должны быть разными.

Подставляя (15) в (14), получаем

$$e^{\lambda_i(T-t_{2i})} = \frac{\delta + \mu_i}{\delta + \alpha_i \mu_i} \left(e^{\lambda_i(T-t_{2i})} - e^{\beta_i \mu_i(T-t_{2i})} \right), \quad i = 1, 2. \quad (16)$$

Решение этого уравнения проводится следующим образом. Из (16) следует

$$e^{\beta_i \mu_i(T-t_{2i})} = \frac{\beta_i \mu_i}{\delta + \mu_i} e^{\lambda_i(T-t_{2i})}, \quad i = 1, 2.$$

Логарифмируя это выражение, получаем

$$\beta_i \mu_i (T - t_{2i}) = \lambda_i (T - t_{2i}) + \ln \left(\frac{\beta_i \mu_i}{\delta + \alpha_i \mu_i} \right)$$

и окончательно

$$T - t_{2i} = \frac{1}{\delta + \alpha_i \mu_i} \ln \left(\frac{\delta + \alpha_i \mu_i}{\beta_i \mu_i} \right), \quad i = 1, 2.$$

Пусть функция $q_1(t)$ растет медленнее, чем $q_2(t)$, т.е. $T - t_{21} > T - t_{22}$. В этом случае должно быть следующее решение задачи. Полагаем $t_{21} = t_2$. На интервале $[t_2, T]$ управление $u_1 = 0$, а управление u_2 строится в виде

$$u_2 = \begin{cases} u_{2oc} & \text{при } t_2 < t < t_{22}, \\ 0 & \text{при } t_{22} < t < T. \end{cases}$$

При таком управлении на интервале $[t_2, t_{22}]$ функция q_2 постоянна и равна -1 . На интервале $[t_{22}, T]$ функция q_2 переходит из значения $q_2(t_{22}) = -1$ в значения $q_2(T) = 0$ (рис. 2).

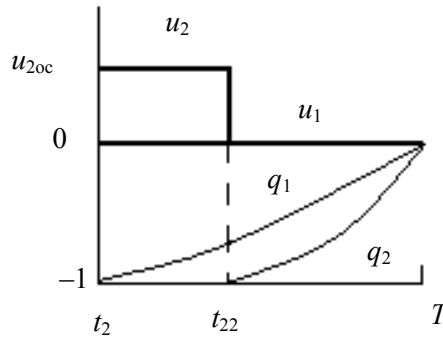


Рис. 2. Сход с магистрали

Таким образом, при сходе с магистрали одно управление равно нулю, а другое является релейным. Непроизводительное потребление на интервале времени $[t_2, t_{22}]$ равно

$$J = \int_{t_2}^{t_{22}} e^{\delta(T-t)} (1 - u_{2oc}) \left(A_1 k_{1oc}^{\alpha_1} e^{-\alpha_1 \mu_1 (t-t_2)} + A_2 k_{2oc}^{\alpha_2} \right) dt,$$

а на интервале $[t_2, t_{22}]$ равно

$$J = \int_{t_{22}}^T e^{\delta(T-t)} \left(A_1 k_{1oc}^{\alpha_1} e^{-\alpha_1 \mu_1 (t-t_2)} + A_2 k_{2oc}^{\alpha_2} e^{-\alpha_2 \mu_2 (t-t_{22})} \right) dt.$$

Эти интегралы легко вычисляются. Аналогичное решение получается, когда функция $q_1(t)$ растет быстрее, чем $q_2(t)$.

Заключение

Решена задача оптимального управления двухсекторной экономикой на конечном интервале времени. На всем интервале времени управления оказываются кусочно-постоянными. Выделяются три временных интервала: выход на магистраль, магистраль и сход с магистрали. Выход на магистраль осуществляется за счет релейного управления с одним моментом переключения. При этом на этом интервале непроизводительных накоплений нет. На магистрали капитал каждого сектора и управления постоянны. При сходе с магистрали произведенный продукт инвестируется только в один сектор экономики в зависимости от параметров задачи. На основании полученных результатов мож-

но сформулировать необходимые условия существования оптимального управления в рассматриваемой задаче, т.е. ограничения на длину интервала $[0, T]$, начальные условия и параметры модели. Эти условия связаны с необходимостью выполнения неравенств $0 \leq t_1 \leq t_2 < T$. В частности, требуется, чтобы интервал $[0, T]$ был достаточно большим.

ЛИТЕРАТУРА

1. Ашманов С.А. Введение в математическую экономику. М. : Наука, 1984.
2. Колмаев В.А. Математическая экономика. М. : ЮНИТИ, 1998.
3. Лобанов С.Г. К теории оптимального экономического роста // Экономический журнал ВШЭ. 1999. № 1. С. 28–41.
4. Минюк С.А., Ровба Е.А., Кузьмич К.К. Математические методы и модели в экономике. Минск : Тетра-Системс, 2002.
5. Габасов Р., Кириллова Ф.М. Особые оптимальные управления. М. : Наука, 1973.

Параев Юрий Иванович, д-р техн. наук, профессор. E-mail: paraev@mail.ru
Томский государственный университет

Поступила в редакцию 5 мая 2014 г.

Paraev Yury I. (Tomsk State University, Tomsk, Russian Federation).

Optimal control by the two-sector economy.

Keywords: two-sector economy; optimum control; highway; Pontryagin maximum principle; production function; non-productive consumption.

The analytical solution of a dynamic problem of optimum control by the two-sector economy on a finite interval of time is received. Control consists in distribution of the produced product into accumulation (investment) and non-productive consumption. The problem consists in selecting such a control, at which, for a planned finite interval of time, the maximum of non-productive consumption is provided.

Both sectors of economy are characterized by the values: k_i is the fixed capital, l_i is the labor resources, and Y_i are the Cobb-Douglas production functions ($i = 1, 2$). The Y_i is the gross product produced by the i -th sector at a time, i.e. $Y_i \Delta t$ is gross product produced during Δt . As a result, we obtain a system of equations describing the behavior of the two-sector economy

$$\begin{aligned}\dot{k}_1 &= -\mu_1 k_1 + W_1, \quad k_1(0) = k_{10} \geq 0, \\ \dot{k}_2 &= -\mu_2 k_2 + W_2, \quad k_2(0) = k_{20} \geq 0, \\ \dot{C} &= \delta C + W_3, \quad C(0) = 0.\end{aligned}$$

Here, $C(t)$ is non-productive consumption, $\mu_i (>0)$ is damping coefficients, $\delta (>0)$ is discount rate, W_1, W_2, W_3 are shares of gross product, aimed at investing sectors of the economy and an increase in non-productive consumption. Control by the two-sector economy consists in choosing values W_1, W_2, W_3 . In this paper, we consider such distributions of gross product when W_1, W_2, W_3 are shares of the total gross domestic product produced by both sectors, i.e.

$$W_i = u_i(Y_1 + Y_2) \quad (i=1, 2, 3),$$

where

$$u_1 + u_2 + u_3 = 1, \quad 0 \leq u_i \leq 1, \quad u_3 = 1 - u_1 - u_2.$$

In this case, the non-productive consumption in the interval $[0, T]$ is equal to

$$J = \int_0^T e^{\delta(T-t)} W_3(t) dt = \int_0^T e^{\delta(T-t)} (1 - u_1 - u_2) Y(t) dt.$$

In the interval $[0, T]$, the main problem is to find such controls u_1 and u_2 , subjected to the restrictions, under which this functional is maximized.

The problem solution is carried out by using the Pontryagin maximum principle. Since the Hamiltonian is linear with respect to the controls u_1 и u_2 , then, as shown, there are the special controls u_{1oc} and u_{2oc} when the capital of both sectors remains constant. The interval $[t_1, t_2]$, in which the special control takes place, corresponds to a site of the balanced equilibrium state of the economy, which is called as the highway.

As a result, in all cases the optimal control is piecewise and constant. The interval $[0, T]$ is broken into three parts: $[0, t_1]$ is an exit to the highway, $[t_1, t_2]$ is the highway, $[t_2, T]$ is a descent from the highway. The problems of the optimal exit to the highway and optimal descent from the highway are solved. The exit to the highway is carried out by the relay control with one switching point. At a descent from the highway, the produced product is invested only in one sector of economy depending on task parameters.

Also, obtained results allow to formulate necessary conditions of optimum control existence for our problem, i.e. restrictions on the length of a interval $[0, T]$, the entry conditions, and the model parameters. These conditions are connected with need of fulfillment of the inequalities $0 \leq t_1 \leq t_2 < T$.

REFERENCES

1. Ashmanov S. A. *Vvedenie v matematicheskuyu ekonomiku* [Introduction in mathematical economy]. Moscow: Nauka Publ., 1984. 293 p.
2. Kolemeyev V.A. *Matematicheskaya ekonomika* [Mathematical economy]. Moscow: YuNITI Publ., 1998. 399 p.
3. Lobanov S.G. K teorii optimal'nogo ekonomicheskogo rosta [To the theory of optimum economic growth]. *Ekonomicheskij zhurnal VShE – HSE Economic Journal*, 1999, no. 1. pp. 28-41.
4. Minyuk S.A., Rovba E.A., Kuzmich K.K. *Matematicheskie metody i modeli v ekonomike* [Mathematical methods and models in economy]. Minsk: Tetra-Sistems Publ., 2002. 429 p.
5. Gabasov R., Kirillova F.M. *Osobyie optimal'nye upravleniya* [Special optimum managements]. Moscow: Nauka Publ., 1973. 256 p.

УДК 681.5.01:62-50

А.А. Шилин, В.Г. Букреев

ИССЛЕДОВАНИЕ ОПТИМАЛЬНОГО И СКОЛЬЗЯЩЕГО РЕЖИМОВ УПРАВЛЕНИЯ С РЕЛЕЙНЫМ ЭЛЕМЕНТОМ, ОХВАЧЕННЫМ ОБРАТНОЙ СВЯЗЬЮ

Рассматривается метод синтеза оптимального управления линейным объектом второго порядка в системе с релейным элементом, охваченным обратной связью. Получено необходимое условие возникновения скользящего режима. Созданный алгоритм управления обладает робастным свойством при интервальных изменениях параметров объекта. Результаты подтверждаются математическим моделированием и экспериментальными исследованиями на действующей теплообменной системе промышленного типа.

Ключевые слова: релейный элемент с обратной связью; оптимальное управление по быстродействию; скользящий режим.

Статья посвящена развитию методов синтеза систем с релейным элементом, охваченным обратной связью, при решении задач конструирования оптимального управления различными техническими системами. Один из вариантов практического применения релейного регулятора для линеаризованных систем рассматривается в [1], где предлагается аппроксимация характеристик релейного элемента линейным уравнением. Работоспособность такого подхода подтверждается промышленной эксплуатацией распространенных аналоговых регуляторов Р-21, Р-29, РП-4, теория синтеза которых достаточно полно представлена в литературе [2, 3]. Использование многофункциональной нелинейной обратной связи в структуре релейного элемента промышленных регуляторов создает предпосылки повышения качества управления сложными динамическими объектами [4, 5].

Определенный теоретический интерес представляют системы с разрывным управлением, функционирующие в скользящем режиме и учитывающие изменение реальных характеристик релейного регулятора (С.В. Емельянов, В.И. Уткин, В.А. Уткин и др.). В классическом исполнении моменты переключения релейного элемента не являются изолированными, и количество переключений регулятора теоретически стремится к бесконечности. Уменьшение количества переключений возможно путем введения гистерезиса в релейную характеристику или запаздывания управляющего воздействия [6]. Другим эффективным способом уменьшения количества переключений является выбор пары управляющих сигналов $(-1, 0)$ или $(0, 1)$ вместо классической пары $(-1, 1)$ при условии, что фазовые траектории для данных пар сигналов направлены к траектории скольжения. Авторами реализован релейный метод управления в скользящем режиме без зоны нечувствительности [7], позволяющий получить минимальное количество переключений для объекта второго порядка:

$$\begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = -h(x) \cdot x_2 + g(x) \cdot u(t), \end{cases} \quad (1)$$

где функции $h(x) = k_o(x)/T_{ob}$, $g(x) = k_p(x)/T_{pr}$ являются положительными и определяются состоянием системы $k_o(x)$ и $k_p(x)$; T_{ob} и T_{pr} – постоянные времени объекта и исполнительного привода.

Такая модель часто применяется для объектов в теплоэнергетике с трехпозиционным электро-механическим клапаном в качестве исполнительного устройства и управлением $u \in (-1, 0, 1)$. Отметим особенность управления, где вместо функции $\text{sign}(x)$ используется скалярная функция, обладающая свойством гистерезиса:

$$y = \eta(x, \delta) = \begin{cases} \text{sign}(x), & \text{если } |x| > \delta, \\ 0, & \text{если } (x \cdot x_{-1}) < 0, \\ y_{-1}, & \text{если } ((x \cdot x_{-1}) > 0) \wedge (|x| < \delta). \end{cases} \quad (2)$$

Значение y_{-1} определяется на предыдущем шаге вычисления, так как характеристика гистерезиса предполагает запоминание прошлого состояния. Длительность дискретизации вычисления должна быть значительно меньше постоянной времени объекта. Значение управления определяется выражением

$$u(t) = \eta(a_1 \cdot x_1(t) + x_2(t), \delta). \quad (3)$$

На рис. 1, *a* представлены фазовые траектории *a, b, c, d, e* для различных значений функции $g(x)$, и для выделения характера управления в виде пары сигналов (0,1) введён фрагмент траекторий *a* и *e*.

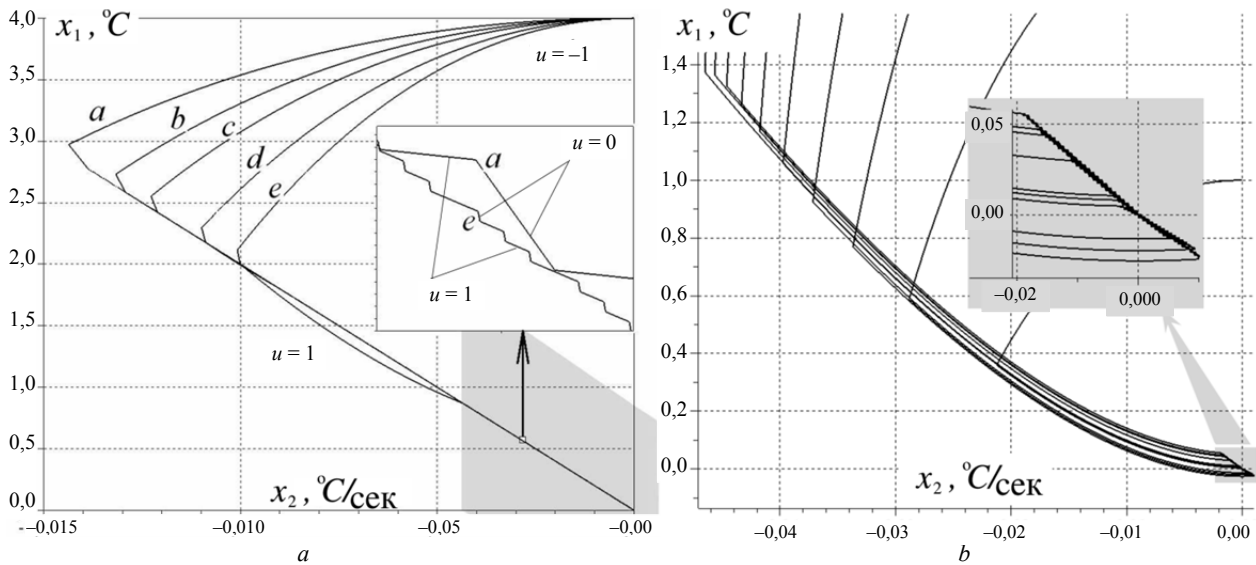


Рис. 1. Результаты моделирования в скользящем режиме с подстройкой a_1

Отметим, что фазовая траектория *e* объекта имеет свойство притяжения к траектории скольжения при втором ее пересечении. Следовательно, возможно предположение, что для заданного положения объекта (1) в фазовой плоскости $(x_1(t_0), x_2(t_0))$ существует такое значение параметра траектории скольжения a_1 в (3), при котором второе пересечение фазовой траектории объекта будет в малой окрестности точки равновесного состояния. В этом случае, при стремлении $\delta \rightarrow 0$, имеет место одно переключение на противоположный знак управления вне окрестности точки равновесного состояния, что соответствует оптимальному управлению по быстродействию. Практический интерес представляют методы управления, обладающие робастными свойствами в скользящем режиме и оптимальными по быстродействию при некоторых стационарных условиях функционирования объекта. Так, в работе [8] рассматриваются алгоритмы, совмещающие скользящий режим с оптимальным управлением в форме отрицательной обратной связи, а в работе [9] получена аппроксимация параметра $a_1(x_1(t_0), x_2(t_0), h, g)$ для стационарного линейного объекта. Результаты моделирования, представленные на рис. 1, *b*, подтверждают существование параметра траектории скольжения a_1 для стационарных условий, где g и h известны и неизменны, что обеспечивает оптимальную по быстродействию траекторию движения объекта из заданных точек на фазовой плоскости. Таким образом, есть основание полагать, что управление (3) с вычислением параметра $a_1(x_1(t_0), x_2(t_0))$ для стационарного объекта гарантирует оптимальное управление по быстродействию. В остальных же случаях, когда выполняется неравенство $g(x) > g$ или $h(x) > h$, следует скользящий режим. В частности, на рис. 1, *b* выделена окрестность в точке равновесия, где наблюдается скользящий режим, вызванный погрешностью вычисления параметра a_1 . Следует отметить, что реализация управления (3) на микроконтроллерах с

малой разрядностью аналого-цифрового преобразователя (АЦП) вызывает большие затруднения при необходимости оценки производной измеренного значения $x_1(t)$.

1. Постановка задачи

Рассмотрим замкнутую систему управления для объекта (1) с регулятором на релейном элементе и обратной связью, структуру которого можно представить в виде уравнений

$$\begin{cases} u = \eta(x_3 - x_1, \delta), \\ \dot{x}_3 = \begin{cases} -a_3 \cdot x_3, & \text{если } u = 0, \\ -k_3 \cdot u, & \text{если } u \neq 0, \end{cases} \end{cases} \quad (4)$$

где дополнительная переменная x_3 описывает динамические свойства обратной связи релейного элемента с учетом положительности значений параметров a_3 и k_3 . Синтез управления не предполагает использование переменной x_2 , следовательно, не требуется оценка производной от измеренного значения x_1 . Таким образом, в системе (4) наблюдается контур с отрицательной обратной связью, где организуется стремление переменной x_3 к переменной x_1 при активном управлении $u \neq 0$. Очевидно, этот контур будет устойчивым при любых $k_3 > 0$, так как имеет первый порядок и при отсутствии управления $u = 0$ переменная x_3 стремится к нулю. Параметры модели объекта и звена обратной связи в общем случае могут быть нелинейными и нестационарными, представленными соответствующими функциями $h(x, t)$, $g(x, t)$, $a_3(x, t)$, $k_3(x, t)$. Рассмотрим возможность реализации для объекта (1) с управлением (4) скользящего режима, а также оптимального управления по быстродействию в случае линеаризованного стационарного объекта.

2. О существовании скользящего режима

Утверждение. Для замкнутой системы (1) и (4) возможно существование траектории скольжения вида

$$s = a_s(t, x(t_0)) \cdot x_1 + x_2 = 0, \quad (5)$$

где $a_s(t, x(t_0)) > 0 \forall t \in (t \geq t_0)$, если выполняются необходимые условия

$$\begin{cases} x_1(t_0) = x_3(t_0), \\ a_3 \geq a_s(t). \end{cases} \quad (6)$$

Доказательство. Предположим, что $\delta > \sigma$, где σ – бесконечно малое положительное число, но не равное нулю. Введем функцию

$$q(t) = \frac{T_{(u \neq 0)}}{T_{(u \neq 0)} + T_{(u = 0)}}, \quad (7)$$

которая представляет собой скважность активного управления при заданных временных отрезках, соответствующих одному периоду автоколебания, вызванного наличием гистерезиса со значением δ . Предел этой функции $\lim_{\delta \rightarrow \sigma} q(t) \rightarrow r(t)$ стремится к конечному числу r , которое принимает значения в диапазоне $(0, 1)$ и может рассматриваться в качестве эквивалентного управления [6]. Тогда модель замкнутой системы для $x_1(t_0) > 0$ можно записать в виде дифференциальных уравнений

$$\begin{cases} \dot{x}_1 = x_2 = -a_s(t) \cdot x_1, \\ \dot{x}_2 = -h(x) \cdot x_2 + g(x) \cdot q(x), \\ \dot{x}_3 = -a_3 \cdot (1 - q(t)) \cdot x_1 + k_3 \cdot q(t). \end{cases} \quad (8)$$

Дифференциальные уравнения (8) при заданных начальных значениях фазовых переменных $x_3(t_0) = x_1(t_0) = c_1, x_2(t_0) = a_s(t_0) \cdot c_2$ имеют единственное решение относительно функции $a_s(t)$ на основе предположения, что функция $q(t)$ является эквивалентным управлением (4), которое обеспечивает выполнение условия режима скольжения $|x_3 - x_1| < \sigma$. Следовательно, при бесконечно малом σ и равенстве $x_1 = x_3$ из уравнения $a_s(t) \cdot x_1 = a_3 \cdot (1 - q(t)) \cdot x_1 - k_3 \cdot q(t)$ можно получить условие

$$q(t) = \frac{a_3 - a_s(t)}{a_3 + k_3 / x_1} \leq 1,$$

где решение относительно $x_1(t)$ имеет положительное значение на всей траектории скольжения. Аналогичным образом для условия $x_1(t_0) < 0$ можно записать дифференциальные уравнения замкнутой системы в виде

$$\begin{cases} \dot{x}_1 = x_2 = -a_s(t) \cdot x_1, \\ \dot{x}_2 = -h(x) \cdot x_2 - g(x) \cdot q(x), \\ \dot{x}_3 = -a_3 \cdot (1 - q(t)) \cdot x_1 - k_3 \cdot q(t). \end{cases} \quad (9)$$

Решение системы дифференциальных уравнений (9) позволяет записать

$$q(t) = \frac{a_3 - a_s(t)}{a_3 - k_3 / x_1} \leq 1,$$

где переменная состояния $x_1(t)$ имеет отрицательные значения по всей траектории скольжения. Таким образом, для рассматриваемых случаев можно записать равенство

$$q(t) = \frac{a_3 - a_s(t)}{a_3 + k_3 / |x_1|} \leq 1.$$

Следовательно, выполнение условия (6) предполагает положительно определенные значения $q(t)$ на интервале $[0 \dots 1]$. Отметим, что выбор значения a_3 определяется условиями существования классического скользящего режима [6], и утверждение (6) свидетельствует о том, что скользящий режим (5) в системе (4) будет косвенно определяться параметрами объекта. Следовательно, о робастных свойствах системы можно говорить только с точки зрения асимптотической устойчивости.

3. Реализация оптимального управления по быстродействию

Апробацию предложенного алгоритма оптимального управления рассмотрим на основе стационарного линейного объекта (1) при условиях $g(x) = g, h(x) = h$. Предположим, что до момента появления возмущающего воздействия система была установлена в нулевой точке равновесного состояния. При появлении возмущений в системе переменная x_1 принимает некоторое положительное значение $x_1(t_0) = C_1$ и в соответствии с уравнениями (4) управление будет равно $u = -1$. Решение дифференциального уравнения замкнутой системы определяется выражениями (1), (4) и отражает встречное движение переменных x_1 и x_3 до момента переключения знака управления, в результате чего переменная x_2 становится равной некоторому значению. Предположим, что параметр k_3 подобран таким образом, что в первоначальной точке переключения управления выполняется неравенство $|x_2| > k_3$. Это означает, что после первого переключения управляющего сигнала в системе не будет возникать скользящий режим и на определенном интервале времени выполняется равенство $u = +1$. Повторное достижение фазового состояния системы траектории скольжения в точке равновесия является условием завершения переходного процесса. Следовательно, актуальной задачей является определение значения параметра k_3 , обеспечивающее оптимальное управление по быстродействию. Рассмотрим оптимальное управление на примере модели объекта (1), полученное на основе принципа максимума Понтрягина. Переходный процесс представлен на рис. 2, а.

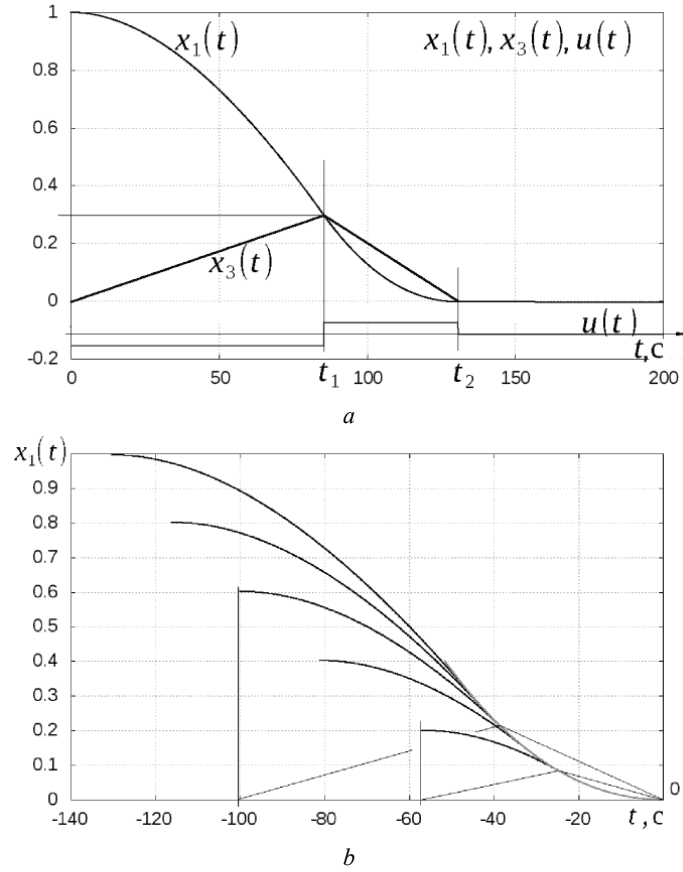


Рис. 2. Переходный процесс переменных состояния в системе с оптимальным регулятором

Переходный процесс переменной x_3 , определяемый вторым дифференциальным уравнением с управлением (4), организуется таким образом, чтобы момент переключения t_1 соответствовал первому уравнению (4). При условии $u(t) \neq 0$ коэффициент k_3 характеризует скорость изменения переменной x_3 , которая функционально зависит от знака сигнала оптимального управления, например:

$$\begin{aligned} k_3(u = -1) &= 0,29 / 85 = 3,4 \cdot 10^{-3} (1/c), \\ k_3(u = +1) &= 0,29 / 43 = 6,7 \cdot 10^{-3} (1/c). \end{aligned}$$

При моделировании переходных процессов в функции от других начальных условий $x_1(t_0) \in (0, 2; 0, 4; 0, 6; 0, 8; 1)$ обнаружилось, что скорость изменения переменной x_3 будет определяться знаком произведения $\text{sign}(x_1(t) \cdot u(t))$ и начальными значениями $x_1(t_0) = C_1$. В этом случае уравнения регулятора (4) принимают вид

$$\begin{cases} u = \eta(x_3 - x_1, \delta), \\ \dot{x}_3 = \begin{cases} -a_3 \cdot x_3, & \text{если } u = 0, \\ -u \cdot (T_{p1}(C_1))^{-1}, & \text{если } \text{sign}(u \cdot x_1) < 0, \\ -u \cdot (T_{p2}(C_1))^{-1}, & \text{если } \text{sign}(u \cdot x_1) > 0. \end{cases} \end{cases} \quad (10)$$

Причина возникновения такого явления и методы вычисления зависимостей (10) исследовалась авторами статьи в [9]. Получение в аналитической форме отображения параметров (g, h, C_1) объекта в параметры регулятора (T_{p1}, T_{p2}) остается открытой задачей, поэтому рассмотрим аппроксимацию данных функций, построенных на результатах частных решений, например методом Фельдбаума (см. рис. 2, b):

$$\begin{aligned}
C_{\max} &= k_{mx} \cdot g \cdot h^{-1}, \\
T_{p2} &= (k_{p20} + k_{p21} \cdot C_{\max} / C_1) \cdot g^{-1}, \\
T_{p1} &= (k_{p10} + k_{p11} \cdot C_1 / C_{\max}) \cdot T_{p2},
\end{aligned} \tag{11}$$

где k_{mx} , k_{p10} , k_{p11} , k_{p20} , k_{p21} – коэффициенты аппроксимации. Для экспериментального объекта с параметрами $g = 1,35^\circ\text{C}/\text{с}$, $h = 30^{-1} \text{с}^{-1}$ функции (11) принимают явный вид

$$\begin{aligned}
C_{\max} &= 4,49 \cdot 1,35 \cdot 30 = 7,68, \\
T_{p2} &= 2,66 + 7,68 / C_1, \\
T_{p1} &= (1,28 + 0,036 \cdot C_1) \cdot T_{p2}.
\end{aligned}$$

Значение коэффициента a_3 в оптимальном регуляторе для стационарного объекта непосредственно не влияет на переходный процесс, однако он определяет робастные свойства и режим скольжения по траектории (5) при изменении параметров объекта. Влияние запаздывания $\tau \approx 10 \text{с}$ по управлению $u(t - \tau)$ приводит к возникновению скользящего режима в конце переходного процесса (рис. 3, а). На рис. 3, б приведены результаты моделирования для объекта, где значение T_{pr} увеличено в три раза, что эквивалентно уменьшению коэффициента усиления в системе [10].

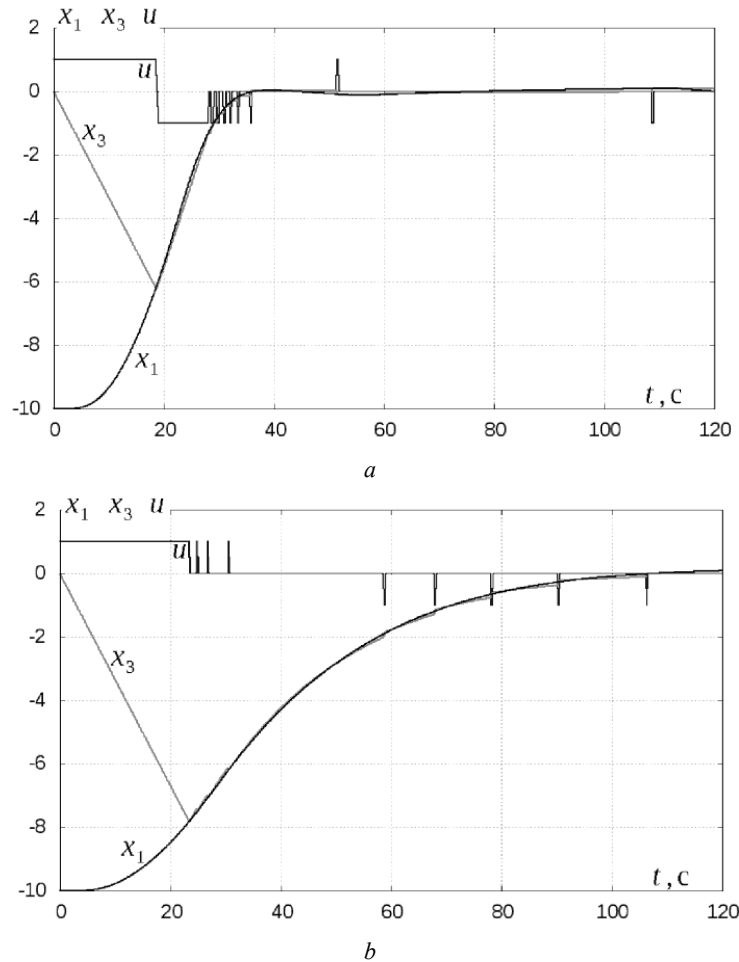


Рис. 3. Переходные процессы в скользящем режиме

В этом случае параметры регулирования не будут соответствовать оптимальным значениям, однако наблюдается скользящий режим и обеспечиваются робастные свойства предлагаемого метода синтеза регулятора.

Результаты моделирования стационарного объекта при начальном условии $|x_1(t_0)| = 10^\circ\text{C}$ представлены в виде переходных процессов на рис. 4, а.

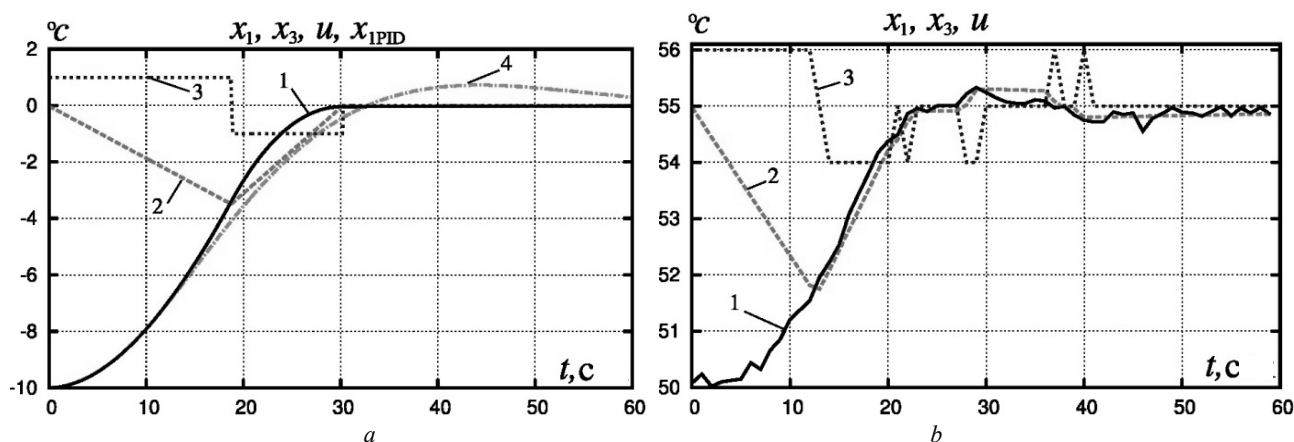


Рис. 4. Результаты моделирования и эксперимента оптимального управления:

1 – сплошная линия $x_1(t)$; 2 – штрихованная $x_3(t)$; 3 – пунктирная $u(t)$; 4 – штрих-пунктирная $x_{1PID}(t)$ – переходный процесс, полученный для ПИД-регулятора в форме отрицательной обратной связи по состоянию полностью наблюдаемой системы

В частности, для начального значения $x_1(t_0) = -10^\circ\text{C}$ управление соответствует принципу максимума Понтрягина, длительность переходного процесса составляет не более 30 с. Экспериментальные исследования выполнены на контроллере ВЭСТ-01М [11] действующей теплообменной системы, в котором реализованы метод управления (10) и средства сохранения усредненных значений временных рядов x_1 , x_3 , u . Результаты исследования отражают присутствие в измеряемой x_1 случайной составляющей и значительные ошибки дискретизации схемы измерения (рис. 4, б). При этом качество управления системой практически не изменилось и можно предположить, что метод синтеза оптимального управления (10) имеет необходимый ресурс для организации управления системами с запаздыванием и помехами в канале измерения.

Таким образом, можно сделать вывод, что отображение параметров (g, h, C_1) объекта управления в параметры (T_{p1}, T_{p2}) регулятора (10) по аппроксимирующим формулам (11) гарантирует в стационарных условиях оптимальный по быстродействию переходный процесс для начальных условий $x_1 = C_1, x_2 = x_3 = 0$, где $C_1 \in (0, C_{\max})$.

Заключение

Для рассмотренного метода синтеза управления с релейным элементом, охваченным обратной связью (10), эффективно реализуемого на микропроцессорных устройствах, предложено функциональное преобразование (11) параметров объекта в параметры регулятора, что позволяет обеспечить настройку замкнутой системы по экспериментально полученным переходным процессам. Для объекта с известными и стационарными параметрами обеспечивается оптимальное управление по быстродействию без вычисления кривой переключения в фазовом пространстве (см. рис. 4). Метод обладает робастными свойствами при изменении параметров объекта в определенных пределах (см. рис. 3, б) и не требует оценки производной измеренного сигнала. В случае транспортного запаздывания в системе рассматриваемый релейный регулятор формирует квазиоптимальное управление со скользящим режимом (рис. 3, а). Метод реализован на недорогом 8-разрядном процессоре с 10-разрядным АЦП на контроллере ВЭСТ-01М и успешно используется в регуляторах температуры теплообменных систем.

ЛИТЕРАТУРА

1. Стефани Е.П. Основы расчета настройки регуляторов теплоэнергетических процессов. М. : Энергия, 1972. 376 с.
2. Ключев А.С. и др. Проектирование систем автоматизации технологических процессов. М. : Энергоатомиздат, 1990. 464 с.
3. Новиков С.И. Оптимизация автоматических систем регулирования теплоэнергетического оборудования. Ч. 2 : Аналоговые регулирующие устройства и реализация АСР : учеб. пособие. Новосибирск : Изд-во НГТУ, 2006. 84 с.
4. Пат. РФ N 1798764, G 05 B 11/14, 1992 г.
5. Пат. RU N 2150726, C1 G05B11/14, 2000 г.
6. Уткин В.И. Скользящие режимы и их применение в системах с переменной структурой. М. : Наука, 1974. 240 с.
7. Шилин А.А., Букреев В.Г. Исследование трехпозиционного релейного регулятора температуры в скользящем режиме работы // Доклады ТУСУРа. 2012. № 1 (часть 2). С. 251–257.
8. Rong Xu, B.S. Optimal sliding mode control and stabilization of under-actuated systems: Dissertation PhD. The Ohio State University, 2007. 153 с.
9. Шилин А.А., Букреев В.Г. Динамическое определение траектории скольжения при релейном управлении нелинейным объектом // Проблемы управления. 2013. № 5. С. 22–28.
10. Шилин А.А., Букреев В.Г., Койков К.И. Математическая модель нелинейной теплообменной системы с запаздыванием // Приборы и системы. Управление, контроль, диагностика. 2013. № 6. С. 15–22.
11. НПО ВЭСТ Автоматический регулятор теплоснабжения ВЭСТ-01М. URL: <http://npowest.ru/6-instrukcii-po-ekspluatacii-priborov.html>

Шилин Александр Анатольевич, канд. техн. наук, доцент. E-mail: shilin@tpu.ru

Букреев Виктор Григорьевич, д-р техн. наук, профессор. E-mail: bukreev@tpu.ru

Томский политехнический университет

Поступила в редакцию 1 июня 2014 г.

Shilin Alexander A., Bukreev Viktor G. (Tomsk Polytechnic University, Tomsk, Russian Federation).

Research and optimal sliding mode control switching element covered feedback.

Keywords: relay element with feedback; speed optimal control; sliding mode.

This paper considers the organization of optimal and sliding modes in control system for non-stationary second order object with parameters that change on some interval. The main object of the control system is the three-position relay controller with feedback and defined area of the hysteresis. The article proves the optimal control problem for stationary objects on stage of achieving slip trajectory when there is a pair of control signals $(-1, 0)$ or $(0, 1)$.

The proposed functional transformation of the object parameters into the controller parameters allows setting up closed system using the experimental transients. The method provides time-optimal control without calculating the switching curve in the phase space for an object with known and stationary parameters. The method has robust properties when object parameters vary within certain limits and does not require estimates of the derivative of the measured signal. In case of transport delay in the system, the relay controller generates a quasi-optimal control with sliding mode. The necessary condition for the existence of a sliding mode indicates indirect dependence of sliding trajectory parameters from the object parameters and robust properties of the system in terms of asymptotic stability.

Time-optimal control is confirmed by modeling and experiments on the existing heating system of industrial type. The presence of delay and object transfer coefficient triple modification lead to the sliding mode with a pair of signals $(-1, 0)$ or $(0, 1)$.

REFERENCES

1. Stefani Ye.P. *Osnovy rascheta nastroyki regulyatorov teploenergeticheskikh protsessov* [Basis of calculation of heat and power regulators setting processes]. Moscow: Energiya Publ., 1972. 376 p.
2. Klyuev A.S. *Proektirovanie sistem avtomatizatsii tekhnologicheskikh protsessov* [Design process automation systems]. Moscow: Energoatomizdat Publ., 1990. 464 p.
3. Novikov S.I. *Optimizatsiya avtomaticheskikh sistem regulirovaniya teploenergeticheskogo oborudovaniya. Ch. 2: Analogovye reguliruyushchie ustroystva i realizatsiya ASR* [Optimization of automatic control systems of thermal power equipment. Part 2. Analog regulating device and the ACP]. Novosibirsk: NSTU Publ., 2006. 84 p.
4. Patent RF N 1798764, G 05 B 11/14, 1992.
5. Patent RF N 2150726, C1 G05B11/14, 2000.
6. Utkin V.I. *Skol'zyashchie rezhimy i ikh primeneniye v sistemakh s peremennoy strukturoy* [Slide modes and their application in systems with variable structure]. Moscow: Nauka Publ., 1974. 240 p.
7. Shilin A.A., Bukreev V.G. Research of three-position relay control the temperature in the sliding mode. *Doklady TUSURa*, 2012, no. 1 (pt. 2), pp. 251-257. (In Russian).
8. Rong Xu.B.S. Optimal sliding mode control and stabilization of under-actuated systems: Dissertation PhD. The Ohio State University, 2007. 153 p.
9. Shilin A.A., Bukreev V.G. Dynamic determination of sliding trajectory in relay control of nonlinear object. *Problemy upravleniya – Control Science*, 2013, no. 5. pp. 22-28. (In Russian).
10. Shilin A.A., Bukreev V.G., Koykov K.I. Mathematical model of the nonlinear heat transfer system with delay. *Pribory i sistemy. Upravlenie, kontrol', diagnostika – Instruments and Systems: Monitoring, Control, and Diagnostics*, 2013, no. 6, pp. 15-22. (In Russian).
11. NPO VEST. *Automatic control of heat consumption VEST-01M*. Available at: <http://npowest.ru/6-instrukcii-po-ekspluatacii-priborov.html>

ОБРАБОТКА ИНФОРМАЦИИ

УДК 519.21

Е.Н. Беккерман, А.М. Горцев

ЭВРИСТИЧЕСКИЙ АЛГОРИТМ ОЦЕНКИ ЧИСЛА СОСТОЯНИЙ АСИНХРОННОГО МС-ПОТОКА СОБЫТИЙ

Изучается асинхронный МС-поток событий, являющийся одной из адекватных математических моделей информационных потоков сообщений в цифровых сетях интегрального обслуживания. Поток функционирует в условиях априорной неопределённости, когда информация о параметрах и числе состояний отсутствует. Приводится эвристический алгоритм оценки числа состояний потока, основанный на свойстве одноэкстремальности огибающей гистограммы оценок простейшего потока событий.

Ключевые слова: асинхронный МС-поток событий; состояния потока; оценка; интервал стационарности; плотность вероятностей; гистограмма.

Математические модели теории массового обслуживания широко применяются при описании реальных физических, технических, экономических и других процессов и систем. В связи с развитием компьютерной техники и информационных технологий появилась важная сфера приложений теории массового обслуживания – проектирование и создание цифровых сетей интегрального обслуживания (ЦСИО).

Условия функционирования реальных объектов и систем таковы, что если в отношении параметров обслуживающих устройств можно сказать, что они известны и с течением времени не меняются, то в отношении интенсивностей входящих потоков этого сказать во многих случаях нельзя. Более того, интенсивности входящих потоков обычно меняются со временем, часто эти изменения носят случайный характер, что приводит к рассмотрению математических моделей дважды стохастических потоков событий. По-видимому, статьи [1, 2] являются одними из первых работ в этом направлении, где дважды стохастический поток определяется как поток, интенсивность которого есть случайный процесс. С другой стороны, функционирование систем массового обслуживания (СМО) зависит от параметров и состояний входящих потоков. В подобных ситуациях наиболее рациональным является применение адаптивных СМО, которые в процессе функционирования оценивают неизвестные параметры либо состояния входящих потоков и изменяют дисциплину обслуживания в соответствии с полученными оценками [3].

Дважды стохастические потоки можно разделить на два класса: к первому классу относятся потоки, интенсивность которых есть непрерывный случайный процесс; ко второму классу – потоки, интенсивность которых есть кусочно-постоянный случайный процесс с конечным числом состояний. Подчеркнём, что потоки второго класса впервые введены в рассмотрение практически одновременно и независимо в 1979 г. в [4–6]. В [4, 5] введённые потоки названы МС (Markov Chain)-потоками; в [6] – MVP (Markov Versatile Processes)-потоками. С начала 1990-х гг. отечественные и зарубежные авторы называют в своих работах введённые потоки событий либо дважды стохастическими потоками, либо МАР-потоками, либо МС-потоками [7–13].

В свою очередь, в зависимости от того, каким образом происходит переход из состояния в состояние, МС-потоки событий можно разделить на три типа: 1) синхронные потоки [14, 15]; 2) асинхронные потоки [16, 17]; 3) полусинхронные потоки [18]. Здесь указаны ссылки на статьи, в

которых авторы впервые рассматривали МС-потоки событий в соответствии с приведённой классификацией. Наиболее полная литература по изучаемым типам МС-потоков событий приведена в [19].

Отметим, что синхронные, асинхронные и полусинхронные потоки возможно представить в виде моделей МАР-потоков событий первого либо второго порядков [20]. В [20] показано, что синхронный МС-поток является частным случаем МАР-потока первого порядка, асинхронный и полусинхронный МС-потоки являются частными случаями МАР-потока второго порядка. Как было отмечено выше, в реальных ситуациях интенсивность входящего потока событий изменяется со временем случайным образом, поэтому при реализации адаптивного управления СМО возникают, в частности, следующие задачи: 1) оценка состояний потока по наблюдениям за моментами наступления событий [21]; 2) оценка параметров потока по наблюдениям за моментами наступления событий [22].

В приведённой литературе [14–21] задачи по оценке состояний и параметров того или иного потока решаются в предположении, что число состояний потока известно и равно двум. В статьях [22–26] решены задачи по оценке параметров и состояний потоков событий, когда число состояний потока произвольно и конечно.

Однако на практике часто возникают ситуации, когда априорные данные о числе состояний потока отсутствуют. Вследствие этого для реализации адаптивного управления в СМО возникает задача об оценке числа состояний потока, которая решается в настоящей статье для асинхронного МС-потока событий (далее асинхронный поток либо просто поток).

1. Постановка задачи

Рассматривается асинхронный поток событий с произвольным конечным числом состояний. Интенсивность потока является кусочно-постоянным случайным процессом $\lambda(t)$ с n состояниями: $\lambda_1, \lambda_2, \dots, \lambda_n$ ($\lambda_1 > \lambda_2 > \dots > \lambda_n > 0$). Процесс (поток) $\lambda(t)$ в момент времени t находится в i -м состоянии, если $\lambda(t) = \lambda_i$ ($i = \overline{1, n}$). В течение времени пребывания в i -м состоянии поток ведет себя, как пуассоновский с параметром λ_i ($i = \overline{1, n}$). Длительность пребывания в i -м состоянии есть экспоненциально распределённая случайная величина с функцией распределения $F_i(\tau) = 1 - e^{-\alpha_{ii}\tau}$, где $\alpha_{ii} = - \sum_{j=1, j \neq i}^n \alpha_{ij}$ ($i = \overline{1, n}$); $\alpha_{ij} > 0$ ($i, j = \overline{1, n}, i \neq j$) – интенсивность перехода процесса $\lambda(t)$ из состояния i в состояние j . В сделанных предпосылках $\lambda(t)$ – скрытый марковский процесс. Блочная матрица инфинитезимальных коэффициентов примет вид

$$D = \left\| \begin{array}{cccc|cccc} -\left(\lambda_1 + \sum_{j=2}^n \alpha_{1j}\right) & \alpha_{12} & \alpha_{13} & \dots & \alpha_{1n} & \lambda_1 & 0 & \dots & 0 \\ \alpha_{21} & -\left(\lambda_2 + \sum_{j=1, j \neq 2}^n \alpha_{2j}\right) & \alpha_{23} & \dots & \alpha_{2n} & 0 & \lambda_2 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \alpha_{n1} & \alpha_{n2} & \alpha_{n3} & \dots & -\left(\lambda_n + \sum_{j=1}^{n-1} \alpha_{nj}\right) & 0 & 0 & \dots & \lambda_n \end{array} \right\| = \|D_0 \mid D_1\|.$$

Элементами матрицы D_1 являются интенсивности перехода процесса $\lambda(t)$ из i -го состояния в i -е ($i = \overline{1, n}$) с наступлением события. Недиагональные элементы матрицы D_0 – интенсивности переходов из состояния i в состояние j ($i, j = \overline{1, n}, i \neq j$) без наступления события. Диагональные элементы матрицы D_0 – интенсивности выхода процесса $\lambda(t)$ из своих состояний, взятые с противоположным знаком.

Отметим, что определённый таким образом поток событий называется асинхронным, так как переходы процесса $\lambda(t)$ из i -го состояния в j -е ($i, j = \overline{1, n}$, $i \neq j$) не привязаны к моментам наступления событий пуассоновских потоков. Асинхронный поток также называют MMP (Markov Modulated Poisson)-потоком [27].

Предполагается, что на вход адаптивной СМО поступает асинхронный поток событий с заданными, но неизвестными параметрами, т.е. значения параметров λ_i, α_{ij} ($i, j = \overline{1, n}$, $i \neq j$) и значение числа состояний потока n неизвестны. Процесс $\lambda(t)$ является принципиально ненаблюдаемым, наблюдению доступны только временные моменты наступления событий потока. Рассматривается стационарный режим функционирования наблюдаемого потока событий, поэтому переходными процессами на отрезке наблюдения $[t_1, t_k]$, $k = 2, 3, \dots$, где t_1 – начало наблюдений (момент наступления первого события потока), t_k – окончание наблюдений (момент наступления k -го события потока), пренебрегаем. Общая задача заключается в следующем: необходимо по наблюдениям (моментам наступления событий $t_1, \dots, t_k$) оценить значения неизвестных параметров λ_i, α_{ij} ($i, j = \overline{1, n}$, $i \neq j$), n .

Замечание 1. Существенным является значение числа n : если n велико, то на отрезке наблюдения $[t_1, t_k]$, $k = 2, 3, \dots$, возможны ситуации, когда процесс $\lambda(t)$ не перейдёт во все возможные состояния, и принятое решение об оценке $\hat{n}$ числа состояний процесса $\lambda(t)$ будет в принципе неверным.

Замечание 2. Если тем или иным статистическим алгоритмом построена оценка $\hat{n}$ числа n , то тогда для оценки параметров λ_i, α_{ij} ($i, j = \overline{1, n}$, $i \neq j$) можно использовать алгоритм, разработанный в [26].

2. Свойство плотности вероятностей оценки интенсивности простейшего потока

В соответствии с определением потока реализация процесса $\lambda(t)$ имеет временные участки, где $\lambda(t) = \lambda_i$, $i = \overline{1, n}$, – так называемые интервалы стационарности. Если алгоритм обработки моментов наступления событий $t_1, \dots, t_k$ каким-либо образом оценит число таких интервалов стационарности, то тем самым будет осуществлена оценка $\hat{n}$.

Предположим, что последовательность моментов наступления событий $t_1, t_2, \dots, t_k$, $k = 2, 3, \dots$, целиком принадлежит некоторому интервалу стационарности, для которого $\lambda(t) = \lambda_i$, $i = \overline{1, n}$. Положим для конкретности $\lambda(t) = \lambda$, $\lambda \in (\lambda_1, \dots, \lambda_n)$, при этом конкретное значение λ неизвестно. Отметим, что число событий на этом интервале может быть равным нулю, единице и т.д. ($k = 0, 1, 2, \dots$). Обозначим $\tau_k = t_{k+1} - t_k$, $k = 1, 2, \dots$, – значение длительности k -го интервала между соседними событиями пуассоновского потока интенсивности λ . Обозначим $T_k = \tau_1 + \tau_2 + \dots + \tau_k$ – значение длительности отрезка $[t_1, t_{k+1}]$, $k = 1, 2, \dots$. Тогда, так как длительность интервала (t_k, t_{k+1}) , $k = 1, 2, \dots$, между соседними событиями в пуассоновском потоке распределена по экспоненциальному закону с параметром λ , то плотность вероятностей суммы независимых экспоненциально распределённых случайных величин при фиксированном k есть [28]:

$$P(T_k) = \lambda \frac{(\lambda T_k)^{k-1}}{(k-1)!} e^{-\lambda T_k}, \quad T_k \geq 0. \quad (1)$$

Как известно [29], оценка максимального правдоподобия $\hat{\lambda} = k/T_k$ интенсивности λ является смещённой оценкой. Будем рассматривать исправленную оценку

$$\hat{\lambda}^{(k)} = (k-1)/T_k, \quad k = 2, 3, \dots \quad (2)$$

Тогда, с учётом (1), математическое ожидание оценки (2), её дисперсия и количество информации о параметре λ в выборке $(\tau_1, \tau_2, \dots, \tau_k)$, $k = 1, 2, \dots$, примут вид

$$M_{\hat{\lambda}} = \lambda, \quad D_{\hat{\lambda}} = \lambda^2 / (k - 2), \quad I_k(\lambda) = k / \lambda^2.$$

Таким образом, оценка (2) является сильно состоятельной, несмещённой и асимптотически эффективной.

Плотность вероятности оценки $\hat{\lambda}^{(k)}$ при фиксированном k , с учётом (1), примет вид

$$P(\hat{\lambda}^{(k)}) = \lambda^k \frac{(k-1)^{k-1}}{(k-2)!} [\hat{\lambda}^{(k)}]^{-(k+1)} e^{-\lambda \frac{k-1}{\hat{\lambda}^{(k)}}}, \quad k = 2, 3, \dots \quad (3)$$

Пусть на интервале стационарности, где $\lambda(t) = \lambda$, реализовалось $N+1$ событий пуассоновского потока интенсивности λ : $t_1, t_2, \dots, t_N, t_{N+1}$. Соответствующие значения длительностей интервалов между соседними событиями есть $\tau_j = t_{j+1} - t_j$, $j = \overline{1, N}$. Построим набор оценок для интенсивности λ с использованием (2) по следующему правилу:

$$\hat{\lambda}_j^{(k)} = (k-1) / \sum_{s=1}^k \tau_{j+s-1}, \quad k = \overline{2, N}; \quad j = \overline{1, N-k+1}. \quad (4)$$

Количество оценок набора (4) $N(N-1)/2$. Доля оценок $\hat{\lambda}^{(k)}$, полученных при фиксированном k ($k = \overline{2, N}$), в наборе (4) составляет величину

$$q_k^{(N)} = \frac{2(N-k+1)}{N(N-1)}, \quad k = \overline{2, N}; \quad \sum_{k=2}^N q_k^{(N)} = 1. \quad (5)$$

Вернёмся к формуле (2), которая даёт оценку интенсивности λ . Так как $\hat{\lambda}^{(k)}$ ($\hat{\lambda}^{(k)} > 0$), с одной стороны, является значением случайной величины (или просто случайной величиной) для каждого k ($k = \overline{2, N}$), с другой стороны, для каждого k определены плотности вероятностей (3), то в (3) число k можно рассматривать как параметр, принимающий значения $2, 3, \dots, N$. При этом величину $q_k^{(N)}$, определённую в (5), можно рассматривать как вероятность того, что параметр k принял некоторое значение из ряда чисел $2, 3, \dots, N$. Тогда в (3) переменную $\hat{\lambda}^{(k)}$ можно заменить на переменную $\hat{\lambda}$ и рассматривать смесь плотностей вероятностей (3) для $k = \overline{2, N}$:

$$P_N(\hat{\lambda}) = \sum_{k=2}^N q_k^{(N)} P_k(\hat{\lambda}), \quad P_k(\hat{\lambda}) = \lambda^k \frac{(k-1)^{k-1}}{(k-2)!} \left(\frac{1}{\hat{\lambda}}\right)^{k+1} e^{-\lambda \frac{k-1}{\hat{\lambda}}}, \quad (6)$$

где $q_k^{(N)}$ определена в (5). Таким образом, (6) является выпуклой линейной комбинацией функций $P_k(\hat{\lambda})$.

Исследуем функцию $P_N(\hat{\lambda})$ как функцию переменной $\hat{\lambda}$ ($\hat{\lambda} > 0$). Введём новую переменную $x = \lambda / \hat{\lambda}$ ($x > 0$). Из (6) вытекает

$$P_N(x) = \frac{1}{\lambda} \sum_{k=2}^N q_k^{(N)} \frac{(k-1)^{k-1}}{(k-2)!} x^{k+1} e^{-(k-1)x}, \quad x > 0. \quad (7)$$

Точки функции (7), подозрительные на экстремум, определяются уравнением

$$P_N'(x) = \frac{1}{\lambda} q_2^{(N)} x^2 e^{-x} \sum_{k=2}^N \frac{q_k^{(N)} (k-1)^{k-1}}{q_2^{(N)} (k-2)!} x^{k-2} e^{-(k-2)x} [(k+1) - (k-1)x] = 0, \quad (8)$$

где $(q_k^{(N)} / q_2^{(N)}) < 1$, $k = \overline{3, N}$. Тогда из (8) следует, что корни функции $P_N'(x)$ находятся из уравнения

$$F_N(x) = 0; \quad F_N(x) = \sum_{k=2}^N f_k^{(N)}(x); \quad (9)$$

$$f_k^{(N)}(x) = \frac{q_k^{(N)} (k-1)^{k-1}}{q_2^{(N)} (k-2)!} x^{k-2} e^{-(k-2)x} [(k+1) - (k-1)x].$$

Единственный ноль функции $f_k^{(N)}(x)$ есть $x_0^{(k)} = 1 + 2(k-1)^{-1}$, $k = \overline{2, N}$. Тогда имеем $F_N(x) > 0$ для $0 < x \leq 1 + 2(N-1)^{-1}$, $F_N(x) < 0$ для $x \geq 3$.

Таким образом, нули функции $F_N(x)$ (один или несколько) будут лежать в полуинтервале $(1 + 2(N-1)^{-1}; 3]$. Покажем, что функция $F_N(x)$ является строго монотонной на полуинтервале $(1 + 2(N-1)^{-1}; 3]$. Тогда $F_N(x)$ будет иметь на этом полуинтервале единственный ноль, а следовательно, функция (7) – единственный максимум. Производная $F_N'(x)$ выпишется в виде

$$F_N'(x) = \sum_{k=2}^N f_k^{(N)'}(x) = \sum_{k=2}^N \frac{q_k^{(N)} (k-1)^{k-1}}{q_2^{(N)} (k-2)!} x^{k-3} e^{-(k-2)x} z_k(x), \quad (10)$$

$$z_k(x) = (k-2)(k-1)x^2 - [(k-1)^2 + (k-2)(k+1)]x + (k-2)(k+1).$$

Отметим, что функция $z_k(x)$ в (10) определяет знак каждого слагаемого функции $F_N'(x)$; кроме того, $z_k(x)$ – выпуклая функция переменной x , её минимум достигается в точке $x_{0k} = 1 + (k-1)^{-1} + (2(k-1))^{-1}$, $k = \overline{3, N}$. Функция $z_k(x)$ отрицательна на интервале $(x_2^{(k)}, x_1^{(k)})$, $k = \overline{3, N}$, где

$$x_{1,2}^{(k)} = \left[1 + \frac{1}{k-1} + \frac{1}{2(k-2)} \right] \pm \sqrt{\frac{1}{k-2} \left(\frac{1}{4(k-2)} + \frac{k^2-2}{(k-1)^2} \right)}, \quad x_2^{(k)} < x_1^{(k)} < 4; \quad x_2^{(k)} < 1. \quad (11)$$

Значение минимума функции $z_k(x)$ в точке x_{0k} равно

$$z_k(x_{0k}) = - \left[(k+1) + (k-3)^2 (4(k-2)(k-1))^{-1} \right], \quad k = \overline{3, N}.$$

Итак, при исследовании функции $F_N(x)$ на количество нулей правомочно рассматривать значения x , принадлежащие отрезку $[1, 3]$.

Предварительно рассмотрим граничный случай $N = 2$. Из (9) получаем $F_2(x) = f_2^{(2)}(x) = 3 - x$, т.е. $x_0^{(2)} = 3$. Из (10) находим $F_2'(x) = -1$. Таким образом, для $N = 2$ функция $P_2(x)$, определённая в (7), имеет единственный максимум в точке $x_0^{(2)} = 3$. Тогда плотность вероятностей $P_2(\hat{\lambda})$, определённая в (6), имеет также единственный максимум в точке $\hat{\lambda}_0 = \lambda/3$.

Теорема. Функция $F_N(x)$, определённая в (9), имеет единственный ноль.

Доказательство. Докажем методом математической индукции. Пусть $N = 3$. Тогда $F_3(x) = (3-x) + 4x(2-x)e^{-x}$ и $F_3'(x) = -1 + 4(x^2 - 4x + 2)e^{-x}$. Отсюда получаем $F_3'(x) < 0$ для $2 - \sqrt{2} < x < 2 + \sqrt{2}$. Следовательно, функция $F_3(x)$ – строго монотонная (убывающая) на интервале $(1, x_1^{(3)}) = (1, 2 + \sqrt{2})$, где $x_1^{(3)}$ определена в (11), при этом $F_3(1) > 0$, $F_3(x_1^{(3)}) < 0$. Таким образом:

1) функция $F_3(x)$ имеет единственный ноль в точке $x_3^0 \in (1, x_1^{(3)})$:

$$1 + \frac{2}{k-1} \Big|_{k=3} < x_3^0 < \left[1 + \frac{1}{k-1} + \frac{1}{2(k-2)} \right] + \sqrt{\frac{1}{k-2} \left(\frac{1}{4(k-2)} + \frac{k^2-2}{(k-1)^2} \right)} \Big|_{k=3};$$

2) функция $F_3(x)$ не достигает нуля для $x > x_3^0$.

Пусть $N = 4$. Тогда $F_4(x) = (3-x) + (16/3)(2-x)xe^{-x} + (9/2)(5-3x)x^2e^{-2x}$, $F_4'(x) = -1 + (16/3)(x^2 - 4x + 2)e^{-x} + (9/2)(6x^2 - 19x + 10)xe^{-2x}$. Представим $F_4(x)$ и $F_4'(x)$ в виде

$$\begin{aligned} F_4(x) &= F_3(x) + \varphi_4(x); \quad F_4'(x) = F_3'(x) + \varphi_4'(x), \\ \varphi_4(x) &= (4/3)[(2-x) + (27/8)(5-3x)]xe^{-x}, \\ \varphi_4'(x) &= (4/3)[(x^2 - 4x + 2) + (27/8)(6x^2 - 19x + 10)xe^{-x}]e^{-x}. \end{aligned} \quad (12)$$

Тогда имеем $\varphi_4'(x) < 0$ для $1 \leq x \leq x_1^{(4)}$ ($x_1^{(4)} = 2,5$). Так как $F_3'(x) < 0$ для $1 \leq x \leq 2,5$, то $F_4'(x) < 0$ для $1 \leq x \leq 2,5$, т.е. $F_4(x)$ – строго монотонная (убывающая) на отрезке $[1; 2,5]$ функция. Так как $\varphi_4(x)$ – строго монотонная (убывающая) на отрезке $[1; 2,5]$ функция, при этом $\varphi_4(x) > 0$ для $1 \leq x \leq 5/3$, $\varphi_4(x) < 0$ для $x \geq 2,5$, то в единственной точке x_4^* ($5/3 < x_4^* < 2,5$) имеет место $\varphi_4(x = x_4^*) = 0$. Тогда $\varphi_4(x) < 0$ для $x > x_4^*$: а) $\varphi_4(x) < 0$ для $x_4^* < x \leq 2,5$ в силу строгой монотонности функции $\varphi_4(x)$ на полуинтервале $(x_4^*; 2,5]$; б) $\varphi_4(x) < 0$ для $x > 2,5$ в силу определения функции $\varphi_4(x)$ в (12). Наконец, в точке $x_1^{(4)}$ имеет место $F_4(x = x_1^{(4)}) < 0$. Следовательно, функция $F_4(x)$ – строго монотонная (убывающая) на интервале $(1, x_1^{(4)}) = (1; 2,5)$, где $x_1^{(4)}$ определена в (11); при этом $F_4(1) > 0$, $F_4(x_1^{(4)}) < 0$. Таким образом: 1) функция $F_4(x)$ имеет единственный ноль в точке $x_4^0 \in (1, x_1^{(4)})$:

$$1 + \frac{2}{k-1} \Big|_{k=4} < x_4^0 < \left[1 + \frac{1}{k-1} + \frac{1}{2(k-2)} \right] + \sqrt{\frac{1}{k-2} \left(\frac{1}{4(k-2)} + \frac{k^2-2}{(k-1)^2} \right)} \Big|_{k=4};$$

2) функция $F_4(x)$ не достигает нуля для $x > x_4^0$; 3) функция $\varphi_4(x)$ – строго монотонная (убывающая) для $1 \leq x \leq x_1^{(4)}$, имеющая единственный ноль в точке x_4^* ($5/3 < x_4^* < x_1^{(4)}$) и не достигающая нуля для $x > x_4^*$; $x_1^{(3)} > x_1^{(4)}$.

Рассмотрим произвольное $N + 1$. Имеем

$$F_{N+1}(x) = F_N(x) + \varphi_{N+1}(x), \quad F'_{N+1}(x) = F'_N(x) + \varphi'_{N+1}(x),$$

$$\varphi_{N+1}(x) = \frac{1}{N(N-1)} \sum_{k=3}^{N+1} \frac{(k-1)^{k-1}}{(k-3)!} x^{k-2} e^{-(k-2)x} [(k+1) - (k-1)x],$$

$$\varphi'_{N+1}(x) = \frac{1}{N(N-1)} \sum_{k=3}^{N+1} \frac{(k-1)^{k-1}}{(k-3)!} x^{k-3} e^{-(k-2)x} z_k(x), \quad N \geq 3,$$

где $z_k(x)$ определена в (10).

Сделаем предположение индукции для произвольного N :

1) функция $F_N(x)$ – строго монотонная (убывающая) на интервале $(1, x_1^{(N)})$, где $x_1^{(N)}$ определена в (11), при этом $F_N(1) > 0$, $F_N(x_1^{(N)}) < 0$;

2) функция $F_N(x)$ имеет единственный ноль в точке $x_N^0 \in (1, x_1^{(N)})$:

$$1 + \frac{2}{N-1} < x_N^0 < \left[1 + \frac{1}{N-1} + \frac{1}{2(N-2)} \right] + \sqrt{\frac{1}{N-2} \left(\frac{1}{4(N-2)} + \frac{N^2-2}{(N-1)^2} \right)}; \quad (13)$$

3) функция $F_N(x)$ не достигает нуля для $x > x_N^0$;

4) функция $\varphi_{N+1}(x)$ – строго монотонная (убывающая) для $1 \leq x \leq x_1^{(N+1)}$, имеющая единственный ноль в точке x_{N+1}^* ($1 + 2/N < x_{N+1}^* < x_1^{(N+1)}$) и не достигающая нуля для $x > x_{N+1}^*$; $x_1^{(N)} > x_1^{(N+1)}$.

Имеем $F_{N+1}(1) > 0$. Так как $\varphi'_{N+1}(x) < 0$ для $1 \leq x \leq x_1^{(N+1)}$ и $F'_N(x) < 0$ для $1 \leq x \leq x_1^{(N)}$ ($x_1^{(N)} > x_1^{(N+1)}$), то $F_{N+1}(x)$ строго монотонна (убывает) на отрезке $[1, x_1^{(N+1)}]$. В точке $x = x_1^{(N+1)}$ функция $F_{N+1}(x = x_1^{(N+1)})$ есть функция дискретного переменного N ($N \geq 3$). Численные расчёты показывают, что $F_{N+1}(x = x_1^{(N+1)}) < 0$ для $N = \overline{3, 10}$. Таким образом, на отрезке $[1, x_1^{(N+1)}]$ функция $F_{N+1}(x)$ имеет единственный ноль в точке x_{N+1}^0 ($1 < x_{N+1}^0 < x_1^{(N+1)}$). Имеем далее $\varphi_{N+1}(x)$ – строго монотонная (убывающая) функция для $1 \leq x \leq x_1^{(N+1)}$. По предположению индукции $\varphi_{N+1}(x)$ имеет единственный ноль в точке $x_{N+1}^* \in (1 + 2/N, x_1^{(N+1)})$ и не достигает нуля для $x > x_{N+1}^*$. Тогда имеем $F_{N+1}(x_{N+1}^*) = F_N(x_{N+1}^*)$, $F_{N+1}(x) > F_N(x)$ для $1 \leq x < x_{N+1}^*$, $F_{N+1}(x) < F_N(x)$ для $x > x_{N+1}^*$. По предположению индукции $F_N(x)$ не достигает нуля для $x > x_N^0$. Так как $x_{N+1}^* < x_{N+1}^0 < x_1^{(N)}$, то $F_{N+1}(x)$ не достигает нуля для $x > x_{N+1}^0$.

Чтобы замкнуть доказательство, нужно доказать, что функция $\varphi_{N+1}(x)$ не достигает нуля для $x > x_{N+1}^*$ и $1 < x_{N+1}^* < x_1^{(N+1)}$. Имеем $\varphi_{N+1}(x) = F_{N+1}(x) - F_N(x)$, $N \geq 3$. В точке $x = x_1^{(N+1)}$ выполняется $F_{N+1}(x_1^{(N+1)}) < 0$. Тогда: 1) если $F_N(x_1^{(N+1)}) < 0$ и $|F_{N+1}(x_1^{(N+1)})| > -F_N(x_1^{(N+1)})$, то $\varphi_{N+1}(x_1^{(N+1)}) < 0$ и $\varphi_{N+1}(x)$ имеет единственный ноль $x_{N+1}^* \in (1, x_1^{(N+1)})$; 2) если $F_N(x_1^{(N+1)}) < 0$ и $|F_{N+1}(x_1^{(N+1)})| < -F_N(x_1^{(N+1)})$,

$< -F_N(x_1^{(N+1)})$, то $\varphi_{N+1}(x_1^{(N+1)}) > 0$, т.е. $F_{N+1}(x) > F_N(x)$ для $1 \leq x \leq x_1^{(N+1)}$ и точка x_{N+1}^* , определяющая ноль функции $\varphi_{N+1}(x)$ на интервале $(1; x_1^{(N+1)})$, отсутствует, что противоречит рассмотренным случаям $N=3$ и $N=4$; 3) если $F_N(x_1^{(N+1)}) > 0$, то $\varphi_{N+1}(x_1^{(N+1)}) < 0$ и $\varphi_{N+1}(x)$ имеет единственный ноль в точке $x_{N+1}^* \in (1; x_1^{(N+1)})$. Таким образом, $\varphi_{N+1}(x^{(N+1)}) < 0$ и $1 < x_{N+1}^* < x_1^{(N+1)}$.

Представим $\varphi_{N+1}(x)$ в виде

$$\varphi_{N+1}(x) = \frac{1}{N(N-1)} x e^{-x} \Phi_{N+1}(x), \quad \Phi_{N+1}(x) = \sum_{k=2}^N \frac{k^{k+1}}{(k-2)!} x^{k-2} e^{-(k-2)x} \left(\frac{k+2}{k} - x \right).$$

Знак функции $\varphi_{N+1}(x)$ определяется знаком функции $\Phi_{N+1}(x)$, при этом точка нуля этих функций есть x_{N+1}^* . Точки $x_k = (k+2)/k$ образуют интервалы $(1 + 2/(k+1); 1 + 2/k)$, $k = \overline{2, N}$, в один из которых попадает точка $x_1^{(N+1)}$, в которой $\Phi_{N+1}(x_1^{(N+1)}) < 0$, при этом $x_1^{(N+1)} > x_{N+1}^*$. Точка $x_1^{(N+1)}$ разбивает слагаемые в $\Phi_{N+1}(x)$ на положительные и отрицательные. При изменении x от $x_1^{(N+1)}$ до 2 количество положительных слагаемых уменьшается (для $x \geq 2$ положительных слагаемых нет). Тогда $\Phi_{N+1}(x) < 0$ для $x > x_1^{(N+1)}$. Таким образом, функция $\varphi_{N+1}(x)$ не достигает нуля для $x > x_{N+1}^*$. Доказательство завершено.

Следствие 1. Функция $P_N(x)$ ($x > 0$), определённая формулой (7), имеет единственный максимум в точке x_N^0 , лежащей в границах (13).

Следствие 2. Плотность вероятностей $P_N(\hat{\lambda})$, определённая в (6), имеет единственный максимум в точке $\hat{\lambda}_0 = \lambda / x_N^0$.

Следствие 3. В силу неравенства (13) при $N \rightarrow \infty$ точка нуля функции (7) $x_N^0 \rightarrow 1$, т.е. $\hat{\lambda}_0 \rightarrow \lambda$.

3. Оценивание числа состояний асинхронного потока

Результатом процесса наблюдения за пуассоновским потоком является последовательность оценок (2), которая позволяет построить гистограмму оценок, являющуюся, в свою очередь, оценкой плотности вероятностей $P_N(\hat{\lambda})$. В силу этого корректно построенная огибающая гистограммы будет иметь единственный максимум. Если на временной оси в течение достаточно большого временного интервала функционирует пуассоновский поток интенсивности λ_1 , а затем на таком же достаточно большом интервале времени – пуассоновский поток интенсивности λ_2 ($\lambda_1 \gg \lambda_2$), то построенная по результатам наблюдений гистограмма оценок будет иметь два максимума. Корректно построенная огибающая первой гистограммы ($\lambda = \lambda_1$) и огибающая второй гистограммы ($\lambda = \lambda_2$) будут иметь общую точку пересечения, которая делит ось оценок на две части: в одной части будет иметь место гипотеза $\hat{\lambda} = \lambda_1$, в другой – гипотеза $\hat{\lambda} = \lambda_2$. Так как асинхронный поток имеет n состояний: $\lambda(t) = \lambda_i$, $i = \overline{1, n}$, то в процессе наблюдения за потоком при построении оценок (2) одна часть оценок будет относиться к оценкам, построенным на участках стационарности, другая часть – к оценкам, построенным на временных интервалах, одна часть которых относится к одному участку стационарности ($\lambda(t) = \lambda_i$, $i = \overline{1, n}$) процесса $\lambda(t)$, другая часть – к другому участку стационарности ($\lambda(t) = \lambda_j$, $j = \overline{1, n}$,

$i \neq j$) процесса $\lambda(t)$. Построенная при этом гистограмма оценок при достаточно больших длительностях участков стационарности будет иметь несколько максимумов. Число этих максимумов $\hat{n}$ и будет оценкой числа состояний n асинхронного потока событий.

Оценки $\hat{\lambda}_i, i = \overline{1, \hat{n}}$, определяются как точки максимума огибающей гистограммы оценок. Другой путь определения оценок $\hat{\lambda}_i, i = \overline{1, \hat{n}}$, заключается в построении $\hat{n}$ функций правдоподобия и решении $\hat{n}$ оптимизационных задач на максимизацию по переменной λ функций правдоподобия.

Заключение

Полученные результаты делают возможным в процессе наблюдения за потоком оценивание неизвестного числа состояний асинхронного МС-потока событий. Кроме того, предложенный эвристический подход позволяет осуществлять оценку $\hat{\lambda}_i, i = \overline{1, \hat{n}}$, величин λ_i , соответствующих тому или иному участку стационарности процесса $\lambda(t)$.

ЛИТЕРАТУРА

1. Cox D.R. The analysis of non-Markovian stochastic processes // Proceedings of Cambridge Philosophical Society. 1955. V. 51, No. 3. P. 433–441.
2. Kingman J.F.C. On doubly stochastic Poisson process // Proceedings of Cambridge Philosophical Society. 1964. V. 60, No. 4. P. 923–930.
3. Горцев А.М., Назаров А.А., Терпугов А.Ф. Управление и адаптация в системах массового обслуживания. Томск: Изд-во ТГУ, 1978. 208 с.
4. Башарин Г.П., Кокотушкин В.А., Наумов В.А. О методе эквивалентных замен расчета фрагментов сетей связи // Изв. АН СССР. Техн. кибернетика. 1979. № 6. С. 92–99.
5. Башарин Г.П., Кокотушкин В.А., Наумов В.А. О методе эквивалентных замен расчета фрагментов сетей связи // Изв. АН СССР. Техн. кибернетика. 1980. № 1. С. 55–61.
6. Neuts M.F. A versatile Markov point process // J. Appl. Probab. 1979. V. 16. P. 764–779.
7. 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.
8. Lucantoni D.M., Neuts M.F. Some steady-state distributions for the MAP/SM/1 queue // Communications in Statistics Stochastic Models. 1994. V. 10. P. 575–598.
9. Card H.C. Doubly stochastic Poisson processes in artificial neural learning // Neural Networks, IEEE Transaction. 1998. V. 9. Issue 1. P. 229–231.
10. Breuer L. An EM algorithm for batch Markovian arrival processes and its comparison to a simpler estimation procedure // Annals of Operations Research. 2002. V. 112. P. 123–138.
11. Telek M., Horvath G. A minimal representation of Markov arrival processes and a moments matching method // Performance Evaluation. 2007. V. 64. P. 1153–1168.
12. Okamura H., Dohi T., Trivedi K.S. Markovian arrival process parameter estimation with group data // IEEE/ACM Transactions on Networking. 2009. V. 17. P. 1326–1339.
13. Horvath A., Horvath G., Telek M. A joint moments based analysis of networks of MAP/MAP/1 queues // Performance Evaluation. 2010. V. 67. P. 759–788.
14. Нежелская Л.А. Нелинейная оптимальная фильтрация дважды стохастического потока с инициативными событиями // Тезисы докладов научно-технической конференции «Микросистема–91». Суздаль. М.: Всесоюз. общество информатики и вычислительной техники, 1991. С. 26–28.
15. Горцев А.М., Нежелская Л.А. Оценка параметров синхронного МС-потока событий // Сети связи и сети ЭВМ: тез. докл. Восьмой Белорус. зимней школы-семинара по теории массового обслуживания. Минск: Изд-во БГУ, 1992. С. 33.
16. Горцев А.М., Нежелская Л.А. Оптимизация параметров адаптера при наблюдении за МС-потоком // Стохастические и детерминированные модели сложных систем. Новосибирск: Изд-во ВЦ СО АН СССР, 1988. С. 20–32.
17. Горцев А.М., Нежелская Л.А. Оптимальная нелинейная фильтрация марковского потока событий с переключениями // Техника средств связи. Сер. Системы связи. 1989. Вып. 7. С. 46–54.
18. Нежелская Л.А. Алгоритм оценивания состояний полусинхронного потока событий с учётом мёртвого времени // Массовое обслуживание: потоки, системы, сети: материалы Четырнадцатой Белорус. зимней школы-семинара по теории массового обслуживания. Минск: Изд-во БГУ, 1998. С. 18–21.

19. Горцев А.М., Калягин А.А., Нежелская Л.А. Оптимальная оценка состояний обобщенного полусинхронного потока событий // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2010. № 2(11). С. 66–81.
20. Горцев А.М., Нежелская Л.А. О связи МС-потоков и МАР-потоков событий // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2011. № 1(14). С. 13–21.
21. Gortsev A.M., Nezhel'skaya L.A., Solov'ev A.A. Optimal State Estimation in MAP Event flows with Unextendable Dead Time // Automation and Remote Control. 2012. V. 73, No. 8. P. 1316–1326.
22. Bushlanov I.V., Gortsev A.M., Nezhel'skaya L.A. Estimating parameters of the synchronous towfold-stochastic flow of events // Automation and Remote Control. 2008. V. 69, No. 9. P. 1517–1533.
23. Bushlanov I.V., Gortsev A.M. Optimal estimation of the state of a synchronous double stochastic flow of events // Automation and Remote Control. 2004. V. 65, No. 9. P. 1389–1399.
24. Горцев А.М., Зуевич В.Л. Оптимальная оценка состояний асинхронного дважды стохастического потока событий с произвольным числом состояний // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2010. № 2(11). С. 44–65.
25. Горцев А.М., Зуевич В.Л. Оптимальная оценка состояний асинхронного потока событий с конечным числом состояний в условиях непродлевающегося мертвого времени // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2010. № 3(12). С. 41–53.
26. Горцев А.М., Зуевич В.Л. Оптимальная оценка параметров асинхронного дважды стохастического потока событий с произвольным числом состояний // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2011. № 4(17). С. 25–40.
27. Наумов В.А. Марковские модели потоков требований // Системы массового обслуживания и информатика. М. : Изд-во УДН, 1987. С. 67–72.
28. Феллер В. Введение в теорию вероятностей и её приложения. М. : Мир, 1967. Т. 2. 752 с.
29. Шуленин В.П. Математическая статистика. Томск : Изд-во НТЛ, 2012. Ч. 1. 540 с.

Беккерман Екатерина Николаевна. E-mail: bekkermankn@tspu.edu.ru

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

Томский государственный университет

Поступила в редакцию 12 апреля 2014 г.

Bekkerman Ekaterina A., Gortsev Alexander M. (Tomsk State University, Tomsk, Russian Federation).

Heuristic algorithm for estimating the number of states of asynchronous MC-flow events.

Keywords: non-synchronous Markov chain flow of events; flow state; estimation; stationary state interval; probability density; histogram.

The non-synchronous MC-flow of events, which is a mathematical model employed in Integrated Services Digital Networks (ISDNs), is considered.

The flow intensity is a piecewise constant stationary random process $\lambda(t)$ with n states $\lambda_1, \lambda_2, \dots, \lambda_n$ ($\lambda_1 > \lambda_2 > \dots > \lambda_n > 0$). Process $\lambda(t)$ at a moment t is in the state i if $\lambda(t) = \lambda_i$ ($i = \overline{1, n}$). During stationary state interval there is a Poisson flow of events with state rate λ_i . Duration of staying in the i -th state is an exponentially distributed random value with the distribution function

$F_i(\tau) = 1 - e^{-\alpha_{ii}\tau}$, $\alpha_{ii} = - \sum_{j=1, j \neq i}^n \alpha_{ij}$ ($i = \overline{1, n}$), where $\alpha_{ij} > 0$ ($i, j = \overline{1, n}$, $i \neq j$) is the transition rate from state i to state j .

A flow occurs under conditions of uncertainty when information about the parameters and the number of states is not available. The problem is to estimate the values of unknown parameters λ_i , α_{ij} ($i, j = \overline{1, n}$, $i \neq j$), and n , using information from the moments $t_1, \dots, t_k$.

According to the definition of a flow, trace includes time intervals with $\lambda(t) = \lambda_i$ (the stationary state intervals), where the flow behaves like a Poisson flow. We consider the histogram of all possible rate estimates of some stationary state interval, on which $N+1$ events of a Poisson flow with rate λ were realized: $t_1, t_2, \dots, t_{N+1}$ ($\lambda \in \lambda_1, \lambda_2, \dots, \lambda_n$). This histogram is an estimate of the mixture of distribution densities:

$$P_N(\hat{\lambda}) = \sum_{k=2}^N q_k^{(N)} P_k(\hat{\lambda}), \quad P_k(\hat{\lambda}) = \lambda^k \frac{(k-1)^{k-1}}{(k-2)!} \left(\frac{1}{\hat{\lambda}}\right)^{k+1} e^{-\lambda \frac{k-1}{\hat{\lambda}}},$$

$$q_k^{(N)} = \frac{2(N-k+1)}{N(N-1)}, \quad k = \overline{2, N}; \quad \sum_{k=2}^N q_k^{(N)} = 1.$$

It is proved that $P_N(\hat{\lambda})$ is an unimodal function, and $\arg \max \left(P_N(\hat{\lambda}) \right) = \hat{\lambda}_0 \xrightarrow{N \rightarrow \infty} \lambda$.

So, the histogram of all possible rate estimates of a Poisson process has the only maximum. The maximum value indicates the flow rate. Each stationary state interval in the trace of MC-flow is a Poisson flow segment of constant intensity. The histogram, based on this trace, has several maxima in the condition of large duration of stationary state intervals. The number of these maxima is the estimate of the number of states n .

The estimates $\hat{\lambda}_i, i = \overline{1, \hat{n}}$ are defined as the maximum points of histogram envelope curve of all possible rate estimates. Another way to determine the estimates $\hat{\lambda}_i, i = \overline{1, \hat{n}}$ is to build $\hat{n}$ likelihood functions and to solve $\hat{n}$ optimization problems on maximization of the likelihood functions by the variable λ .

REFERENCES

1. Cox D.R. The analysis of non-Markovian stochastic processes. *Proceedings of Cambridge Philosophical Society*, 1955, vol. 51, no. 3, pp. 433-441.
2. Kingman J.F.C. On doubly stochastic Poisson process. *Proceedings of Cambridge Philosophical Society*, 1964, vol. 60, no. 4, pp. 923-930. (1964).
3. Gortsev A.M., Nazarov A.A., Terpugov A.F. *Upravlenie i adaptatsiya v sistemakh massovogo obsluzhivaniya* [Control and adaptation in queueing systems]. Tomsk: Tomsk State University Publ., 1978. 208 p.
4. Basharin G.P., Kokotushkin V.A., Naumov V.A. O metode ekvivalentnykh zamen rascheta fragmentov setey svyazi [On a method of equivalent substitutions for communications network fragments calculation]. *Izv. AN SSSR. Tekhn. Kibernetika*, 1979, no. 6, pp. 92-99.
5. Basharin G.P., Kokotushkin V.A., Naumov V.A. O metode ekvivalentnykh zamen rascheta fragmentov setey svyazi [On a method of equivalent substitutions for communications network fragments calculation]. *Izv. AN SSSR. Tekhn. kibernetika*. 1980, no. 1, pp. 55-61.
6. Neuts M.F. A versatile Markov point process. *Journal of Applied Probability*, 1979, vol. 16, pp. 764-779.
7. Lucantoni D.M. New results on the single server queue with a batch Markovian arrival process. *Communications in Statistics Stochastic Models*, 1991, vol. 7, pp. 1-46. DOI: 10.1080/15326349108807174.
8. Lucantoni D.M., Neuts M.F. Some steady-state distributions for the MAP/SM/1 queue. *Communications in Statistics Stochastic Models*. 1994, vol. 10, pp. 575-598. DOI: 10.1080/15326349408807311.
9. Card H.C. Doubly stochastic Poisson processes in artificial neural learning. *Neural Networks, IEEE Transaction*. 1998, vol. 9, is. 1, pp. 229-231. DOI: 10.1109/72.655046.
10. Breuer L. An EM algorithm for batch Markovian arrival processes and its comparison to a simpler estimation procedure. *Annals of Operations Research*. 2002, vol. 112, pp. 123-138. DOI: 10.1023/A:1020981005544.
11. Telek M., Horvath G. A minimal representation of Markov arrival processes and a moments matching method. *Performance Evaluation*. 2007, vol. 64, pp. 1153-1168. DOI: 10.1016/j.peva.2007.06.001.
12. Okamura H., Dohi T., Trivedi K.S. Markovian arrival process parameter estimation with group data. *IEEE/ACM Transactions on Networking*. 2009, vol. 17, pp. 1326-1339. DOI: 10.1109/TNET.2008.2008750.
13. Horvath A., Horvath G., Telek M. A joint moments based analysis of networks of MAP/MAP/1 queues. *Performance Evaluation*. 2010, vol. 67, pp. 759-788. DOI: 10.1109/QEST.2008.26.
14. Nezhe'skaya L.A. Nonlinear optimal filtration of double stochastic flow with enterprising events. *Tezisy dokladov nauchno-tekhnicheskoy konferentsii "Mikrosistema-91"* [Theses of scientific and technical conference "Microsystem-91"]. Suzdal', Moscow: Vsesoyuz. obshchestvo informatiki i vychislitel'noy tekhniki Publ., 1991, pp. 26-28. (In Russian).
15. Gortsev A.M., Nezhe'skaya L.A. [Parameters estimation of synchronous Markov chain (MC) flow of events]. *Seti svyazi i seti EVM : tez. dokl. Vos'moy Belorusskoy zimney shkoly-seminara po teorii massovogo obsluzhivaniya* [Telecommunications and computer networks: theses of eighth Belarusian winter workshop on queueing theory]. Minsk: Belarusian State University Publ., 1992, p. 33. (In Russian).
16. Gortsev A.M., Nezhe'skaya L.A. *Optimizatsiya parametrov adaptera pri nablyudenii za MC-potokom* [Parameters optimization in monitoring the Markov chain (MC) flow]. In: Topchiy V.A. (ed.). *Stokhasticheskie i determinirovannye modeli slozhnykh sistem* [Stochastic and deterministic models of complex systems]. Novosibirsk: VTs SO AN SSSR Publ., 1988, pp. 20-32.
17. Gortsev A.M., Nezhe'skaya L.A. Optimal'naya nelineynaya fil'tratsiya markovskogo potoka sobytii s pereklyucheniymi [The optimal nonlinear filtration of Markovian flow of events with commutation]. *Tekhnika sredstv svyazi. Ser. Sistemy svyazi*, 1989. Vyp. 7, pp. 46-54.
18. Nezhe'skaya L.A. [Algorithm of states evaluation of semisynchronous flow of events in consideration of dead, time]. *Massovoe obsluzhivanie: potoki, sistemy, seti : materialy Chetyrnadtsatoy Belorusskoy zimney shkoly-seminara po teorii massovogo obsluzhivaniya* [Queueing theory: flows, systems and networks: materials of 14th Belarusian winter workshop on queueing theory]. Minsk: Belarusian State University Publ., 1998, pp. 18-21. (In Russian).
19. Gortsev A.M., Kalyagin A.A., Nezhe'skaya L.A. Optimal states estimation of integrated semisynchronous flow of events. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika - Tomsk State University Journal of Control and Computer Science*, 2010, no. 2(11), pp. 66-81. (In Russian).

20. Gortsev A.M., Nezhel'skaya L.A. On connection of MC flows and MAP flows of events. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika - Tomsk State University Journal of Control and Computer Science*, 2011, no. 1(14), pp. 13-21. (In Russian).
21. Gortsev A.M., Nezhel'skaya L.A., Solov'ev A.A. Optimal state estimation in MAP event flows with unextendable dead time. *Automation and Remote Control*, 2012, vol. 73, no. 8, pp. 1316-1326. DOI: 10.1134/S000511791208005X.
22. Bushlanov I.V., Gortsev A.M., Nezhelskaya L.A. Estimating parameters of the synchronous twofold-stochastic flow of events. *Automation and Remote Control*, 2008, no. 9, pp. 1517-1533. DOI: 10.1134/S0005117908090075.
23. Bushlanov I.V., Gortsev A.M. Optimal estimation of the state of a synchronous double stochastic flow of events. *Automation and Remote Control*, 2004, vol. 65, no. 9, pp. 1389-1399. DOI: 10.1023/B:AURC.0000041418.09187.63.
24. Gortsev A.M., Zuevich V.L. Optimal state estimation of nonsynchronous doubly stochastic finite state flow of events. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika - Tomsk State University Journal of Control and Computer Science*, 2010, no. 2(11), pp. 44-65. (In Russian).
25. Gortsev A.M., Zuevich V.L. Optimal state estimation of nonsynchronous finite state flow of events in conditions of constant dead time. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika - Tomsk State University Journal of Control and Computer Science*, 2010, no. 3(12), pp. 41-53. (In Russian).
26. Gortsev A.M., Zuevich V.L. Optimal parameter estimation of nonsynchronous doubly stochastic finite state flow of events. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika - Tomsk State University Journal of Control and Computer Science*, 2011, no. 4(17), pp. 25-40. (In Russian).
27. Naumov V.A. *Markovskiye modeli potokov trebovaniy. Sistemy massovogo obsluzhivaniya I informatika* [Markovian models of arrival processes. Queuing systems and computer science]. Moscow: Peoples' Friendship University (PFU) Publ., 1987, pp. 67-72.
28. Feller W. *Vvedeniye v teoriyu veroyatnostey i eye prilozheniya* T.2 [An Introduction to Probability Theory and its applications. Vol. II]. Translated from English by Yu.A. Prokhorov. Moscow: Mir Publ., 1967. 752 p.
29. Shulenin V.P. *Matematicheskaya statistika. Chast' 1* [Mathematical statistics. Vol. 1]. Tomsk: NTL Publ., 2012. 540 p.

УДК 004.8, 004.942, 622.279

Ю.А. Келлер

ПРИМЕНЕНИЕ КЛАСТЕРИЗАЦИИ ДАННЫХ НА ОСНОВЕ САМООРГАНИЗУЮЩИХСЯ КАРТ КОХОНЕНА ПРИ ПОДБОРЕ СКВАЖИН-КАНДИДАТОВ ДЛЯ МЕТОДОВ УВЕЛИЧЕНИЯ НЕФТЕОТДАЧИ

Предложена методология предварительной кластеризации промысловой информации на основе самоорганизующихся карт Кохонена в рамках работ по подбору скважин-кандидатов для выполнения методов увеличения нефтеотдачи. Приведены результаты численных расчётов, характеризующие эффективность разработанной методики.

Ключевые слова: методы увеличения нефтеотдачи; нагнетательная скважина; выравнивание профиля приёмистости; самоорганизующиеся карты Кохонена; нейронные сети.

При разработке нефтяных месторождений одной из главных проблем на данный момент является преждевременное обводнение добываемой продукции. Уже сейчас это приводит к стабилизации темпов роста извлечения нефти, а в ближайшие 10–20 лет при сохранении выбранной стратегии разработки может привести к его снижению [1]. Поэтому с целью увеличения рентабельного срока эксплуатации нефтяных месторождений для снижения текущей обводненности продукции применяется выравнивание профиля приёмистости (ВПП) на нагнетательных скважинах. Для этого в скважину закачивается специальный реагент для уменьшения проницаемости промытых прослоёв пласта, что, в свою очередь, приводит к более равномерному фронту вытеснения нефти водой, вовлечения в разработку непромытых зон пласта, уменьшая тем самым обводненность продукции добывающих скважин.

На данный момент критерии выбора скважин-кандидатов под методы увеличения нефтеотдачи (МУН) проработаны в полной мере, поэтому на первый план выходит корректная оценка ожидаемой дополнительной добычи нефти за счет снижения обводненности продукции в результате проведения МУН. Известны методики по оцениванию технологической эффективности ВПП на основе теории нейронных сетей [2–4].

В данной работе предлагается осуществлять предварительную кластеризацию исходной промысловой информации с помощью самоорганизующихся карт (СОК) Кохонена с последующим построением нейронных сетей по методологии [4] для оценки технологической эффективности ВПП.

1. Алгоритм кластеризации

Особенностью моделирования задач оценки технологической эффективности ВПП является большая степень дифференциации исходной информации, обусловленная расчлененностью характеристик пласта (пористость, проницаемость, выдержанность и др.), различными промысловыми показателями скважин (дебиты, величины забойного давления), степенью эффективности поддержки пластового давления в зоне отборов и др. Таким образом, при попытке создания модели искусственной нейронной сети (ИНС), основанной на несгруппированной исходной информации [Там же], средняя ошибка предсказания дебита нефти добывающей скважины после ВПП составила более 50%. Поэтому применение данной модели на практике вносило существенную долю неопределенности в прогнозируемые характеристики. На данный момент известны успешные опыты внедрения кластеризации

информации при решении нефтегазопромысловых задач в области МУН [5, 6]. С целью улучшения предсказательной способности модели ИНС [4] было принято решение использовать СОК Кохонена для предварительной кластеризации исходной информации.

Задача кластеризации сводится к разбиению множества объектов, представленных векторами $x^i \in R^n$, $i = 1, 2, \dots, N$, на группы, называемые кластерами. Объекты формируются в кластеры таким образом, чтобы те из них, что попали в один и тот же кластер, были максимально похожи друг на друга и одновременно максимально отличными от объектов из других кластеров [7]. До внедрения систем искусственного интеллекта предпочтение отдавалось классическим методам кластеризации (древовидная кластеризация, метод k -средних и т.д.). В данной группе методов предполагается, что наиболее типичный представитель кластера (ядро кластера) известен, и задача кластеризации решается путём «стягивания» к ядрам кластера других векторов [8. С. 241]. Однако в последние годы бурно развивающиеся альтернативные классическим методам кластеризации стали СОК Кохонена. Они применяются при моделировании, прогнозировании, поиске закономерностей в больших массивах данных, выявлении кластеров [9, 10]. Важное преимущество новой методологии кластеризации заключается в особенности сети Кохонена от других ИНС: данный тип нейронных сетей способен к неконтролируемому обучению, т.е. обучающая выборка состоит лишь из значений входных переменных, в процессе обучения нет сопоставления выходов нейронов с эталонными значениями. Схематично структура сети Кохонена представлена на рис. 1.

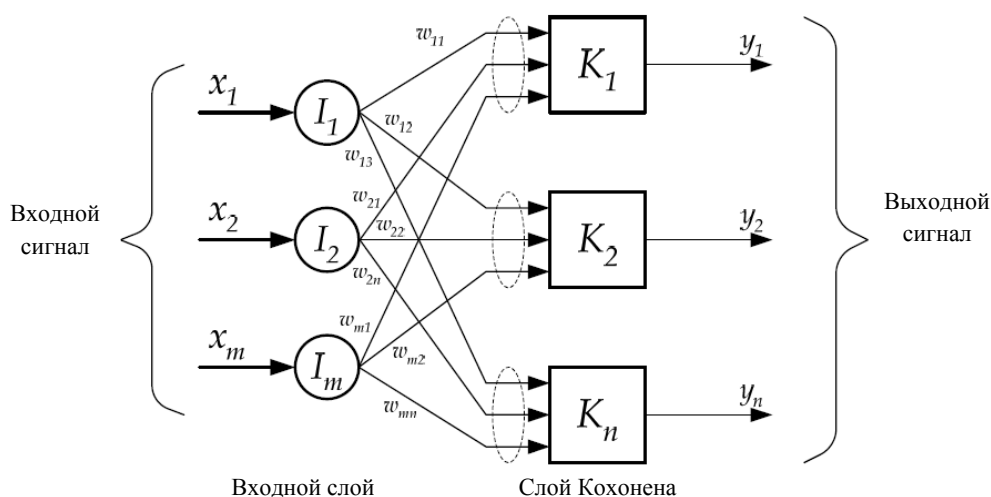


Рис. 1. Структура самоорганизующейся сети Кохонена:

$x_1, x_2, \dots, x_m$ – множество входных характеристик; ω_{ij} – весовые коэффициенты сети; $y_1, y_2, \dots, y_n$ – множество выходных характеристик, соответствующих n кластерам

Сам алгоритм обучения получил название «победитель забирает всё», который представляет последовательность следующих шагов [11]:

1. Выбирается шаг обучения $\eta > 0$ (обычно от 0,1 до 0,7).
2. Происходит инициализация весов сети случайными числами.
3. Осуществляется настройка весов. Для каждого вектора производится:
 - расчёт выходов нейронов слоя Кохонена. Выбирается нейрон-победитель r с наибольшим значением на выходе;
 - корректировка весов победившего нейрона:

$$\omega_r = \omega_r + \eta(x - \omega_r);$$

- нормировка вектора весов нейрона-победителя:

$$\omega_r = \frac{\omega_r}{\|\omega_r\|}.$$

Пункты 1–3 представляют собой серию последовательных приближений. Во время обучения сети на вход подаются исходные характеристики и сеть настраивается под определенные закономерности, обнаруженные в выборке. На начальном этапе процедура обучения начинается со случайно выбранного выходного расположения центров. Далее на вход сети подаются обучающие примеры из выборки и находится наиболее похожий нейрон, для которого скалярное произведение весов ω_{ij} и обучающего примера минимально. Система назначает этот нейрон победителем, и он объявляется центральным для настройки весов всех близкорасположенных нейронов. В отличие от других нейронных сетей, суть обучения сводится к настройке весов таким образом, чтобы достичь наилучшего совпадения с поданными на вход примерами. После того как на вход сети подано достаточное количество обучающих примеров, сеть имеет возможность выделить группы и система весов генерируется так, чтобы близкие с точки зрения топологии узлы были чувствительны к идентичным входным сигналам. В итоге центр кластера фиксируется так, чтобы наилучшим образом группировать примеры, для которых такой нейрон выигрышный. Поэтому завершением обучения СОК Кохонена будет задание меры соседства нейронов, окружающих победивший нейрон. СОК формирует такую топологическую карту, где идентичные примеры представляют кластеры, близко расположенные друг к другу. В итоге СОК Кохонена разбивает входные примеры на кластеры, схожие по совокупности своих характеристик, и позволяет визуальнo представить их в плоскости нейронов. Отсюда следует еще одно важное свойство сетей Кохонена – они позволяют трансформировать N -мерное пространство входных признаков в двумерное. Поэтому задача инженерного анализа может быть сведена к простому визуальнoму наблюдению кластеров на топологической карте с выдачей соответствующих рекомендаций.

При этом СОК Кохонена не лишены недостатка классических методов кластеризации: перед проведением обучения сети необходимо выбрать число кластеров, на которое будет разбиваться исходное множество.

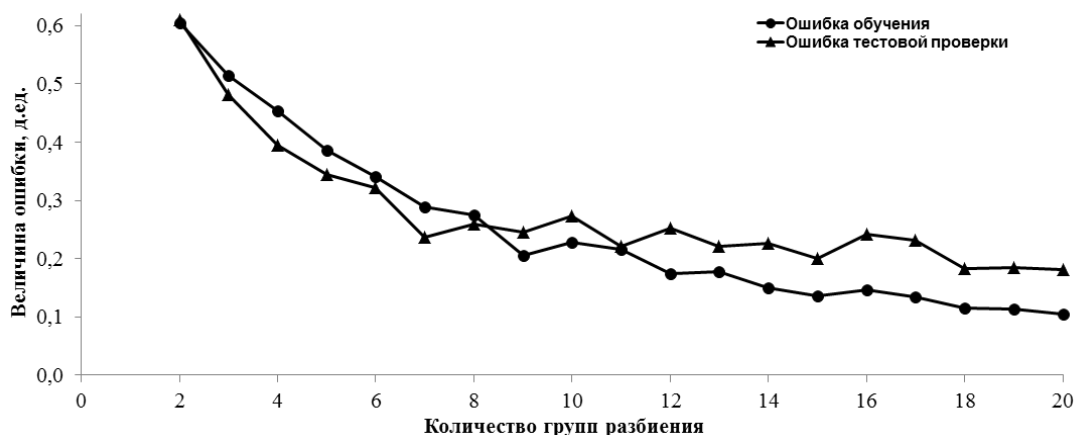


Рис. 2. Зависимость величины ошибки обучения от числа кластеров

Исходя из собственного опыта проведения работ по подбору физико-химическим МУН на нефтяных месторождениях РФ, а также из опыта работы других авторов [12], был пересмотрен набор исходных параметров, предложенных в работе [4]. Исключены из модели ИНС параметры: источник обводнения скважины, вязкость нефти, толщина обрабатываемого слоя коллектора; добавлены в модель ИНС: потенциал динамического уровня и средняя компенсация на участке. Основной целью при пересмотре входного набора параметров было избавиться от тех, что несут в себе долю неопределенности либо определение которых с необходимой степенью достоверности невозможно. Построение сети Кохонена осуществлялось в программном обеспечении Statistica. В качестве входной информации для обучения СОК использовался опыт проведения ВПП на одном из нефтяных месторождений Поволжского региона в период с 2007 по 2011 г. В обучающую выборку попали промысловые пока-

затели по более чем 100 скважинам. 90% исходной информации использовалось для обучения сети Кохонена, 10% – для контрольной проверки работоспособности созданной модели. Выбор оптимального числа кластеров для разбиения осуществлялся следующим образом: поочередно увеличивая число кластеров, на которые будет разбиваться исходная выборка, оценивались величина ошибки обучения модели и величина ошибки кластеризации для тестовой выборки. На оси абсцисс отмечалось число кластеров, на оси ординат – величина ошибки, соответствующая данному числу кластеров. Результаты представлены на рис. 2.

Из рис. 2 видно, что при достижении определенного числа кластеров (количества групп разбиения) величина ошибки значительно не уменьшается и дальнейшее увеличение числа кластеров приводит лишь к увеличению времени обучения без существенного улучшения точности обучения. Поэтому было принято решение выделить четыре группы (кластера) для кластеризации исходной информации.

2. Результаты моделирования

Моделирование осуществлялось следующим образом: рассматривались два варианта проведения моделирования для предсказания эффективности проведения ВПП. В первом варианте предварительная кластеризация исходной информации не осуществлялась и ИНС создавалась на всем объеме исходной информации. Во втором варианте на первом этапе вся исходная информация была разделена на четыре кластера и в рамках каждого кластера создавалась своя ИНС (более подробно процедура создания ИНС представлена в [4]). В качестве выходного (предсказываемого) параметра модели ИНС рассматривалась обводненность добывающей скважины после ВПП. Выбор данной характеристики был обусловлен тем, что успешность проведения ВПП на месторождении характеризуется снижением обводненности продукции добывающих скважин (как по отдельным скважинам, так и в целом по участку). Под обводненностью продукции понимается отношение добываемой воды в объеме добываемого флюида. Например, обводненность в 90% на добывающей скважине означает, что из 1000 м³ добываемой жидкости в сутки только 100 м³ составляет нефть, а остальные 900 м³ – вода. Поэтому суммарное снижение обводненности продукции после ВПП позволяет получить дополнительную добычу нефти.

Далее для каждого из двух вариантов оценивалось среднее арифметическое расхождения предсказываемой и фактической обводненности после ВПП. Кросс-плот распределения расчетных и фактических обводненностей представлен на рис. 3, 4. Рис. 3 соответствует первому варианту моделирования (без предварительной кластеризации информации), рис. 4 – второму варианту моделирования (с предварительной кластеризацией информации).

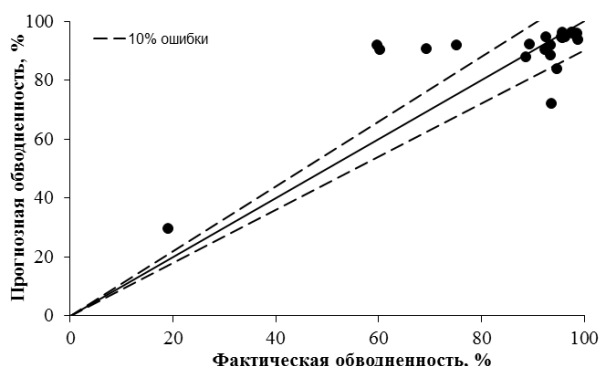


Рис. 3. Сравнение фактических и расчетных значений обводненности после ВПП без предварительной кластеризации информации

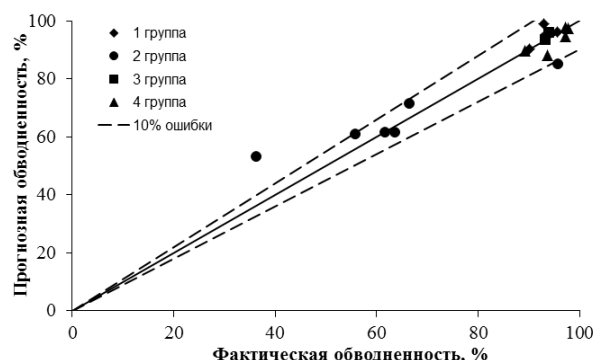


Рис. 4. Сравнение фактических и расчетных значений обводненности после ВПП с предварительной кластеризацией информации

Средняя ошибка предсказания обводненности после ВПП без предварительной кластеризации составила 10,6%, а с предварительной кластеризацией – 6,5%, что почти в два раза меньше. Данный результат моделирования подтверждает, что при попытке использования всего объема выборки для обучения ИНС результат предсказания может быть неудовлетворительным в связи с большой дифференциацией исходной информации. Группировка такой информации по кластерам, которые группируются по степени схожести характеристик, помогает строить более точные с точки зрения предсказания модели ИНС. Важно отметить, что и в первом и во втором случае имеются отдельные скважины, ошибка предсказания обводненности для которых значительно превышает 10%-ный порог. Это объясняется тем, что на данной скважине после ВПП было произведено другое геолого-технологическое мероприятие (ГТМ), спровоцировавшее резкий рост обводненности продукции. Поскольку в обучающей выборке такие прецеденты отсутствовали, то построенная модель ИНС для данной скважины не смогла адекватно предсказать ожидаемую обводненность после ВПП. В дальнейшем планируется доработка методики [4] с включением в исходный набор данных информации о проводимых на участке ГТМ.

Заключение

Решена задача предварительной кластеризации данных с помощью СОК Кохонена при оценке эффективности проведения ВПП на нефтяном месторождении с помощью математического моделирования. Показано, что применение предварительной кластеризации позволило почти вдвое снизить погрешность предсказания обводненности на добывающих скважинах одного из нефтяных месторождений Поволжского региона.

ЛИТЕРАТУРА

1. *Нефтегазодобывающая и нефтеперерабатывающая промышленность: Тенденции и прогнозы.* URL: <http://vid1.rjan.ru/ig/ratings/oil9.pdf> (дата обращения: 28.05.2014).
2. *Herbas J., Moreno R., Marin T.A., Romero M.F., Coombe D.* Reservoir simulation of non selective of a polymer gel treatment to improve water injection profiles and sweep efficiency in the Lagomar field Western Venezuela (SPE-92025-MS) // SPE international petroleum conference. Puebla, 2004. 10 p.
3. *Ferreira I., Gammiero A., Llamedo M.* Design of a neural network model for predicting well performance after water shutoff treatments using polymer gels (SPE 153908) // SPE Latin American and Caribbean petroleum conference. Mexico, 2012. 9 p.
4. *Келлер Ю.А., Усков А.А.* Применение теории регрессионного анализа и нейронных сетей для оценки эффективности применения потокоотклоняющих технологий на группе месторождений ОАО «Самаранефтегаз» // Инженерная практика. 2013. № 10. С. 56, 57.
5. *Иванов Е.Н., Кононов Ю.М.* Выбор методов увеличения нефтеотдачи на основе аналитической оценки геолого-физической информации // Известия Томского политехнического университета. 2012. Т. 321, № 1. С. 149–154.
6. *Применение метода главных компонент при выборе новых эффективных технологий увеличения нефтеотдачи для условий месторождений ЗАО «Лукойл-Пермь» / И.И. Абызбаев и др.* // Геология, геофизика и разработка нефтяных месторождений. 2000. № 11. С. 18–21.
7. *Сараев П.В.* Нейросетевые методы искусственного интеллекта : учеб. пособие. Липецк : ЛГТУ, 2007. 64 с.
8. *Халафян А.А.* STATISTICA 6. Статистический анализ данных. М. : Бином, 2007. 508 с.
9. *Kohonen T.* Self-Organizing Maps (Third Extended Edition). N.Y., 2001. 501 p.
10. *Манжула В.Г., Федяшов Д.С.* Нейронные сети Кохонена и нечеткие нейронные сети в интеллектуальном анализе данных // Фундаментальные исследования. 2011. № 4. С. 108–114.
11. *Сеньковская И.С., Сараев П.В.* Автоматическая кластеризация в анализе данных на основе самоорганизующихся карт Кохонена // Вестник МГТУ им. Г.И. Носова. 2001. № 2. С. 78, 79.
12. *Жуков Р.Ю., Петраков А.М., Серов А.В.* Применение физико-химических технологий воздействия на завершающей стадии разработки месторождений // Нефтяное хозяйство. 2011. № 1. С. 51–53.

Келлер Юрий Александрович, канд. физ.-мат. наук. E-mail: kua1102@rambler.ru
Томский государственный университет

Поступила в редакцию 27 мая 2014 г.

Keller Yuri A. (Tomsk State University, Tomsk, Russian Federation).

The application of data clustering on basis of the Kohonen self-organizing maps by selecting candidate wells for enhanced oil recovery methods.

Keywords: enhanced oil recovery methods; injector well; improving water injection profile; self-organizing maps; artificial neural networks.

A methodology of preliminary clustering of field information is suggested on the basis of Kohonen self-organizing maps within the process of selecting candidate wells for enhanced oil recovery methods. The peculiarity of modeling problems of estimation of the technological efficiency of improving water injection profile is the high degree of differentiation of initial information, which is caused by the compartmentalization of formation characteristics (porosity, permeability, lateral continuity etc.), the different well characteristics (rates, bottom hole pressure values), the degree of efficiency of formation pressure maintenance in the production zone etc. Therefore, by creating an artificial neural network model based on ungrouped initial information, the average error of oil-production well rate prediction after improving water injection profile was over 50 per cent. To improve the prediction ability of a neural network, the initial data for training the network have been revised and preliminary grouping initial information by means of the Kohonen self-organizing maps has been done. It has been shown that preliminary grouping improved the prediction efficiency of a neural network nearly twice. The average error in the well water-cut prediction is 6.5 per cent. The suggested method is characterized by the far lesser work-time in comparison with the hydrodynamic modeling.

REFERENCES

1. *Neftegazodobyvayushchaya i neftepererabatyvayushchaya promyshlennost': Tendentsii i prognozy* [Oil and gas extraction and refining industry: Trends and Forecasts]. Available at: <http://vid1.ran.ru/ig/ratings/oil9.pdf>. (Accessed: 28th May 2014).
2. Herbas J., Moreno R., Marin T.A., Romero M.F., Coombe D. Reservoir simulation of non selective of a polymer gel treatment to improve water injection profiles and sweep efficiency in the Lagomar field Western Venezuela (SPE-92025-MS). *Proc. SPE international petroleum conference*. Puebla, 2004. 10 p.
3. Ferreira I., Gammiero A., Llamado M. Design of a neural network model for predicting well performance after water shutoff treatments using polymer gels (SPE 153908). *Proc. SPE Latin American and Caribbean petroleum conference*. Mexico, 2012. 9 p.
4. Keller Yu.A., Uskov A.A. Primenenie teorii regressionnogo analiza i neyronnykh setey dlya otsenki effektivnosti primeneniya potokootklonyayushchikh tekhnologiy na gruppe mestorozhdeniy OAO "Samaraneftegaz" [The application of the theory of regression analysis and neural networks to assess the effectiveness of fluid flow diverting methods on a group of fields "Samaraneftegaz"]. *Inzhenernaya praktika - Oilfield Engineering*, 2013, no. 10, pp. 56-57.
5. Ivanov E.N., Kononov Yu.M. The choice of enhanced oil recovery methods on the basis of analytical assessment of geological and field information. *Izvestiya Tomskogo politekhnicheskogo universiteta – Bulletin of the Tomsk Polytechnic University*, 2012, vol. 321, no. 1, pp. 149-154. (In Russian).
6. Katoshin A.F., Yakimenko G.F., Zotikov V.I., Abyzbaev I.I., Gafurov O.G. Primenenie metoda glavnykh komponent pri vybore novykh effektivnykh tekhnologiy uvelicheniya nefteotdachi dlya usloviy mestorozhdeniy ZAO "Lukoil-Perm" [The application of principal component analysis in the selection of new efficient technologies for increasing oil field conditions JSC "Lukoil-Perm"]. *Geologiya, geofizika i razrabotka neftyanykh mestorozhdeniy*, 2000, no. 11, pp. 18-21.
7. Saraev P.V. *Neyrosetevye metody iskusstvennogo intellekta* [Neural network artificial intelligence techniques]. Lipetsk: LTSU Publ., 2007. 64 p.
8. Khalafyan A.A. *STATISTICA 6. Statisticheskii analiz dannykh* [STATISTICA 6. Statistical analysis]. Moscow: Binom Publ., 2007. 508 p.
9. Kohonen T. *Self-organizing maps*. New York: Springer-Verlag, 2001. 501 p.
10. Manzhula V.G., Fedyashov D.S. Kohonen neural networks and fuzzy neural networks in data mining. *Fundamental'nye issledovaniya – Fundamental research*, 2011, no. 4, pp. 108-114. (In Russian).
11. Senkovskaya I.S., Saraev P.V. Avtomaticheskaya klasterizatsiya v analize dannykh na osnove samoorganizuyushchikhsya kart Kokhonena [Automatic clustering in data analysis based on Kohonen self-organizing maps]. *Vestnik MGTU im. G.I. Nosova*, 2001, no. 2, pp. 78-79.
12. Zhukov R.Yu., Petrakov A.M., Serov A.V. The Application of EOR-methods on the final stages of field development. *Neftyanoe khozyaistvo - Oil Industry*, 2011, no. 1, pp. 51-53. (In Russian).

О.М. Китурко, М.А. Маталыцкий

АСИМПТОТИЧЕСКИЙ АНАЛИЗ ДОХОДОВ В ЗАМКНУТОЙ НМ-СЕТИ С ПЕРЕМЕННЫМ ЧИСЛОМ ПРИОРИТЕТНЫХ И БЕСПРИОРИТЕТНЫХ ЗАЯВОК

Проведено исследование замкнутой экспоненциальной сети массового обслуживания с доходами, переменным числом приоритетных и беспriorитетных заявок и зависимыми от времени параметрами. Целью работы является нахождение среднего дохода каждой системы сети. Проведен асимптотический анализ в случае большого числа заявок в сети. Выведены дифференциальные уравнения для ожидаемых доходов каждой системы.

Ключевые слова: НМ-сеть; ожидаемый доход; метод диффузионной аппроксимации; приоритетные заявки.

Рассмотрим замкнутую сеть, в которой циркулирует определенное число заявок первого и второго типов, причем заявки не могут менять свой тип. Однотипные заявки, стоящие в очереди некоторой системы массового обслуживания (СМО), выбираются на обслуживание в произвольном порядке, например FIFO. Заявки первого типа имеют абсолютный приоритет по отношению к заявкам второго типа. В данном случае это будет означать выполнение двух условий: а) если в момент освобождения линии некоторой СМО после обслуживания заявки в ее очереди имеются приоритетные заявки, то любая из них занимает освободившуюся линию; б) если в систему обслуживания, все линии которой заняты обслуживанием, но не только приоритетных заявок, поступает приоритетная заявка, то она вытесняет неприоритетную заявку с одной из линий и начинает обслуживаться этой линией; вытесненная заявка становится в очередь рассматриваемой СМО. Когда вытесненная заявка поступает на обслуживание повторно, она дообслуживается в течение оставшегося времени обслуживания. Поскольку время обслуживания имеет показательное распределение, то можно считать, что вытесненная заявка будет обслуживаться заново, т.е. имеем так называемое неидентичное обслуживание.

Рассмотрим замкнутую экспоненциальную сеть массового обслуживания (МО) с приоритетными заявками, состоящую из $n+1$ систем МО (СМО) $S_0, S_1, \dots, S_n$, где S_0 – внешняя среда, под состоянием которой будем понимать вектор $k(t) = (k, t) = (k_{11}(t), k_{12}(t); k_{21}(t), k_{22}(t); \dots; k_{n1}(t), k_{n2}(t))$. Пусть $K_1(t)$ и $K_2(t)$ – общее число соответственно приоритетных и беспriorитетных заявок, обслуживаемых в сети; $K_1(t) + K_2(t) = K(t)$ – общее число обслуживаемых заявок в момент времени t . При этом $K_s(t)$ является кусочно-постоянными функциями времени с q интервалами постоянства:

$$K_s(t) = \begin{cases} K_{s1}, & t \in [T_0, T_1), \\ K_{s2}, & t \in [T_1, T_2), \\ \dots \\ K_{sq}, & t \in [T_{q-1}, T], \end{cases}$$

где K_{sl} – число заявок типа s в сети на l -м интервале времени $[T_{l-1}, T_l)$, $l = \overline{1, q}$, $K_l = K_{l1} + K_{l2}$. Например, в логистических транспортных системах (ЛТС) в связи с достаточно высокой стоимостью транспортных средств и относительно долгим периодом их эксплуатации число таких средств меняется через значительные промежутки времени, т.е. интервалы постоянства достаточно велики (на практике обычно 1–2 года). Число линий обслуживания в СМО $m_i(t)$, $i = \overline{1, n}$, $m_0(t) = K_1(t) + K_2(t)$, вероятности переходов заявок между ними $p_{ij}(t)$, $i, j = \overline{0, n}$, зависят от времени. Дисциплины об-

служивания заявок обоих типов в каждой СМО – FIFO. Обозначим через $\mu_{js}(t)$ интенсивности обслуживания заявок типа s в j -й СМО в момент времени t , $j = \overline{0, n}$, $s = 1, 2$. Введем также следующие обозначения:

$$\begin{aligned} \varepsilon_{j1}(k_{j1}(t), m_j(t)) &= \min\{k_{j1}(t), m_j(t)\}, \quad j = \overline{0, n}, \\ \varepsilon_{j2}(k_{j1}(t), k_{j2}(t), m_j(t)) &= \begin{cases} k_{j2}(t), & k_{j1}(t) + k_{j2}(t) < m_j(t), \\ m_j(t) - k_{j1}(t), & k_{j1}(t) < m_j(t), \quad k_{j1}(t) + k_{j2}(t) \geq m_j(t), \\ 0, & k_{j1}(t) \geq m_j(t). \end{cases} \quad j = \overline{0, n}, \end{aligned} \quad (1)$$

1. Вывод уравнения для плотности распределения ожидаемого дохода отдельной системы

Обозначим через $v_{cl}^*(k, t)$ полный ожидаемый доход, который получит на l -м интервале времени СМО S_c замкнутой сети с приоритетными заявками за время t , если в начальный момент интервала она находится в состоянии k . В течение малого промежутка времени Δt сеть может остаться в состоянии (k, t) либо совершить переход в состояние $(k + I_{i1} - I_{j1}, t + \Delta t)$, $(k + I_{i2} - I_{j2}, t + \Delta t)$, при этом для простоты будем считать, что $p_{ii}(t) = 0$, $i = \overline{0, n}$. Здесь I_{os} – $2n$ -вектор, состоящий из нулей; I_{is} – $2n$ -вектор с нулевыми компонентами, за исключением компоненты с номером $2(i-1) + s$, которая равна 1, $i = \overline{1, n}$, $s = 1, 2$ [1].

Теорема. Плотность распределения дохода на l -м интервале времени $p_{vcl}^*(x_l, t)$ системы S_c сети удовлетворяет с точностью до членов порядка малости $\varepsilon_l^2 = K_l^{-2}$ дифференциальному уравнению в частных производных

$$\frac{\partial p_{vcl}^*(x_l, t)}{\partial t} = - \sum_{i=1}^n \sum_{s=1}^2 A_{isl}(x_l, t) \frac{\partial p_{vcl}^*(x_l, t)}{\partial x_{isl}} + \frac{\varepsilon_l}{2} \sum_{i,j=1}^n \sum_{s=1}^2 B_{ijsl}(x_l, t) \frac{\partial^2 p_{vcl}^*(x_l, t)}{\partial x_{isl} \partial x_{jsl}} + r_{cl}^*(x_l, t) \quad (2)$$

в точках существования производных, где

$$\begin{aligned} A_{i1l}(x_l, t) &= \sum_{j=1}^n \mu_{j1}(t) p_{ji}^*(t) \varepsilon_{j1}(x_{j1l}, l_{jl}(t)) + \mu_{01}(t) p_{0i}(t) \left(\frac{K_{1l}}{K_l} - \sum_{j=1}^n x_{j1l} \right), \\ A_{i2l}(x_l, t) &= \sum_{j=1}^n \mu_{j2}(t) p_{ji}^*(t) \varepsilon_{j2}(x_{j1l}, x_{j2l}, l_{jl}(t)) + \mu_{02}(t) p_{0i}(t) \left(\frac{K_{2l}}{K_l} - \sum_{j=1}^n x_{j2l} \right), \end{aligned} \quad (3)$$

$$B_{ij1l}(x_l, t) = -\mu_{j1}(t) p_{ji} \varepsilon_{j1}(x_{j1l}, l_{jl}(t)), \quad B_{ii1l}(x_l, t) = \sum_{j=0}^n \mu_{j1}(t) q_{ji}^*(t) \varepsilon_{j1}(x_{j1l}, l_{jl}(t)), \quad (4)$$

$$B_{ij2l}(x_l, t) = -\mu_{j2}(t) p_{ji} \varepsilon_{j2}(x_{j1l}, x_{j2l}, l_{jl}(t)), \quad B_{ii2l}(x_l, t) = \sum_{j=0}^n \mu_{j2}(t) q_{ji}^*(t) \varepsilon_{j2}(x_{j1l}, x_{j2l}, l_{jl}(t)), \quad (5)$$

$$p_{ji}^*(t) = \begin{cases} p_{ji}(t) - 1, & j = i, \\ p_{ji}(t), & j \neq i, \end{cases} \quad q_{ji}^*(t) = \begin{cases} 1 + p_{ji}(t), & j = i, \\ p_{ji}(t), & j \neq i, \end{cases} \quad l_{jl}(t) = \frac{m_j(t)}{K_l}, \quad j = \overline{1, n},$$

$$r_{cl}^*(x, t) = K_l \sum_{i,j=0}^n [\mu_{j1}(t) \varepsilon_{j1}(x_{j1l}, l_{jl}(t)) r_{jicl}^{(1)}(t) + \mu_{j2}(t) \varepsilon_{j2}(x_{j1l}, x_{j2l}, l_{jl}(t)) r_{jicl}^{(2)}(t)] p_{ji}(t) + r_{cl}(t),$$

$$r_{jicl}^{(s)}(t) = K_l^n R_{jic}^{(s)}(t), \quad r_{cl}(t) = K_l^n R_c(t), \quad i, j, c = \overline{1, n}, \quad s = 1, 2, \quad l = \overline{1, q}.$$

Доходы от переходов между состояниями сети $R_{jic}^{(s)}(t)$, $R_c(t)$ определены внутри доказательства.

Доказательство. Положим, что если на интервале времени $[t, t + \Delta t]$ сеть совершает переход из состояния (k, t) в состояние $(k + I_{i1} - I_{j1}, t + \Delta t)$ (это может произойти с вероятностью $\mu_{j1}(t)\varepsilon_{j1}(k_{j1}(t), m_j(t))u(k_{j1}(t))u(K_1(t) - k_{i1}(t))p_{ji}(t)\Delta t + o(\Delta t)$), то доход системы S_c составит $R_{ji1}^{(1)}(t)$, поэтому доход данной СМО в момент времени $t + \Delta t$ будет равен этой величине плюс ожидаемый доход $v_{cl}^*(k + I_{i1} - I_{j1}, t)$, который она получает за оставшееся время t , если бы начальным было состояние $(k + I_{i1} - I_{j1}, t)$. Аналогично, если на интервале $[t, t + \Delta t]$ сеть совершает переход из состояния (k, t) в состояние $(k + I_{i2} - I_{j2}, t + \Delta t)$ с вероятностью $\mu_{j2}(t)\varepsilon_{j2}(k_{j1}(t), k_{j2}(t), m_j(t)) \times u(k_{j2}(t))u(K_2(t) - k_{i2}(t))p_{ji}(t)\Delta t + o(\Delta t)$, то доход системы S_c составит $R_{ji2}^{(2)}(t)$ плюс ожидаемый доход $v_{cl}^*(k + I_{i2} - I_{j2}, t)$, который она получает за оставшееся время t , если бы начальным было состояние $(k + I_{i2} - I_{j2}, t)$. Кроме того, будем считать, что система S_c получает доход $R_c(t)$ за единицу времени в течение пребывания сети в состоянии (k, t) . Сеть остаётся в состоянии (k, t) в течение времени Δt с вероятностью $1 - \sum_{j=0}^n [\mu_{j1}(t)\varepsilon_{j1}(k_{j1}(t), m_j(t)) + \mu_{j2}(t)\varepsilon_{j2}(k_{j1}(t), k_{j2}(t), m_j(t))] \times \Delta t + o(\Delta t)$, при этом доход системы S_c составит $R_c(t)\Delta t + v_{cl}^*(k, t)$.

Из вышесказанного следует, что полный ожидаемый доход $v_{cl}^*(k, t + \Delta t)$ системы S_c в момент времени $t + \Delta t$ удовлетворяет системе разностных уравнений

$$\begin{aligned} v_{cl}^*(k, t + \Delta t) = & \left\{ 1 - \sum_{j=0}^n [\mu_{j1}(t)\varepsilon_{j1}(k_{j1}(t), m_j(t)) + \mu_{j2}(t)\varepsilon_{j2}(k_{j1}(t), k_{j2}(t), m_j(t))] \Delta t \right\} \times \\ & \times (R_c(t)\Delta t + v_{cl}^*(k, t)) + \sum_{i,j=0}^n [\mu_{j1}(t)\varepsilon_{j1}(k_{j1}(t), m_j(t))u(k_{j1}(t))u(K_{i1} - k_{i1}(t))p_{ji}(t)\Delta t \times \\ & \times (R_{ji1}^{(1)}(t) + v_{cl}^*(k + I_{i1} - I_{j1}, t)) + \mu_{j2}(t)\varepsilon_{j2}(k_{j1}(t), k_{j2}(t), m_j(t))u(k_{j2}(t)) \times \\ & \times (K_{i2} - k_{i2}(t))p_{ji}(t)\Delta t (R_{ji2}^{(2)}(t) + v_{cl}^*(k + I_{i2} - I_{j2}, t))] + o(\Delta t). \end{aligned} \quad (6)$$

В силу определения выражения ε_{j1} , ε_{j2} , согласно (1) и определению функции Хевисайда в соотношении (6) можно опустить функции $u(k_{j2}(t))$. Кроме того, в дальнейшем мы будем проводить асимптотический анализ при $K_s \leq N \rightarrow \infty$, $s = 1, 2$, поэтому можно считать, что $u(K_{i1} - k_{i1}(t)) = u(K_{i2} - k_{i2}(t)) = 1$. Учитывая это, при $\Delta t \rightarrow 0$ из (6) получаем систему разностно-дифференциальных уравнений (РДУ) для ожидаемых доходов системы S_c

$$\begin{aligned} \frac{dv_{cl}^*(k, t)}{dt} = & \sum_{i,j=0}^n \left\{ [\mu_{j1}(t)\varepsilon_{j1}(k_{j1}(t), m_j(t)) (v_{cl}^*(k + I_{i1} - I_{j1}, t) - v_{cl}^*(k, t)) + \right. \\ & \left. + \mu_{j2}(t)\varepsilon_{j2}(k_{j1}(t), k_{j2}(t), m_j(t)) (v_{cl}^*(k + I_{i2} - I_{j2}, t) - v_{cl}^*(k, t))] p_{ji}(t) \right\} + \\ & + \sum_{i,j=0}^n [\mu_{j1}(t)\varepsilon_{j1}(k_{j1}(t), m_j(t))R_{ji1}^{(1)}(t) + \mu_{j2}(t)\varepsilon_{j2}(k_{j1}(t), k_{j2}(t), m_j(t))R_{ji2}^{(2)}(t)] p_{ji}(t) + R_c(t). \end{aligned} \quad (7)$$

Перейдем к плотности распределения дохода СМО S_c $p_{vcl}^*(x_l, t)$. Рассматривая случай большого числа заявок $1 \ll K_l < N$ и переходя к вектору относительных переменных $\xi_l^*(t) = \left(\frac{k_{11}(t)}{K_l}, \frac{k_{12}(t)}{K_l}, \frac{k_{21}(t)}{K_l}, \frac{k_{22}(t)}{K_l}, \dots, \frac{k_{n1}(t)}{K_l}, \frac{k_{n2}(t)}{K_l} \right)$ [2, 3], возможные значения которого принадлежат ограниченному замкнутому множеству

$$G_l^* = \left\{ x_l = (x_{1l}, x_{12l}; x_{21l}, x_{22l}; \dots; x_{nl}, x_{n2l}) : x_{isl} \geq 0, s = 1, 2; \sum_{i=1}^n (x_{i1l} + x_{i2l}) \leq 1 \right\},$$

в котором они располагаются в узлах $2n$ -мерной решетки на расстоянии ε_l друг от друга, можем воспользоваться аппроксимацией функции $v_{cl}^*(k, t)$: $K_l^{2n} v_{cl}^*(k, t) = K_l^{2n} v_{cl}^*(x_l K_l, t) = p_{vcl}^*(x_l, t)$, где $p_{vcl}^*(x_l, t)$ – плотность распределения дохода вектора $\xi_l^*(t)$. В качестве начального условия для уравнения (2) можно взять $p_{vcl}^*(x_l, t_0) = p_{vc0l}^*(x_l)$, где $p_{vc0l}^*(x_l)$ – некоторая известная плотность распределения.

Пусть $e_{i1l} = \varepsilon_l I_{i1}$, $e_{i2l} = \varepsilon_l I_{i2}$, $i = \overline{1, n}$, и при $K_l \rightarrow \infty$

$$K_l^{2n} R_{jic}^{(1)}(t) = r_{jic}^{(1)}(t), K_l^{2n} R_{jic}^{(2)}(t) = r_{jic}^{(2)}(t), K_l^{2n} R_c(t) = r_{cl}(t). \quad (8)$$

Умножив обе части (7) на K_l^{2n} , получаем

$$\begin{aligned} \frac{\partial p_{vcl}^*(x_l, t)}{\partial t} = & K_l \sum_{i,j=0}^n \left\{ \left[\mu_{j1}(t) \varepsilon_{j1}(x_{j1l}(t), l_{j1}(t)) \left(p_{vcl}^*(x_l + e_{i1l} - e_{j1l}, t) - p_{vcl}^*(x_l, t) \right) + \right. \right. \\ & \left. \left. + \mu_{j2}(t) \varepsilon_{j2}(x_{j1l}(t), x_{j2l}(t), l_{j1}(t)) \left(p_{vcl}^*(x_l + e_{i2l} - e_{j2l}, t) - p_{vcl}^*(x_l, t) \right) \right] p_{ji}(t) \right\} + r_{cl}^*(x_l, t). \end{aligned} \quad (9)$$

Разложим функции $p_{vcl}^*(x_l + e_{isl} - e_{jsl}, t)$, $p_{vcl}^*(x_l + e_{0sl} - e_{jsl}, t) = p_{vcl}^*(x_l - e_{jsl}, t)$, $p_{vcl}^*(x_l + e_{isl} - e_{0sl}, t) = p_{vcl}^*(x_l + e_{isl}, t)$ в ряд Тейлора в окрестности точки (x_l, t) :

$$\begin{aligned} p_{vcl}^*(x_l + e_{isl} - e_{jsl}, t) &= p_{vcl}^*(x_l, t) + \varepsilon_l \left(\frac{\partial p_{vcl}^*(x_l, t)}{\partial x_{isl}} - \frac{\partial p_{vcl}^*(x_l, t)}{\partial x_{jsl}} \right) + \\ &+ \frac{\varepsilon_l^2}{2} \left(\frac{\partial^2 p_{vcl}^*(x_l, t)}{\partial x_{isl}^2} - 2 \frac{\partial^2 p_{vcl}^*(x_l, t)}{\partial x_{isl} \partial x_{jsl}} + \frac{\partial^2 p_{vcl}^*(x_l, t)}{\partial x_{jsl}^2} \right) + o(\varepsilon_l^2), \\ p_{vcl}^*(x_l - e_{jsl}, t) &= p_{vcl}^*(x_l, t) - \varepsilon_l \frac{\partial p_{vcl}^*(x_l, t)}{\partial x_{jsl}} + \frac{\varepsilon_l^2}{2} \frac{\partial^2 p_{vcl}^*(x_l, t)}{\partial x_{jsl}^2} + o(\varepsilon_l^2), \\ p_{vcl}^*(x_l + e_{isl}, t) &= p_{vcl}^*(x_l, t) + \varepsilon_l \frac{\partial p_{vcl}^*(x_l, t)}{\partial x_{isl}} + \frac{\varepsilon_l^2}{2} \frac{\partial^2 p_{vcl}^*(x_l, t)}{\partial x_{isl}^2} + o(\varepsilon_l^2), \quad i, j, c = \overline{0, n}. \end{aligned}$$

Подставим это разложение в уравнение (9):

$$\begin{aligned} \frac{\partial p_{vcl}^*(x_l, t)}{\partial t} = & \sum_{i=1}^n \left[\sum_{j=1}^n \left\{ \mu_{j1}(t) p_{ji}(t) \varepsilon_{j1}(x_{j1l}(t), l_{j1}(t)) \left[\left(\frac{\partial p_{vcl}^*(x_l, t)}{\partial x_{i1l}} - \frac{\partial p_{vcl}^*(x_l, t)}{\partial x_{j1l}} \right) + \right. \right. \right. \\ & \left. \left. \left. + \frac{\varepsilon_l}{2} \left(\frac{\partial^2 p_{vcl}^*(x_l, t)}{\partial x_{i1l}^2} - 2 \frac{\partial^2 p_{vcl}^*(x_l, t)}{\partial x_{i1l} \partial x_{j1l}} + \frac{\partial^2 p_{vcl}^*(x_l, t)}{\partial x_{j1l}^2} \right) \right] \right\} + \right. \\ & \left. + \mu_{01}(t) p_{0i}(t) \left(\frac{K_{1l}}{K_l} - \sum_{j=1}^n x_{j1l} \right) \left(\frac{\partial p_{vcl}^*(x_l, t)}{\partial x_{i1l}} + \frac{\varepsilon_l}{2} \frac{\partial^2 p_{vcl}^*(x_l, t)}{\partial x_{i1l}^2} \right) + \right. \\ & \left. + \sum_{j=1}^n \left\{ \mu_{j2}(t) p_{ji}(t) \varepsilon_{j2}(x_{j1l}(t), x_{j2l}(t), l_{j1}(t)) \left[\left(\frac{\partial p_{vcl}^*(x_l, t)}{\partial x_{i2l}} - \frac{\partial p_{vcl}^*(x_l, t)}{\partial x_{j2l}} \right) + \right. \right. \right. \end{aligned}$$

$$\begin{aligned}
& + \frac{\varepsilon_l}{2} \left(\frac{\partial^2 p_{vcl}^*(x_l, t)}{\partial x_{i2l}^2} - 2 \frac{\partial^2 p_{vcl}^*(x_l, t)}{\partial x_{i2l} \partial x_{j2l}} + \frac{\partial^2 p_{vcl}^*(x_l, t)}{\partial x_{j2l}^2} \right) \Bigg] \Bigg\} + \\
& + \mu_{02}(t) p_{02}(t) \left(\frac{K_{2l}}{K_l} - \sum_{j=1}^n x_{j2l} \right) \left(\frac{\partial p_{vcl}^*(x_l, t)}{\partial x_{i2l}} + \frac{\varepsilon_l}{2} \frac{\partial^2 p_{vcl}^*(x_l, t)}{\partial x_{j2l}^2} \right) \Bigg] + \\
& + \sum_{j=1}^n \left\{ \mu_{j1}(t) p_{j0}(t) \varepsilon_{j1}(x_{j1l}(t), l_{j1}(t)) \left[-\frac{\partial p_{vcl}^*(x_l, t)}{\partial x_{j1l}} + \frac{\varepsilon_l}{2} \frac{\partial^2 p_{vcl}^*(x_l, t)}{\partial x_{j1l}^2} \right] \right\} + \\
& + \sum_{j=1}^n \left\{ \mu_{j2}(t) p_{j0}(t) \varepsilon_{j2}(x_{j1l}(t), x_{j2l}(t), l_{j1}(t)) \left[-\frac{\partial p_{vcl}^*(x_l, t)}{\partial x_{j2l}} + \frac{\varepsilon_l}{2} \frac{\partial^2 p_{vcl}^*(x_l, t)}{\partial x_{j2l}^2} \right] \right\} + r_{cl}^*(x_l, t).
\end{aligned}$$

Используя (3)–(5), последнее уравнение с точностью до членов порядка ε_l^2 можно записать в виде (2). Терма доказана.

2. Нахождение ожидаемых доходов отдельных систем

Заметим, как следует из (4), (5), что $\frac{\varepsilon_l}{2} \sum_{i,j=1}^n B_{ijsl}(x_l, t) \frac{\partial^2 p_{vcl}^*(x_l, t)}{\partial x_{isl} \partial x_{jsl}} = O(\varepsilon_l^2)$, поэтому из (2) следует, что с точностью до ε_l^2

$$\frac{\partial p_{vcl}^*(x_l, t)}{\partial t} = - \sum_{i=1}^n \sum_{s=1}^2 A_{isil}(x_l, t) \frac{\partial p_{vcl}^*(x_l, t)}{\partial x_{isl}} + r_{cl}^*(x_l, t). \quad (10)$$

Проинтегрируем обе части уравнения (10) по x_l в области G_l^* и разделим обе части на объем области G_l^* , равной $m(G_l^*)$:

$$\begin{aligned}
\frac{1}{m(G_l^*)} \iint \dots \int_{G_l^*} \frac{\partial p_{vcl}^*(x_l, t)}{\partial t} dx_l &= - \frac{1}{m(G_l^*)} \sum_{i=1}^n \sum_{s=1}^2 \iint \dots \int_{G_l^*} A_{isil}(x_l, t) \frac{\partial p_{vcl}^*(x_l, t)}{\partial x_{isl}} dx_l + \\
&+ \frac{1}{m(G_l^*)} \iint \dots \int_{G_l^*} r_{cl}^*(x_l, t) dx. \quad (11)
\end{aligned}$$

Рассмотрим интегралы в правой части (11). Используем интегрирование по частям, а также предположим, что выполняются граничные условия

$$A_{isil}(x_l, t) p_{vcl}^*(x_l, t) \Big|_{x \in \Gamma(G_l^*)} = 0, \quad i = \overline{1, n},$$

где $\Gamma(G_l^*)$ – граница области G_l^* , т.е. $A_{nsl}(x_l, t) p_{vcl}^*(x_l, t) \Big|_{x_{nsl}=0}^{x_{nsl}=1-x_{1sl}-x_{2sl}-\dots-x_{n-1,sl}} = 0$,

$A_{n-1,sl}(x_l, t) p_{vcl}^*(x_l, t) \Big|_{x_{n-1,sl}=0}^{x_{n-1,sl}=1-x_{1sl}-x_{2sl}-\dots-x_{n-2,sl}-x_{nsl}} = 0$, ..., $A_{1sl}(x_l, t) p_{vcl}^*(x_l, t) \Big|_{x_{1sl}=0}^{x_{1sl}=1-x_{2sl}-x_{3sl}-\dots-x_{nsl}} = 0$, которые

означают, что не допускается поток дохода через границу области G_l^* или что в граничных точках области G_l^* поставлены отражающие экраны. Тогда, учитывая, что $\frac{\partial A_{isil}(x_l, t)}{\partial x_{isl}}$ не зависят от x_{jsl} ,

$j = \overline{1, n}$, получаем

$$\frac{1}{m(G_l^*)} \iint_{G_l^*} \dots \int A_{isl}(x_l, t) \frac{\partial p_{vcl}(x, t)}{\partial x_{isl}} dx = - \frac{\partial A_{isl}(x_l, t)}{\partial x_{isl}} \bar{v}_{G_l^* cl}^{pr}(t), \quad i = \overline{1, n},$$

где $\bar{v}_{G_l^* cl}^{pr}(t)$ – среднее по x_l значение дохода системы S_c сети с приоритетными заявками при условии изменения начального состояния x_l в области G_l^* . Применив для обеих частей (11) рассуждения, как в [4], получим следующее уравнение для среднего дохода системы S_c :

$$\frac{d}{dt} \bar{v}_{G_l^* cl}^{pr}(t) = \sum_{i=1}^n \sum_{s=1}^2 \frac{\partial A_{isl}(x_l, t)}{\partial x_{isl}} \bar{v}_{G_l^* cl}^{pr}(t) + \frac{1}{m(G_l^*)} \iint_{G_l^*} r_{cl}^*(x_l, t) dx_l. \quad (12)$$

Из (3) следует, что коэффициент $A_{il}(x_l, t)$ является кусочно-линейной по x_{jl} , $j = \overline{1, n}$, функцией, а коэффициент $A_{i2l}(x_l, t)$ – кусочно-линейной по x_{jl} , x_{j2l} , $j = \overline{1, n}$, функцией, поэтому (12) – это ДУ с разрывной правой частью. Решать его можно путем разбиения фазового пространства G_l^* на ряд областей в зависимости от того, какие значения принимают x_{jl} , x_{j2l} , $j = \overline{1, n}$, и решения уравнения в каждой области.

Отметим, что общий ожидаемый доход сети на каждом интервале времени может быть найден как сумма доходов всех СМО на этом интервале.

Пример. Транспортное предприятие (ТП, система S_2), имеющее большое число автомобилей (заявок), посылает их для перевозки и приемки грузов в различные города (внешняя среда, система S_0), после чего они возвращаются на базу ТП, разгружаются на складе (система S_1), получают новое задание, и процесс продолжается аналогичным образом. В силу того что прибываемые грузы могут быть первоочередного значения, скоропортящимися или, например, прибывать из-за границы железнодорожным транспортом (при этом составы должны быть скорее возвращены отправителю), определенные заявки могут иметь приоритет при обслуживании (разгрузке – погрузке, оформлении документов), при прогнозировании доходов могут быть применены результаты данного раздела.

Состояние сети в данном случае описывается вектором $k(t) = (k, t) = (k_{11}(t), k_{12}(t); k_{21}(t), k_{22}(t))$, $k_{01}(t) = K_1 - k_{11}(t) - k_{21}(t)$, $k_{02}(t) = K_2 - k_{12}(t) - k_{22}(t)$ – числа приоритетных и непероритетных заявок в системе S_0 . Вероятности переходов заявок между системами равны $p_{12} = p_{23} = p_{31} = 1$, остальные $p_{ij} = 0$, $i, j = \overline{1, 3}$. За единицу времени примем один месяц.

$$\begin{aligned} K_1(t) &= \begin{cases} 30, t \in [0, 3), \\ 35, t \in [3, 6), \\ 30, t \in [6, 9), \\ 27, t \in [9, 12], \end{cases} & K_2(t) &= \begin{cases} 27, t \in [0, 3), \\ 30, t \in [3, 6), \\ 25, t \in [6, 9), \\ 30, t \in [9, 12], \end{cases} & \mu_{01}(t) &= \begin{cases} 8, t \in [0, 3), \\ 6, t \in [3, 6), \\ 7, t \in [6, 9), \\ 5, t \in [9, 12], \end{cases} & \mu_{02}(t) &= \begin{cases} 5, t \in [0, 3), \\ 7, t \in [3, 6), \\ 4, t \in [6, 9), \\ 6, t \in [9, 12], \end{cases} \\ \mu_{11}(t) &= \begin{cases} 8, t \in [0, 3), \\ 7, t \in [3, 6), \\ 7, t \in [6, 9), \\ 6, t \in [9, 12], \end{cases} & \mu_{12}(t) &= \begin{cases} 8, t \in [0, 3), \\ 7, t \in [3, 6), \\ 8, t \in [6, 9), \\ 7, t \in [9, 12], \end{cases} & \mu_{21}(t) &= \begin{cases} 7, t \in [0, 3), \\ 6, t \in [3, 6), \\ 4, t \in [6, 9), \\ 4, t \in [9, 12], \end{cases} & \mu_{22}(t) &= \begin{cases} 7, t \in [0, 3), \\ 4, t \in [3, 6), \\ 5, t \in [6, 9), \\ 6, t \in [9, 12], \end{cases} \\ m_1(t) &= \begin{cases} 5, t \in [0, 3), \\ 6, t \in [3, 6), \\ 6, t \in [6, 9), \\ 2, t \in [9, 12], \end{cases} & m_2(t) &= \begin{cases} 6, t \in [0, 3), \\ 5, t \in [3, 6), \\ 7, t \in [6, 9), \\ 4, t \in [9, 12], \end{cases} & R_{0ic}^{(1)}(t) &= \begin{cases} 5, t \in [0, 3), \\ 4, t \in [3, 6), \\ 5, 5; t \in [6, 9), \\ 7, t \in [9, 12], \end{cases} & R_{0ic}^{(2)}(t) &= \begin{cases} 5, t \in [0, 3), \\ 6, t \in [3, 6), \\ 5, 5; t \in [6, 9), \\ 7, t \in [9, 12], \end{cases} \end{aligned}$$

$$R_{lic}^{(1)}(t) = \begin{cases} 6, & t \in [0, 3) , \\ 5, & t \in [3, 6) , \\ 6,5; & t \in [6, 9) , \\ 6, & t \in [9, 12] , \end{cases} \quad R_{lic}^{(2)}(t) = \begin{cases} 7, & t \in [0, 3) , \\ 6, & t \in [3, 6) , \\ 6,5; & t \in [6, 9) , \\ 5, & t \in [9, 12] , \end{cases}$$

$$R_{2ic}^{(1)}(t) = \begin{cases} 7, & t \in [0, 3) , \\ 5, & t \in [3, 6) , \\ 7,5; & t \in [6, 9) , \\ 5, & t \in [9, 12] , \end{cases} \quad R_{2ic}^{(2)}(t) = \begin{cases} 3, & t \in [0, 3) , \\ 2, & t \in [3, 6) , \\ 7,5; & t \in [6, 9) , \\ 3, & t \in [9, 12] , \end{cases}$$

$$R_1(t) = \begin{cases} 4, & t \in [0, 3) , \\ 2,5; & t \in [3, 6) , \\ 2, & t \in [6, 9) , \\ 3, & t \in [9, 12] , \end{cases} \quad R_2(t) = \begin{cases} 5, & t \in [0, 3) , \\ 2,5; & t \in [3, 6) , \\ 4, & t \in [6, 9) , \\ 2, & t \in [9, 12] . \end{cases}$$

Найдем решение уравнения (12) на четырех интервалах $[0, 3)$, $[3, 6)$, $[6, 9)$, $[9, 12]$ в области $G_{il}^* = \{x_l : l_{il}(t) < x_{1sl} \leq 1, \quad l_{2l}(t) < x_{2sl} \leq 1, \sum_{s=1}^2 (x_{1sl} + x_{2sl}) \leq 1\}$, $l = 1, 2$. И пусть $\bar{v}_{G_1^*}^{pr}(0) = 0$. Зависимость среднего дохода системы S_1 от времени представлена на рис. 1.

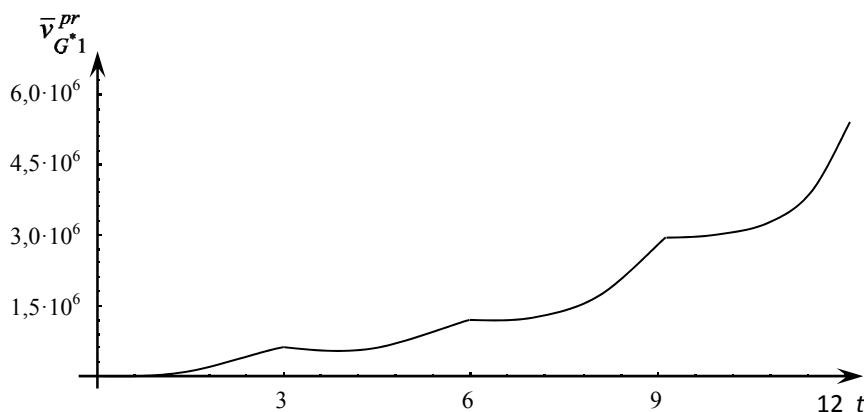


Рис. 1. Средний доход системы S_1

Заключение

Проведен асимптотический анализ замкнутой сети с приоритетными заявками, когда вероятности переходов заявок между СМО и параметры обслуживания зависят от времени, а величины приоритетных и неперитетных заявок являются кусочно-постоянными функциями времени. Получены уравнения в частных производных для плотностей распределения ожидаемых доходов СМО сети и ОДУ для самих ожидаемых доходов при большом числе заявок в сети.

Это дает возможность применить результаты при прогнозировании ожидаемых доходов ЛТС с учетом приоритетности обслуживания, например, погрузки – разгрузки определенных транспортных средств, для которых рассчитаны конкретные примеры.

ЛИТЕРАТУРА

1. Маталыцкий М.А., Русилко Т.В. Математический анализ стохастических моделей обработки исков в страховых компаниях. Гродно : ГрГУ, 2007. 334 с.
2. Медведев Г.А. Об оптимизации замкнутой системы массового обслуживания // Известия АН СССР. Техническая кибернетика. 1975. № 6. С. 65–73.
3. Медведев Г.А. Замкнутые системы массового обслуживания и их оптимизация // Известия АН СССР. Техническая кибернетика. 1978. № 6. С. 199–203.
4. Тихонов В.И., Миронов М.А. Марковские процессы. М. : Сов. радио, 1977. 488 с.

Китурко Ольга Михайловна. E-mail: sytaya_om@mail.ru

Маталыцкий Михаил Алексеевич, д-р физ.-мат. наук, профессор.

E-mail: m.matalytski@gmail.com

Гродненский государственный университет им. Я. Купалы, Беларусь

Поступила в редакцию 02 апреля 2014 г.

Kiturko Olga M., Matalytski Mikhail A. (Grodno State University, Minsk, Belarus).

Asymptotic analysis of incomes in closed HM-network with variable number of priority and non-priority messages.

Keywords: HM-network; expected incomes; method of diffusion approximation; priority messages.

An asymptotic analysis of incomes in the closed exponential HM-network, in which messages are served with an absolute priority and non-priority messages. The number of priority $K_1(t)$ and non-priority $K_2(t)$ messages are piecewise constant functions of time with q intervals of constancy, $K_{sl}(t) = K_{sl}$, $t \in [T_{l-1}, T_l)$, $K_{1l} + K_{2l} = K_l$, $l = \overline{1, q}$. The number of lines in service queueing systems (QS) $m_i(t)$, $i = \overline{1, n}$, $m_0(t) = K_1(t) + K_2(t)$, transition probabilities between messages $p_{ij}(t)$, $i, j = \overline{0, n}$, depend on time. We denote $v_{cl}^*(k, t)$ the expected income, which will receive at l time interval the QS of closed network during time t , if at the initial time the closed network is in a condition k , I_{0s} is the $2n$ -vector consisting of zeros, I_{is} is the $2n$ -vector with zero components and components with the exception of the room $2(i-1) + s$, which is equal to 1, $i = \overline{1, n}$, $s = 1, 2$.

It is shown that the distribution density $p_{vcl}^*(x_l, t)$ of the income of the system S_c at time interval l satisfies, up to terms of order of smallness $\varepsilon_l^2 = K_l^{-2}$, partial differential equations

$$\frac{\partial p_{vcl}^*(x_l, t)}{\partial t} = - \sum_{i=1}^n \sum_{s=1}^2 A_{isl}(x_l, t) \frac{\partial p_{vcl}^*(x_l, t)}{\partial x_{isl}} + \frac{\varepsilon_l}{2} \sum_{i,j=1}^n \sum_{s=1}^2 B_{ijsl}(x_l, t) \frac{\partial^2 p_{vcl}^*(x_l, t)}{\partial x_{isl} \partial x_{jsl}} + r_{cl}^*(x_l, t).$$

Denote by $\bar{v}_{G_l^*}^{pr}(t)$ the average value of income over x_l for the system S_c with priority messages, provided the initial state x changes in the region G_l^* , where $G_l^* = \{x_l = (x_{11l}, x_{12l}; x_{21l}, x_{22l}; \dots; x_{n1l}, x_{n2l}) : x_{isl} \geq 0, s = 1, 2; \sum_{i=1}^n (x_{i1l} + x_{i2l}) \leq 1\}$. For this quantity we received the inhomogeneous ordinary differential equation

$$\frac{d}{dt} \bar{v}_{G_l^*}^{pr}(t) = \sum_{i=1}^n \sum_{s=1}^2 \frac{\partial A_{isl}(x_l, t)}{\partial x_{isl}} \bar{v}_{G_l^*}^{pr}(t) + \frac{1}{m(G_l^*)} \int \dots \int r_{cl}^*(x_l, t) dx_l.$$

The results are applied in forecasting incomes of the carrier enterprise.

REFERENCES

1. Matalytskiy M.A., Rusilko T.V. *Matematicheskiy analiz stokhasticheskikh modeley obrabotki iskov v strakhovykh kompaniyakh* [Mathematical analysis of stochastic models of claims processing in insurance companies]. Grodno: Grodno State University Publ., 2007. 334 p.
2. Medvedev G.A. Ob optimizatsii zamknutoy sistemy massovogo obsluzhivaniya [On the optimization of closed queueing systems]. *Izvestiya AN SSSR. Tekhnicheskaya kibernetika*, 1975, no. 6, pp. 65-73.
3. Medvedev G.A. Zamknutyie sistemy massovogo obsluzhivaniya i ikh optimizatsiya [Closed queueing systems and their optimization]. *Izvestiya AN SSSR. Tekhnicheskaya kibernetika*, 1978, no. 6, pp. 199-203.
4. Tikhonov V.I., Mironov M.A. *Markovskie protsessy* [Markov processes]. Moscow: Sov. Radio Publ., 1977. 488 p.

ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ

УДК 004.652.8

А.М. Бабанов

ДВА СОВРЕМЕННЫХ ПОДХОДА К СЕМАНТИЧЕСКОМУ МОДЕЛИРОВАНИЮ – ORM И ERM

Используя схему Дейта для синтеза семантической модели [1] в качестве методики анализа, автор сравнивает два современных подхода к семантическому моделированию – модель Халпина «Объект – Роль» (OR-модель) [2] и собственную модель «Сущность – Связь – Отображение» (ERM-модель) [3]. Обе эти модели отличаются от предшествующих семантических моделей (ER-модели и ее модификаций) более богатым набором структурных и ограничивающих понятий, позволяющий представить в схеме данных практически всю семантику моделируемого мира. Проведенный анализ показывает, что возможности определения структур данных и ограничений целостности ERM-модели, по крайней мере, не хуже, чем у OR-модели, а в чем-то даже превосходят их.

Ключевые слова: семантическая модель данных; OR-модель, ERM-модель, неформальные семантические концепции; формальные понятия; формальные правила; формальные операторы.

Своим возникновением семантическое моделирование обязано П. Чену и его модели «Entity – Relationship», или, кратко, ER-модели [4]. В указанной статье вместе с представлением новой модели заложил основы семантической методики проектирования схем БД. Эта методика блестяще дополняет классическую методику нормализации, снимая такие проблемы последней, как нетрадиционный для большинства людей и весьма нетривиальный способ восприятия и формализации предметной области (ПрО), практическая неприменимость для сложных ПрО, неоднозначность решения проблемы проектирования, приводящая к прямому перебору многочисленных вариантов схемы в поисках наиболее подходящего. Успех применения семантической методики проектирования БД, помимо знаний и умений человека, определяют два фактора: мощность выразительных средств используемой семантической модели и детальность анализа семантической схемы в применяемых правилах ее трансляции в модель СУБД [5].

Именно поэтому представляет интерес синтез максимально выразительной семантической модели, в рамках которой на этапе первоначальной формализации описаний ПрО возможна фиксация в схеме всех основных понятий ПрО и всех закономерностей взаимоотношений между ними. Идеальная семантическая схема ПрО должна содержать полное формальное описание ПрО с тем, чтобы все последующие межмодельные преобразования схемы носили чисто синтаксический характер и не требовали повторного анализа семантики ПрО и работы с текстами на естественном языке, в которых предварительно зафиксированы выявленные бизнес-правила. Возможность построения таких семантических схем откроет перспективу полной автоматизации процесса проектирования БД, включая генерацию таких элементов реализации, как представления, триггеры и т.д.

Далее в статье будет представлен анализ двух наиболее выразительных семантических моделей – OR-модели и ERM-модели.

1. Подход Дейта к проблеме семантического моделирования

В своей монографии К. Дейт определяет процесс синтеза семантической модели следующим образом. «Общий подход к проблеме семантического моделирования характеризуется четырьмя основными этапами.

1. Прежде всего, попытаемся выявить некоторое множество семантических концепций (понятий), которые могут быть полезны при неформальном обсуждении реального мира.
2. Далее попытаемся определить набор соответствующих символических (т.е. формальных) объектов, которые могут использоваться для представления упомянутых выше семантических концепций.
3. Кроме того, следует определить набор формальных общих правил целостности, предназначенных для работы с такими формальными объектами.
4. Наконец, необходимо также определить набор формальных операторов, предназначенных для манипулирования этими формальными объектами» [1].

2. Неформальные семантические концепции

В поисках неформальных семантических концепций авторы обоих анализируемых моделей обратились к самой мощной семиотической системе – естественному языку. Вот как толкуют их современные логики. Все значащие выражения языка, т.е. выражения, являющиеся знаками, могут быть разбиты на некоторую совокупность классов, называемых семантическими категориями. Семантическая категория – это класс выражений с однотипными предметными значениями, при этом включающий все выражения с предметным значением данного типа.

Имена – слова и словосочетания, являющиеся знаками предметов. По существу, предметом мысли является все то, что обозначено каким-либо именем. Имена бывают единичные, их предметными значениями являются отдельные предметы, и общие. Общее имя является знаком произвольного, любого предмета некоторого класса (определенного вида предметов) и в силу этого как знак представляет в нашем мышлении именно этот класс, который и считается предметным значением общего имени.

Предикаторы – выражения языка (слова и словосочетания), предметными значениями которых являются свойства (одноместные предикаторы) или отношения (многоместные предикаторы). При этом имеются в виду свойства и отношения, которые употребляются как характеристики предметов познания, т.е. наличие или отсутствие чего у предметов мы утверждаем в наших высказываниях.

Предметные функторы – это знаки так называемых предметных функций. Наряду с математическими функциями сюда относят такие характеристики предметов, как скорость, плотность, возраст, пол и т.п. Такие характеристики иногда называют свойствами, хотя это весьма неточно; свойства – это, например, «жидкий», «электропроводный» и т.п. В отличие от свойств предметные функции могут сопоставлять предметам значения из произвольных множеств (числа, строки, даты), а не только истинностные значения Истина и Ложь.

Логические термины – знаки логических отношений (операций), с помощью которых в языке строятся сложные понятия и высказывания. Обобщенно все они могут трактоваться как знаки специфических – логических – функций.

Повествовательные предложения – знаковые формы суждений. Возможными предметными значениями повествовательных предложений являются Истина и Ложь. Смыслом повествовательного предложения является суждение. Суждение, взятое вместе со знаковой формой, в логике принято называть высказыванием.

В логическом анализе языка с целью придания этому анализу большей точности и достижения при этом некоторых обобщений применяется разработанная в логике функциональная трактовка некоторых выражений языка. В логике осуществлено значительное обобщение математического понятия «функция», в силу которого все значимые выражения языка, кроме предложений, единичных имен и их аналогов – переменных, могут трактоваться как функции [6].

Методология мировосприятия каждого человека в той или иной степени совпадает с мнением логики, но, тем не менее, каждый подходит к нему по-своему. Вот мнение Т. Халпина [2, 7], автора ОР-модели, по поводу лежащих в ее основе концепций восприятия моделируемого мира. Поскольку люди естественно обращаются со словами, картинками и примерами, лучший способ достичь ясного описания ПрО – это использовать естественный язык, интуитивно понятные диаграммы и примеры.

По аналогии с термином UML мы называем эти примеры вариантами использования данных, поскольку они являются образцами данных, используемых в системе.

ORM началось в начале 1970-х гг. как семантический подход к моделированию, который воспринимает мир просто как объекты, которые играют роли. Элементарный факт – это простое утверждение или атомарное суждение о том, что конкретный объект играет конкретную роль. Слово «факт» указывает, что утверждение принято истинным пользователями этой ПрО. Мы можем думать о ПрО как о множестве объектов, играющих роли.

Роль, играемую объектом, иногда называют свойством объекта. Тогда элементарный факт утверждает, что некоторый объект имеет некоторое свойство. Это также называют одноместной взаимосвязью, поскольку только одна роль вовлечена в факт [2].

А каковы же семантические концепции, лежащие в основе ERM-моделирования?

Моделируемый мир составляют объекты. Основной информационной (мыслительной) единицей является суждение, которое может быть выражено в языковой форме – высказывании. Причем главную роль (по крайней мере, при проектировании и формировании БД) играют единичные атрибутивные суждения и единичные суждения об отношениях, утверждающие наличие у конкретного объекта определенного свойства или отношения к другим объектам.

Характерным для единичных суждений является использование в качестве логических подлежащих их высказываний единичных имен, предметными значениями которых являются отдельные предметы или объекты (эмпирические или идеальные в зависимости от реальности или идеальности моделируемого мира). Таким образом, имеем одну из основных семантических концепций ERM-модели – объект.

Логическое сказуемое в таких высказываниях может быть задано с использованием одиночного общего имени, предикатора или предметного функтора, либо более сложного логического выражения, включающего эти знаки и логические термины. Все указанные семантические категории могут быть выражены через предметные функции. Последние и представляют вторую основную семантическую концепцию ERM-модели, которую мы будем называть более подходящим термином «отображение». Он, во-первых, не несет математического, количественного смысла образов и прообразов, традиционного для понятия «функция». Во-вторых, с функцией, как правило, ассоциируется требование единственности образа, чего нам также хотелось бы избежать.

Итак, по мнению автора ERM-модели (которое солидаризуется с мнением логиков), для выражения любых суждений о моделируемом мире достаточно двух семантических концепций – объект и отображение. Объект ПрО есть уникальная целостность, которую человек в процессе мировосприятия и мышления способен отличить от всего того, что не является данной целостностью. Отображение ПрО – это некоторый закон предметной области, по которому каждому объекту моделируемого мира (прообразу) может быть поставлен в соответствие (а может быть, и нет) один или более объектов (образов).

Отметим, что пока речь идет о конкретных понятиях «объект» и «отображение», элементами объема которых являются эмпирические или идеальные явления ПрО. Синтез на их основе формальных понятий ERM-модели еще впереди.

3. Формальные понятия

Базовые неформальные семантические концепции играют настолько важную роль, что они часто составляют название модели. Так произошло и с анализируемыми моделями (ERM-модель первоначально называлась OM-моделью – модель «Object – Mapping»). Эти же концепции нашли однозначное отражение в базовых формальных понятиях обеих моделей. Отметим, что базовых понятий OR-модели «объект» и «роль» недостаточно для представления большей части информации. Так, зная только то, что Джон играет роль Муж, а Мэри – роль Жена, нельзя установить факт их супружества. Необходимы дополнительные формальные понятия, например, позволяющие представить тот факт, что «два или более объектов участвуют вместе во взаимосвязи» [2]. А значит, помимо «одно-

местных взаимосвязей», необходимы, по крайней мере, еще и взаимосвязи порядка больше одного. В ERM-модели же понятия «объект» и «отображение» обеспечивают всю выразительную силу языка а остальные формальные понятия являются производными от них. Этому потенциалу они обязаны функциональной трактовке семантических категорий естественного языка.

Итак, сначала рассмотрим систему формальных понятий Халпина [2, 8, 9].

Элементарные факты утверждают, что объекты играют роли. Объекты являются или значениями, или сущностями. Значение – константа, которая самоидентифицируется в смысле, что когда вы видите константу, записанную в некотором контексте, вы всегда знаете то, на что она указывает. В результате на значения можно сослаться непосредственно, без необходимости идентифицировать их описанием.

Концептуально на сущность (например, конкретного человека или автомобиль) ссылаются в информационной системе посредством определенного описания. Например, кенгуру прыгают по сущности, идентифицированной как «Страна по имени 'Австралия'». В отличие от значений, некоторые, а в действительности – большинство объектов могут изменяться со временем. Сущность может быть материальным объектом (например, «Город по имени 'Париж'») или абстрактным объектом (например, «Курс с кодом 'CS 114'»). Мы рассматриваем и сущности и значения как объекты, которые существуют в ПрО.

Теперь, когда мы знаем, как указать объекты, поговорим о том, как мы определяем роли, которые они играют. Мы используем логические предикаты. В логике предикат – это аналог повествовательного предложения с пустотами для объектов в нем. Чтобы окончательно сформировать предложение, надо пустоты для объектов заполнить терминами объектов.

Предикат можно представить как «...болеет», используя многоточие для обозначения пустоты или «ниши» для экземпляра объекта. Мы продемонстрировали одноместный предикат или предложение с одной нишей для объекта. Его можно также назвать свойством или одноместным типом взаимосвязей.

Бинарный предикат – это предложение с двумя нишами для объектов. Большинство предикатов в информационных моделях являются бинарными. Такой предикат можно показать как «...разработал...» с двумя нишами для объектов. Заметим, что порядок, в соответствии с которым объекты будут помещены в него, важен.

Тернарный предикат – предложение с тремя нишами для объектов. N -арный предикат – предложение с n нишами для объектов ($n > 0$). Так как порядок существен, заполненный n -арный предикат связан с последовательностью из n терминов объектов, необязательно различных. Значение n – арность или степень предиката. Элементарный факт утверждает суждение в форме: $R O_1 \dots O_n$, где R – предикат арности n и $O_1 \dots O_n$ – n терминов объектов, не обязательно различных.

При изменении порядка терминов мы полностью меняем предикат. Мы говорим, что предикат «был разработан» – инверсия предиката «разработал». Хотя семантически мы могли бы расценить два предложения, построенные с их участием, как выражения одного и того же факта, синтаксически они различны. Большинство логиков описали бы это как случай двух различных предложений, выражающих одно и то же суждение. Важно не рассмотреть оба предложения как различные, несвязанные факты. Наш подход к бинарным типам фактов состоит в том, чтобы выбрать один первичный способ объявления предиката и опционально позволить инверсное прочтение, например: «The Person with surname 'Wirth' *designed* / *was designed* by the Language named 'Pascal'».

Здесь предикат слева от «/» используется для чтения слева направо. Предикат справа от «/» используется для инверсного чтения. Наличие различных способов говорить о типе фактов может помочь взаимодействию и улучшить спецификацию ограничений. N -арный тип фактов имеет много возможных упорядочений, но обычно показывается только один из них.

В отличие от ER-модели, ORM не использует атрибуты. Все факты представляются в терминах объектов (сущностей или значений), играющих роли. Хотя это часто приводит к большим диаграммам, подход без использования атрибутов имеет преимущества для концептуального анализа, включающие простоту, семантическую стабильность и легкость изменения.

Другая особенность ORM – ее гибкая поддержка специализации, включая множественное наследование, основанная на формальных определениях подтипов [2].

Основные базовые понятия ERM-модели – объект, класс и отображение, взаимно-однозначно соответствующие нашим семантическим концепциям, обеспечивают большую часть выразительной мощи модели. Неудивительно, что именно эти понятия представлены в синтаксисе и аксиоматике формальной системы теории семантически значимых отображений, являющейся математической основой ERM-модели. Однако человек не всегда использует этот весьма абстрактный уровень мышления. Для простоты работы со схемой выделяются частные виды объектов, классов и отображений, образующие множество производных понятий модели.

Объекты, мыслимые в высказываниях о ПрО как предметы, представляют собой сущности, а классы таких объектов есть не что иное, как множества сущностей. Идеальные объекты, такие как числа, даты, строки символов, являются значениями. Они не обладают свойствами, характеристиками и не вступают в отношения с другими объектами, кроме того что являются значениями характеристик объектов. Их объединяют во множества значений по синтаксическим особенностям. Объекты, подпадающие под конкретные понятия об n -ках предметов, представляют собой связи, а классы объектов, составляющих объемы таких понятий, есть не что иное, как множества связей. Каждый объект в n -ке связи играет определенную роль, характеризующую его функцию в этой связи.

Отображения, определяемые множествами связей, которые в качестве областей определения и значений имеют одиночные множества сущностей или их Декартовы произведения, называются реляционными. Роли объектов-прообразов и объектов-образов этих отображений такие же, как и роли сущностей в связях. Общее количество реляционных отображений, определяемых одним множеством связей степени n , равно $2^n - 2$.

Отображение, ставящее в соответствие объекту истинностное значение, называется отображением-свойством. Если в качестве области значений в отображении используется произвольное множество значений, такое отображение будем называть отображением-характеристикой. Отображения-характеристики являются не чем иным, как атрибутивными отображениями или просто атрибутами.

Таким образом, нам удалось связать основные базовые понятия ERM-модели со структурными понятиями модели «Сущность – Связь». Все вместе они составляют понятийный базис ERM-модели. Отметим, что понятие «объект» является обобщением понятий «сущность», «связь» и «значение». Соответственно, понятие «класс» является обобщением понятий «множество сущностей», «множество связей» и «множество значений». А понятия «реляционное отображение» и «атрибутивное отображение» суть специальные виды отображений. Очевидно, в отличие от первых двух последняя специализация неполна.

Следует отметить, что наряду с основными структурными базовыми понятиями – «класс» и «отображение» – для более полного выражения семантики ПрО используются дополнительные базовые понятия – специализация и категоризация, роли объектов-прообразов и объектов-образов в экземплярах отображений, а также операции и отношения (непересекаемости, включения и равенства – для классов, несовместности, следствия и эквивалентности – для отображений), определенные на множествах классов и отображений.

Предлагаемый подход к выбору структурных понятий модели (абстрактных базовых и более простых производных) позволяет при формализации ПрО оперировать в основном знакомыми понятиями («сущность», «связь»), прибегая к использованию новых форм («отображение») лишь в случае недостаточной выразительности первых. В некоторых случаях проектирования схем данных использования понятий «класс» и «отображение» удастся избежать вовсе.

Рассматривая недостаточно выразительные семантические концепции – объект и роль, – Халпин уже на этапе определения формальных понятий своей модели вынужден ввести действительно необходимое ему понятие «предикат» для представления таких важных явлений, как взаимосвязи между объектами. Но тем самым модель теряет индивидуальность и оригинальность, приближаясь по своим базовым понятиям к ER-модели и реляционной модели.

Понимая важность различных упорядочиваний объектов в предикатах, Халпин вводит их в свою модель в виде прочтений предикатов. При этом он ограничивается ролью этих прочтений исключительно для вербализации схемы и данных – наличие различных способов говорить о типе фактов может помочь взаимодействию с экспертом и улучшить спецификацию ограничений. Лежащих в основе этого отображений он не видит, а они открывают дополнительные возможности для формализации бизнес-правил.

4. Формальные правила для задания ограничений целостности

Говоря о типах ограничений целостности, предусмотренных в модели Халпина, отмечается: ER-нотация менее выразительна, чем ORM, по охвату ограничений и бизнес-правил. Отказ ER- и UML-диаграмм включать стандартные нотации для многих важных ORM-ограничений затрудняет создание всесторонней модели или выполнение преобразований модели [2].

Вот какие дополнительные «стандартные» ограничения целостности Халпин предлагает определять на OR-диаграмме. Внутренние ограничения уникальности указывают, какие роли или комбинации ролей одной ассоциации должны иметь уникальные вхождения. Внешние ограничения уникальности действуют аналогично, но применяются к ролям различных предикатов, концептуально соединенных друг с другом.

Ограничение обязательности роли означает, что каждый экземпляр класса, для которого она определена, должен играть эту роль. Роли, для которых отсутствуют такие ограничения, называются необязательными или опциональными.

Ограничение включающего «или» указывает, что дизъюнкция двух или более ролей является обязательной. Ограничение исключения запрещает одновременное включение ролей ассоциаций. Можно определить два ограничения одновременно – ограничение обязательности и взаимной исключительности ролей. Оба эти ограничения вместе обеспечивают так называемое хог-ограничение (исключающее «или»).

Ограничения сравнения множеств ограничивают способ экземпляризации одной роли или последовательности ролей, касающийся экземпляризации других ролей. Рассматриваются следующие отношения, которыми два множества могут быть связаны: подмножество, равенство, исключение [2].

Следует отметить, что указанные ограничения могут применяться как к одной роли, так и к группе ролей или всей ассоциации.

Из оставшихся нерассмотренными OR-ограничений упомянем ограничения на специализации, ограничения частоты появления объекта в роли и кольцевые ограничения.

Отличительной особенностью аппарата ограничений целостности модели Халпина является, то, что для любой полезной особенности данных в нем изобретены свои специальные термин и графический примитив, что приводит к громоздкости определения модели и сложности ее применения.

Теперь посмотрим, как обстоят дела с правилами задания ограничений целостности в ERM-модели.

Первую группу ограничений целостности составляют правила, определяющие области значений отображений-характеристик. Область значений такого отображения можно задать определением синтаксиса составляющих его значений, перечислением допустимых значений (экстенсионалом), операцией сравнения, логическим выражением.

Вторая группа ограничений целостности определяет количественные характеристики (глобальное минимальное кардинальное число, реальное минимальное кардинальное число, максимальное кардинальное число) и тип (неограниченное, полностью определенное, частичное функциональное, полное функциональное) каждого отображения-характеристики и обратного ему отображения.

Третья группа ограничений целостности определяет количественные характеристики и тип реляционных отображений.

Четвертая группа ограничений целостности определяет количественные характеристики и тип отображений, сопутствующих атрибутным отображениям множеств связей.

Пятая группа ограничений целостности определяет получаемые отображения (образуемые с помощью операций над отображениями), а также их количественные характеристики и тип.

Шестая группа ограничений целостности определяет взаимоотношения между классами и между отображениями.

Последняя группа ограничений целостности связана со специализациями. Для каждой из них необходимо указать, является она полной или частичной, пересекающейся или непересекающейся, а также определить отображения, которые являются основаниями выделения подклассов.

Отличительной особенностью аппарата ограничений целостности ERM-модели является то, что, в отличие от OR-модели, в ней предлагается ограниченный универсальный набор конструкций, с помощью которых можно представить многие (если не все) законы взаимоотношения данных. По крайней мере, все типы ограничений целостности OR-модели можно выразить предлагаемыми в ERM-модели средствами.

5. Формальные операторы

Если Халпин вообще не рассматривает операции над данными (по крайней мере, в монографии [2]), то в ERM-модели они играют важную роль еще при построении схемы, поскольку с их помощью определяются производные отображения, которые, так же как и базовые отображения, являются объектом ограничений целостности. Все операции над отображениями составляют алгебру отображений.

Отображение φ^{-1} является инверсией отображения φ тогда и только тогда, когда $\forall x \forall y (y = \varphi(x) \leftrightarrow x = \varphi^{-1}(y))$.

Отображение $\psi(\varphi())$ является композицией отображений φ и ψ тогда и только тогда, когда $\forall x \forall z (z = \psi(\varphi(x)) \leftrightarrow \exists y (y = \varphi(x) \wedge z = \psi(y)))$.

Отображение $\varphi \cup \psi$ является объединением отображений φ и ψ тогда и только тогда, когда $\forall x \forall y (y = (\varphi \cup \psi)(x) \leftrightarrow y = \varphi(x) \vee y = \psi(x))$.

Отображение $\varphi \cap \psi$ является пересечением отображений φ и ψ тогда и только тогда, когда $\forall x \forall y (y = (\varphi \cap \psi)(x) \leftrightarrow y = \varphi(x) \wedge y = \psi(x))$.

Отображение $\varphi - \psi$ является разностью отображений φ и ψ тогда и только тогда, когда $\forall x \forall y (y = (\varphi - \psi)(x) \leftrightarrow y = \varphi(x) \wedge \neg y = \psi(x))$.

Отображение $\varphi[Y]$ является проекцией отображения φ на определяемую роль (или группу ролей) Y тогда и только тогда, когда $\forall x \forall y (y = \varphi[Y](x) \leftrightarrow \exists z (<y, z> = \varphi(x))$. Здесь для упрощения записи мы опустили имена ролей и сложный объект-образ представили упорядоченным кортежем $<y, z>$.

Отображение $\varphi \diamond \psi$ является агрегатом отображений φ и ψ тогда и только тогда, когда $\forall x \forall y \forall z (<y, z> = (\varphi \diamond \psi)(x) \leftrightarrow y = \varphi(x) \wedge z = \psi(x))$.

Из-за ограниченности объема статьи здесь представлена не вся алгебра отображений, и приведенные операции даются в варианте для простых отображений (один объект – прообраз и один объект – образ).

6. Метасхемы моделей

Понятийный базис модели наглядно представляется ее метасхемой. Если обычная схема данных описывает моделируемый мир, то метасхема представляет структуру схемы ПрО, в ней данными являются элементы этой схемы.

Из-за разнообразия специфических базовых понятий полная метасхема OR-модели Халпиным не приводится. Ее фрагменты из [2, 10, 11] представлена на рис. 1 и 2.

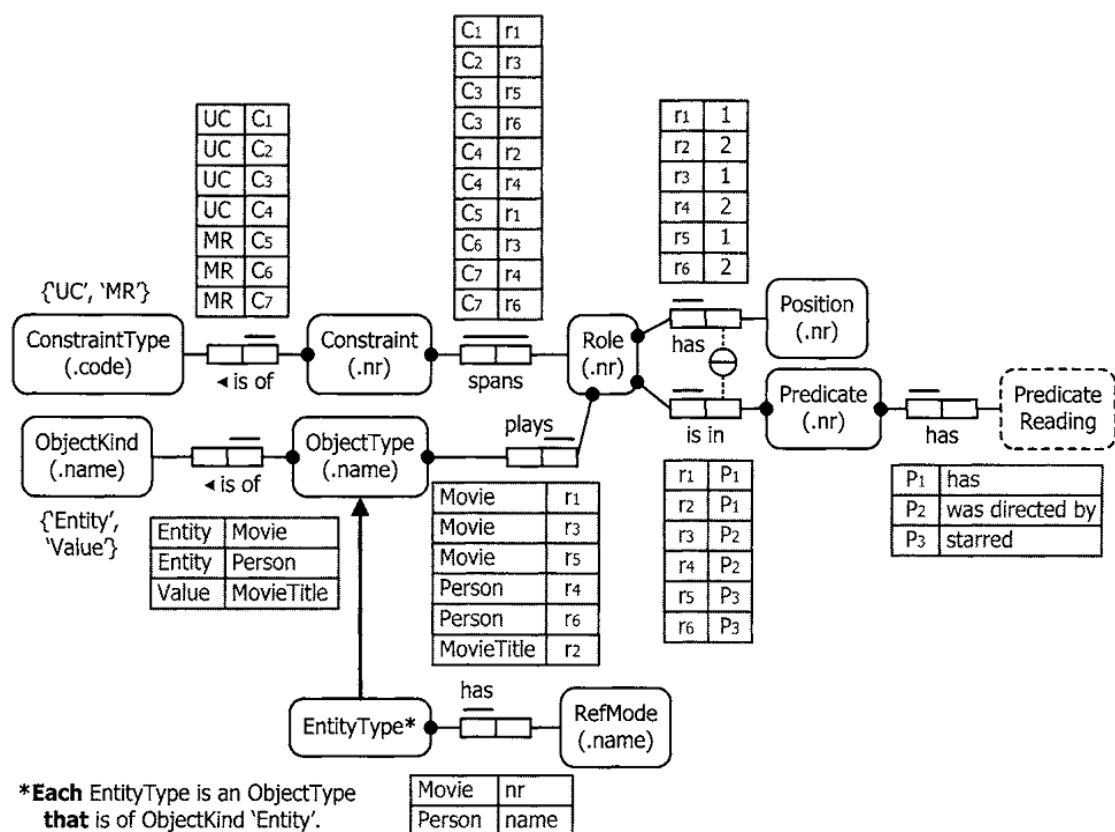


Рис. 1. Фрагмент метасхемы OR-модели (объекты, предикаты и внутренние ограничения целостности); таблицы представляют экстенсионалы ассоциаций

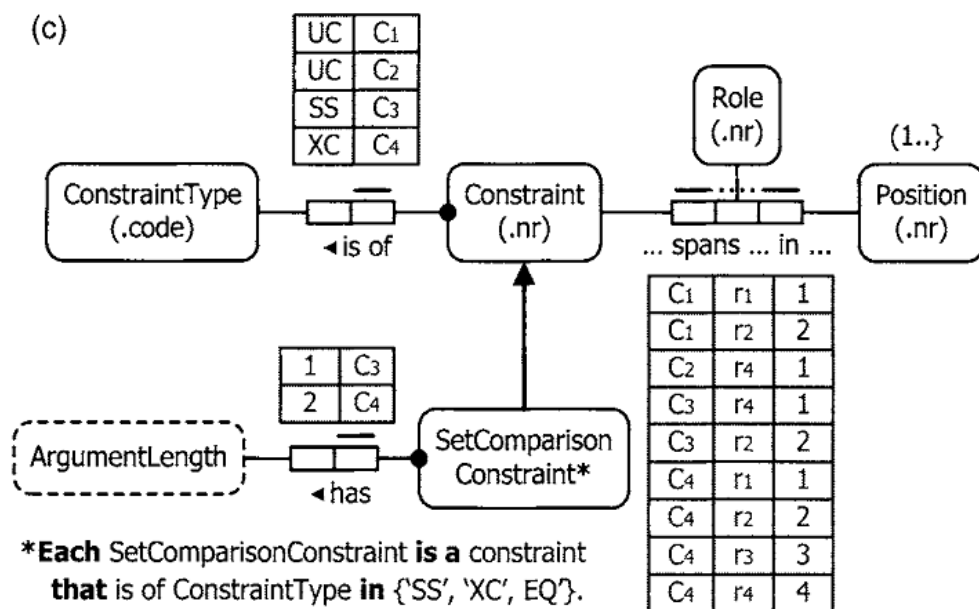


Рис. 2. Фрагмент метасхемы OR-модели (ограничения сравнения множеств)

На рис. 3 представлена полная текущая метасхема ERM-модели. Поскольку модель находится в развитии, постоянно пересматривается и ее метасхема. Серым цветом показаны базовые концепции, белым – производные. Для компактности рисунка опущены атрибуты множеств сущностей и множеств связей метасхемы.

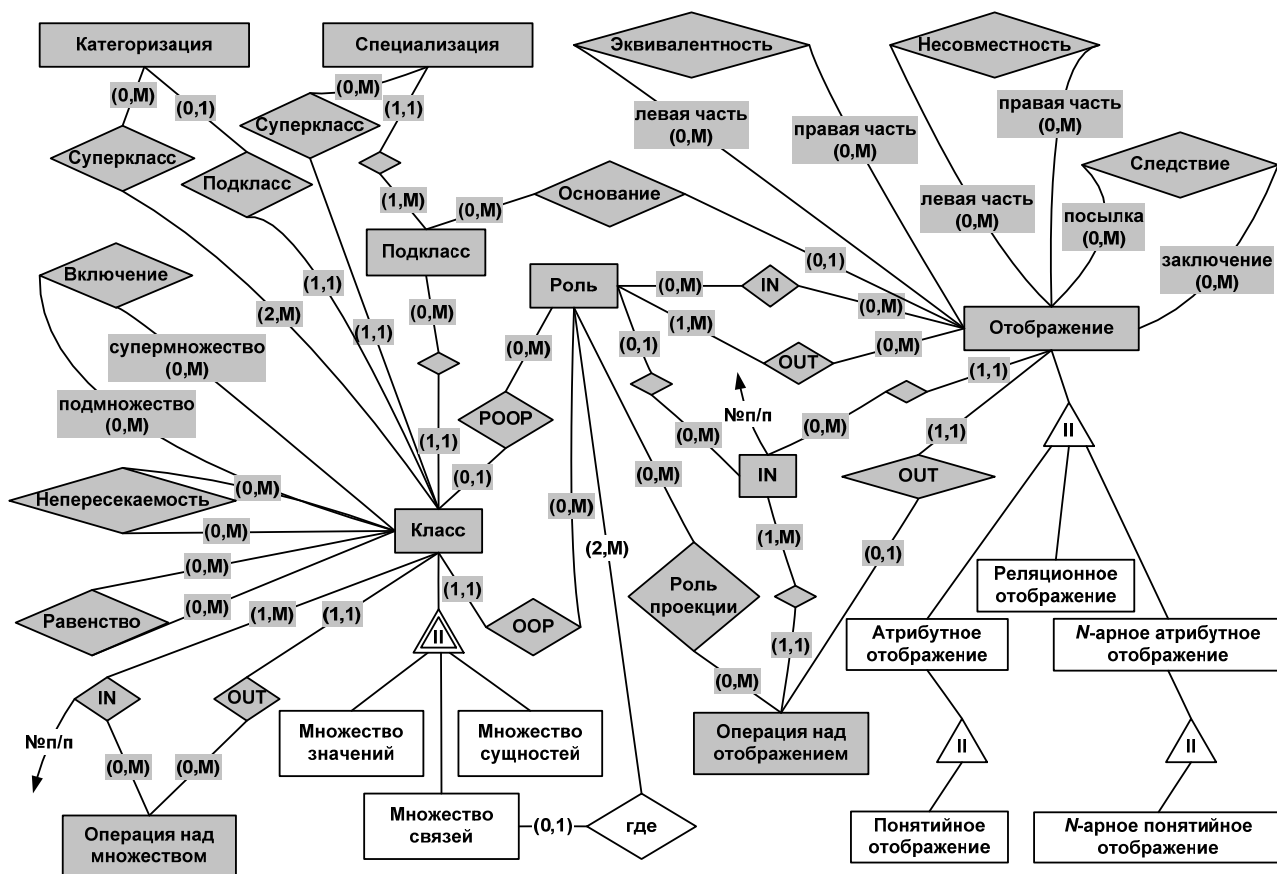


Рис. 3. Полная метасхема ERM-модели

На рис. 3 хорошо видны универсальные структурные и ограничивающие базовые понятия, заимствованные из логики и математики, но это не мешает (или помогает?) ER-модели по своим выразительным возможностям приближаться к естественному языку.

Заключение

Проведенный анализ показывает, что, понимая важность синтеза мощной семантической модели и рассматривая в качестве эталона выразительности естественный язык, авторы ORM- и ERMM-моделей идут различными путями. Если Халпин декомпозирует предикат (или взаимосвязь) степени n на множество n ролей, агрегация экстенционалов которых не восстанавливает экстенционал предиката, то в ERMM-модели вместо типа связей степени n рассматривается $2^n - 2$ типов отображений, каждый из которых эквивалентен одной связи. Вместо n ролей мы имеем $2^n - 2$ отображений, характеристики которых полнее описывают семантику ПрО. Если к этому добавляется возможность определения производных отображений с использованием алгебры отображений, то описательные способности ERM-модели становятся практически безграничными, что, кстати, подтверждается реализацией в ней всех более-менее значащих примеров OR-схем, приведенных в [2].

Богатые выразительные возможности ERM-модели – не результат механистического изобретения специальных конструкций для каждой характерной особенности данных (как это делает Халпин). Основу ERM-модели составляют универсальные понятия логики и математики, сочетая которые, как и в естественном языке, достигается вся ее выразительная мощь.

ЛИТЕРАТУРА

1. Дейт К. Введение в системы баз данных. 7-е изд. : пер. с англ. М. : Вильямс, 2001. 1072 с.

2. Halpin T., Morgan T. Information Modeling and Relational Databases, Second Edition. Morgan Kaufman, 2008. 943 p.
3. Бабанов А.М. Семантическая модель «Сущность – Связь – Отображение» // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2007. № 1. С. 77–91.
4. Chen P.P.-S. The Entity-Relationship Model – Toward a Unified View of Data // ACM Transactions on Database Systems. 1976. V. 1, No. 1. P. 9–36.
5. Бабанов А.М. Семантическая методика проектирования БД и ее перспективы, открывающиеся с применением ERM-модели данных // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2011. № 3(16). С. 58–66.
6. Войцивилло Е.К., Дегтярев М.Г. Логика как часть теории познания и научной методологии (фундаментальный курс) : в 2 кн. М. : Наука, 1994.
7. Halpin T. What is an elementary fact? // Proceedings of First NIAM-ISDM Conference. 1993. Utrecht. P. 1–11.
8. Halpin T. Object-Role Modeling: Principles and Benefits // International Journal of Information Systems Modeling and Design. 2010. No. 1(1). P. 32–54.
9. Halpin T. Object-Role Modeling // Encyclopedia of Database Systems. Springer. 2009. P. 1941–1946.
10. Cuyler D., Halpin T. Two Meta-Models for Object-Role Modeling // Information Modeling Methods and Methodologies. Idea Publishing Group, Hershey PA, USA, 2005. P. 17–42.
11. Halpin T. Comparing Metamodels for ER, ORM and UML Data Models // Advanced Topics in Database Research. Idea Publishing Group, Hershey PA, USA, 2004. V. 3, pt. II. P. 23–44.

Бабанов Алексей Михайлович, канд. техн. наук. E-mail: babanov2000@mail.ru
Томский государственный университет

Поступила в редакцию 4 марта 2014 г.

Babanov Alexey M. (Tomsk State University. Tomsk, Russian Federation).

Two modern approaches to semantic modeling: ORM and ERMM.

Keywords: semantic data model; OR-model; ERM-model; informal semantic concepts; formal structural concepts; formal integrity rules; formal data operators.

Semantic modeling occurrence is obliged to P. Chen and his model «Entity – Relationship». He laid the bases for a semantic technique of the DB schemes designing in his initial article together with the representation of a new model. This technique excellent supplements a classical technique of normalization. Two factors determine a success of semantic technique application (besides designer knowledge and skills): used semantic model expressiveness and minuteness of the semantic scheme analysis in used rules of its translation in the DBMS model.

For this reason, synthesis of the maximum expressive semantic model is of interest. Thus, fixing in the scheme of all basic concepts and all laws of mutual relations between them will be possible within the framework of this model at a stage of initial data domain formalization. The ideal data domain semantic scheme should contain the full formal data domain description, so that all subsequent intermodeling transformations of the scheme had be only syntactic and did not demand the repeated analysis of data domain semantics and works with texts in a natural language, in which the revealed business rules are preliminary fixed. The opportunity of construction of such semantic schemes will open prospect of full automation of DB designing process, including generation of such realization elements as views, triggers, etc.

Using Date's pattern for synthesis of semantic model as a technique of the analysis, the author compares two modern approaches to semantic modeling: Halpin's model «Object – Role» (OR-model) and own model «Entity – Relationship – Mapping» (ERM-model). Both these models are distinguished from previous semantic models with the richer set of the structural and constraining concepts, allowing to present in the data scheme practically all semantics of the modeled world.

In the ERM-model concepts «object» and «mapping» provide all expressive force of language, and other formal concepts are derivatives from them. For this potential they are obliged to functional treatment of natural language semantic categories.

Semantic concepts of OR-model «object» and «role» are not enough for representation of the most part of the information. For this purpose, Halpin already at a stage of model formal concepts definition entered the concept «predicate». But thus the model loses the individuality and originality, coming nearer on the base concepts to the ER-model and relational model.

Understanding the importance of various objects orderings in predicates, Halpin enters them in the model as the perusals of predicates. But he does not go beyond usage of these perusals for verbalization of the scheme and the data. He does not see mappings underlying it, and they open additional opportunities for the formalization of business rules.

Distinctive feature of Halpin's model integrity constraints is, that he is invented the special term and graphic primitive for any useful feature of the data. That leads to the unhandiness of the model definition and complexity of its application. As against the OR-model and ERM-model, we offer the limited universal set of constructions. The basis of ERM-model composes the generalized concepts of logic and mathematics, combined in use, as well as in a natural language, provides all its expressive power.

If Halpin does not consider data operation at all, in ERM-model they play the important role even at the construction of the scheme as with their help derivative mappings are defined, and they are the object of integrity constraints as well as the base mappings.

Understanding the importance of powerful semantic model synthesis and considering as the standard of expressiveness a natural language, authors of the ORM and ERMM go by various ways. If Halpin decomposes the predicate (or relationship) of degree n on

the set of n roles, which extensions aggregation does not restore predicate extension, in the ERMM instead of the relationship type of degree n is considered as $2^n - 2$ types of mappings, which each copy is equivalent to the whole relationship in information sense. Instead of n roles we have $2^n - 2$ mappings which characteristics more full describe semantics. If to this, the opportunity of determining the derivatives of mappings, using algebra maps, is added, then descriptive abilities of ERM-model become practically boundless.

The carried out analysis shows that the opportunities of data structures and integrity constraints definition of the ERM-model are not worse, than ones of the OR-model, and sometimes even surpass them.

REFERENCES

1. Date C.J. *An introduction to database systems*. Addison-Wesley Longman Inc., 2000. 1072 p.
2. Halpin T., Morgan T. *Information modeling and relational databases*. Morgan Kaufman, 2008. 943 p.
3. Babanov A.M. Semantic model "Entity – Relationship – Mapping". *Tomsk State University Journal of Control and Computer Science*, 2007, no. 1(1), pp. 77-91. (In Russian).
4. Chen P.P-S. The entity-relationship model – toward a unified view of data. *ACM Transactions on Database Systems*, 1976, vol. 1, no. 1, pp. 9-36. DOI: 10.1145/320434.320440.
5. Babanov A.M. Semantic method of database designing and its prospects opening with application of the ERM data model. *Tomsk State University Journal of Control and Computer Science*, 2011, no. 3(16), pp. 58-66. (In Russian).
6. Vojshvillo E.K., Degtyarev M.G. *Logika kak chast' teorii poznaniya i nauchnoy metodologii (fundamental'nyy kurs)* [Logic as a part of the theory of knowledge and scientific methodology]. Moscow: Nauka Publ., 1994. (In Russian).
7. Halpin T. What is an elementary fact? *Proc. of First NIAM-ISDM Conference*. Utrecht, 1993, pp. 1-11.
8. Halpin T. Object-Role modeling: Principles and Benefits. *International Journal of Information Systems Modeling and Design*, 2010, no 1(1), pp. 32-54. DOI: 10.4018/jismd.2010092302.
9. Halpin T. Object-Role modeling. *Encyclopedia of Database Systems*, Springer, 2009, pp. 1941-1946. DOI: 10.4018/jismd.2010092302.
10. Cuyler D., Halpin T. *Two meta-models for object-role modeling*. In: Krogstie J., Halpin T., Siau K. (eds.) *Information Modeling Methods and Methodologies*. Idea Publishing Group, Hershey PA, USA, 2005, pp. 17-42.
11. Halpin T. *Comparing Metamodels for ER, ORM and UML Data Models*. In: Siau K. (ed.) *Advanced Topics in Database Research*. Idea Publishing Group, Hershey PA, USA, 2004, pt. II, vol. 3, pp. 23-44.

А.Н. Моисеев, А.М. Политов, В.В. Рахимов, М.О. Хомич

БАЗОВАЯ АРХИТЕКТУРА СИСТЕМЫ СОВМЕСТНОГО РЕДАКТИРОВАНИЯ ДИАГРАММ С СИНХРОНИЗАЦИЕЙ В РЕЖИМЕ РЕАЛЬНОГО ВРЕМЕНИ

Разрабатывается система совместного редактирования диаграмм с синхронизацией в режиме реального времени. Основной задачей является одновременная реализация свойств расширяемости и кроссплатформенности в условиях клиент-серверной архитектуры. Для достижения этих свойств выработана специальная модель описания представления и трехслойная архитектура клиент-серверного взаимодействия на ее основе. Система доработана до состояния архитектурного прототипа и реализует весь критический функционал на базовом уровне.

Ключевые слова: объектно-ориентированная разработка; клиент-серверная архитектура; синхронизация в режиме реального времени.

Диаграммы и различного рода схемы применяются в самых различных областях: при проектировании и разработке ПО, анализе требований, в радиофизике, математике. Кроме того, нередко над одной и той же диаграммой работают несколько людей одновременно (например, при проектировании ПО), а синхронизация с помощью пересылки файлов или с использованием системы контроля версий зачастую сводит преимущества от совместной работы к нулю. Подобное программное обеспечение для совместного редактирования с синхронизацией в режиме реального времени получает все большее распространение в последнее время, в связи с чем растут и требования к подобным системам. Для систем совместного редактирования диаграмм важным свойством является расширяемость – возможность добавлять функционал в систему без изменения ее исходного кода. Вторым немаловажным свойством является кроссплатформенность, т.е. возможность написания клиентского приложения (клиента) под любую платформу. Реализация системы, удовлетворяющей одновременно свойствам расширяемости и кроссплатформенности, связана с определенными техническими трудностями. В [1] описана концепция системы совместного редактирования, удовлетворяющая данным требованиям. В настоящей работе представлено более детальное описание самой концепции.

В области разработки систем совместного редактирования имеется достаточно большое число различных решений. Рассмотрим самые популярные и крупные системы совместного редактирования в аспекте соответствия перечисленным выше свойствам:

1. Документы Google [2] – это веб-ориентированный онлайн-сервис, включающий в себя текстовый и табличный процессор, инструмент для создания презентаций, а также облачное хранилище. Система поддерживает многопользовательское совместное редактирование, однако основным недостатком является очень скудная поддержка редактирования диаграмм без возможности расширения.

2. Google Wave – веб-ориентированный онлайн-сервис для взаимодействия и совместной работы. Система поддерживала совместную работу с форматированным текстом, фотографиями, видео, простыми графическими примитивами. С 30 апреля 2012 г. сервис прекратил свою работу [3].

3. Различные веб-ориентированные сервисы редактирования диаграмм, например LucidCharts [4] или Cасoo [5]. Системы предоставляют возможность многопользовательского совместного редактирования, централизованное хранилище, достаточно активно поддерживаются и развиваются (создаются новые типы диаграмм и элементов, совершенствуется пользовательский интерфейс). Основным недостатком является ориентированность на конкретную клиентскую платформу (веб) и отсутствие поддержки формальной модели редактируемых диаграмм, что можно было бы использовать для поддержки целостной модели проекта).

4. VisualParadigm [6] – мощный инструмент проектирования, обладающий следующими преимуществами: реализация клиентского приложения на языке Java, что позволяет работать на большом числе платформ, поддержка формальной модели редактируемых диаграмм, генерации кода, а также обратного проектирования, расширяемая модель. Основные недостатки: отсутствие совместного редактирования диаграмм с синхронизацией в режиме реального времени, отсутствие поддержки JVM на некоторых современных платформах (например, iOS).

Разрабатываемая авторами система Obordesus спроектирована таким образом, чтобы удовлетворять заявленным выше требованиям расширяемости и кроссплатформенности. В настоящей работе представлены архитектурно значимые решения, принятые при создании системы.

1. Свойства системы

Obordesus [1] – централизованная система совместного редактирования диаграмм с синхронизацией в режиме реального времени. Ее ключевые свойства:

1. Многопользовательская работа с диаграммами – возможность одновременного доступа нескольких участников проекта к одной диаграмме.

2. Синхронизация диаграмм в режиме реального времени – изменения, вносимые одним участником, сразу же отображаются у других участников, редактирующих тот же документ.

3. Расширяемость – возможность добавления функционала к системе без переработки самой системы.

4. Кроссплатформенность – возможность реализации клиентского приложения под любую платформу.

5. Устойчивость клиентского приложения к изменению функционала системы – расширение функционала не требует внесения изменений в клиентские приложения.

6. Централизованность – доступ к документам, созданным с помощью системы, осуществляется через единую глобальную точку, обеспечивая таким образом их повсеместную оперативную доступность.

2. Модель предметной области

Основным элементом предметной области (рис. 1) является диаграмма (объект Document). Диаграммы представляют собой набор визуальных элементов (объект ViewItem), которые являются образами некоторых моделируемых сущностей (объект ModelItem). Моделируемую сущность также будем называть моделью элемента, а его визуальный образ – представлением элемента. Стоит отметить, что несколько визуальных элементов могут представлять одну и ту же модель.

Элементы диаграммы могут быть связаны между собой (например, классы в языке UML [7] могут быть связаны между собой ассоциацией). Связи между элементами рассматриваются как отдельные элементы, которые также имеют свою модель и представление. Модель в данном случае определяет, какие элементы связаны, а представление – как эта связь отображается.

Диаграммы объединяются в проекты (объект Project). Совокупность всех моделей элементов проекта является моделью проекта, а совокупность диаграмм – представлением проекта.

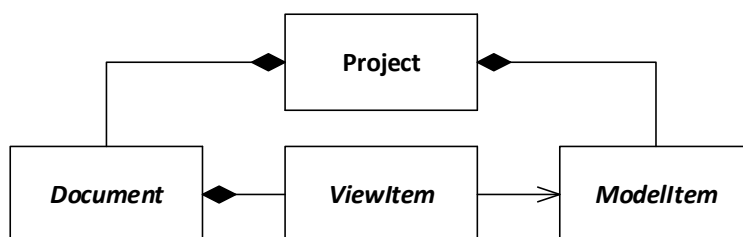


Рис. 1. Модель предметной области

3. Универсальное представление элементов

Одной из задач клиент-серверного взаимодействия [8] в разрабатываемой системе совместного редактирования диаграмм является обеспечение возможности расширения функционала без необходимости модификации клиентских приложений. С учетом требования кроссплатформенности, реализовывать модуль расширения с новым функционалом для каждого клиента слишком затратно. Кроме того, ограничением при выборе решения является также и тот факт, что взаимодействие происходит через сеть, а значит, присутствует некоторая задержка между совершением действия пользователем и реакцией сервера на это действие.

Таким образом, механизм клиент-серверного взаимодействия должен удовлетворять следующим требованиям:

1. Отображение произвольных элементов. Единственным ограничением является набор примитивов, из которых потом будет строиться представление элемента. Этот набор должен быть фиксирован при разработке и быть достаточно полным для представления любых графических элементов.

2. Синхронизация данных между клиентами и сервером. Интерфейс для обмена данными должен быть достаточно абстрактным, чтобы оставаться устойчивым к расширению функционала. Кроме того, особое внимание при проектировании механизма синхронизации необходимо уделять конфликтам синхронизации, вероятность которых очень высока при одновременном редактировании.

3. Взаимодействие пользователя с элементами и элементов между собой. Эта задача появляется из-за ограничения, накладываемого сетевым взаимодействием, так как постоянно присутствует некоторая задержка (до нескольких секунд), во время которой клиент работает самостоятельно, не получая отклика от сервера. В таком случае недостаточно просто уметь отображать элементы, необходимо дублировать взаимодействие между ними на стороне клиента.

Для удовлетворения этим требованиям предлагается следующее высокоуровневое решение (рис. 2): объектная модель, в которой работают клиентские приложения (Client Object Model), должна опираться не непосредственно на серверную объектную модель (Object Model), а на некоторую промежуточную модель описания представления (Presentation Description Model).

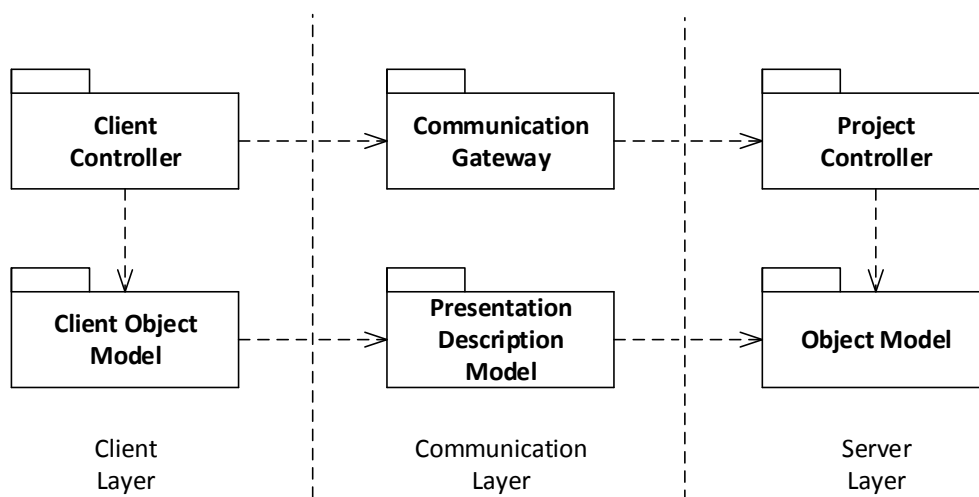


Рис. 2. Трехслойная архитектура системы

Обучение клиента новым типам элементов происходит следующим образом:

1. На сервере генерируется (либо берется готовое) описание представления нового типа элементов.
2. Описание представления доставляется на клиента.
3. Клиент по определенному заранее алгоритму строит свою объектную модель этого типа элементов.

Такое решение позволяет:

- 1) только один раз реализовать серверную модель;
- 2) только один раз специфицировать протокол взаимодействия и модель описания представления;
- 3) для каждой клиентской платформы реализовать модуль преобразования из модели описания представления в объектную модель конкретного клиентского приложения.

В рамках предложенного решения достигается выполнение всех поставленных требований:

1. Отображение произвольных элементов обеспечивается универсальностью модели описания представления.
2. Синхронизация между клиентом и сервером (Client Controller и Project Controller через Communication Gateway) происходит по заранее определенному протоколу в терминах серверной объектной модели, на основе которой построены остальные модели, что позволяет реализовать универсальный механизм синхронизации данных безотносительно к особенностям конкретных элементов.
3. Взаимодействие пользователя с элементами и элементов между собой достигается с помощью особенностей модели описания представления (подробнее см. раздел 7).

5. Применение существующих технологий

В данном разделе приводится обзор существующих технологий, являющихся кандидатами для реализации предложенного решения. Во-первых, существует некоторое количество языков разметки визуальных графических элементов, например SVG [9], VML [10] и др. При использовании этих языков в чистом виде возникают следующие проблемы:

1. Невозможно задать связь представления, описанного с помощью разметки, с серверной моделью.
2. Отсутствие алгоритмической составляющей для организации взаимодействия пользователя с элементами и задания сложной логики представления элементов.

Среди языков разметки интерфейсов существуют решения, позволяющие решить первую проблему (XAML [11], механизм DataBinding [12]), однако в них тоже очень слабо развита алгоритмическая составляющая. Но несмотря на то, что ни один из языков нельзя использовать непосредственно, разумно выделить из них удачные решения и использовать их для задания представления элементов.

Другим вариантом является использование какого-либо скриптового языка (JavaScript, Python) либо языка, исполняемого на виртуальной машине (Java байт-код). Недостатки такого решения:

1. Существуют реализации интерпретаторов (Java-машины в случае с Java) не под все платформы, а сложность реализации интерпретатора любого из таких языков неоправданно высока.
2. Необходимость серьезно ограничивать возможности языка, так как все они являются языками программирования общего назначения, для системы же необходим узкоспециализированный язык. Трудоемкость реализации такого ограничения (особенно при использовании сторонних компонентов для интерпретации) может быть также неоправданно высока.
3. Неудобство использования языков программирования для описания представления. Этот недостаток можно было бы устранить, используя сочетание языка разметки и скриптового языка (пример: HTML+JavaScript), однако это не снимает остальных серьезных ограничений.

Таким образом, более простой в реализации является разработка специального языка описания представления, который будет включать в себя минимально необходимый набор возможностей. С одной стороны, это снимет необходимость реализации ограничения функциональности языка, а с другой – сделает более реальной задачу написания интерпретатора такого языка.

6. Базовые концепции строения элемента диаграммы

По результатам исследования предметной области и функциональных требований к механизму клиент-серверного взаимодействия приняты следующие базовые концепции описания элементов:

1. Информация об элементах содержится в их свойствах.
2. Представление элемента – это функция от свойств данного элемента и, возможно, других элементов.
3. Каждый элемент состоит из трех частей: модели данных, визуальной модели и представления.
4. Представление элементов создается с помощью набора примитивов.

Таким образом, имеется следующая структура элемента: элемент состоит из трех частей (рис. 3), две из которых (Model и VisualModel) являются, по сути, хранилищами значений свойств, инкапсулируя тем самым всю информацию об элементе. Таким образом, для синхронизации информации достаточно использовать только данные этих двух частей элемента. Третья составная часть элемента, Presentation, задается с помощью набора примитивов, значения параметров которых вычисляются на основе значений свойств Model и VisualModel.

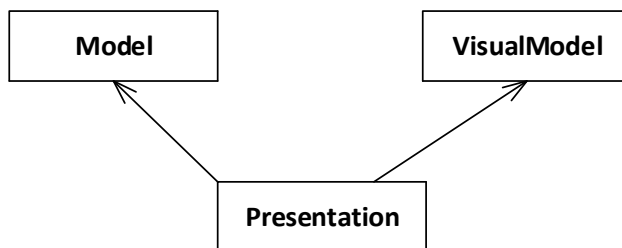


Рис. 3. Структура элемента диаграммы

Такое строение элемента идеально согласуется с моделью предметной области (для этого достаточно объединить VisualModel и Presentation в один блок – View). Кроме того, описанные выше концепции позволяют достичь решения всех трех задач, поставленных перед механизмом клиент-серверного взаимодействия:

1. Отображение произвольных элементов – концептуально нет никаких ограничений на задание набора свойств и набора примитивов, из которых будет строиться описание элемента, т.е. с помощью такого механизма можно построить представление любого элемента.
2. Синхронизация данных между клиентами и сервером – механизм синхронизации сводится к синхронизации свойств Model и VisualModel.
3. Воздействие пользователя на элементы и взаимодействие элементов между собой – фиксированный, заранее определенный набор примитивов позволяет описать указанные сложные взаимодействия в виде обработки параметров этих примитивов с помощью некоторого алгоритмического языка.

7. Модель описания представления

Рассмотрим объектную модель описания представления на высоком уровне (рис. 4).

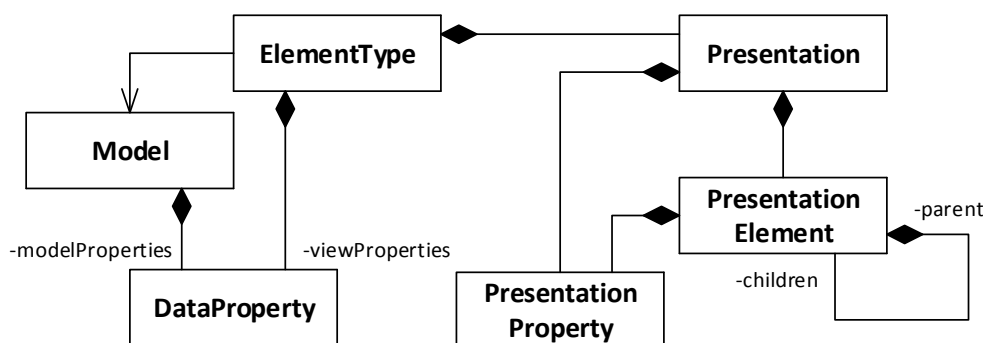


Рис. 4. Модель описания представления

Основным классом является `ElementType`, представляющий типы элементов, из которых в конечном итоге строится визуальное представление диаграммы. Для типа элемента определяется его название. Каждый элемент имеет два набора свойств (`DataProperty`), представляющих свойства модели данных и визуальной модели.

У каждого элемента есть его графическое представление (`Presentation`), которое имеет свой набор свойств (`PresentationProperty`) и строится как композиция атомарных графических элементов (линий, прямоугольников, текстовых меток и т.п.), представленных наследниками класса `PresentationElement`. Некоторые графические элементы могут содержать в себе другие графические элементы, причем из разных соображений: иногда исключительно для удобства (наследование родительских координат), а иногда по структурной необходимости (ячейки вложены в таблицу).

Каждый графический элемент в свою очередь состоит из набора свойств `PresentationProperty`. Представление элемента `Presentation` можно рассматривать как композитный графический элемент, что позволяет использовать `PresentationProperty` для задания свойств как самого представления `Presentation`, так и для задания свойств графических элементов.

Для каждого свойства представления (`PresentationProperty`) задаются 2 функции – `get` и `set`. Функция `get` не обладает побочными эффектами и должна в любой момент времени возвращать актуальное значение свойства. Функция `set` необходима для установки нового значения для свойства в результате действий пользователя и может быть опущена, если свойство не должно редактироваться через пользовательский интерфейс.

8. Серверная архитектура

В серверной части данной системы используется 5 логически разделённых типов серверов (рис. 5).

1. **Root Server** (корневой сервер) отвечает за управление серверами и их взаимодействие друг с другом.
2. **Accounts Server** (сервер пользователей) отвечает за централизованное управление учетными записями пользователей.
3. **Repository Server** (репозиторий) отвечает за централизованное управление проектами.
4. **Login Server** (сервер аутентификации) – пользовательский шлюз для аутентификации и управления проектами.
5. **Project Server** (сервер проектов) отвечает за работу с проектами.

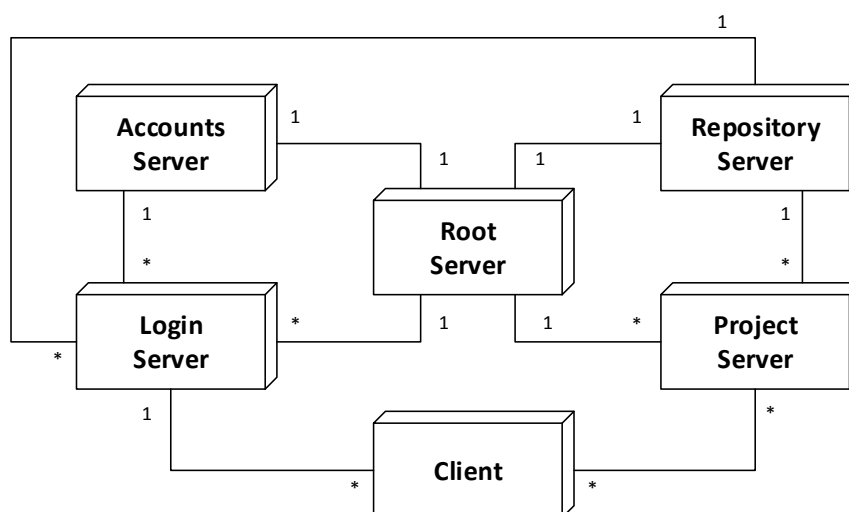


Рис. 5. Взаимодействие серверов и клиентов

Первые три типа серверов составляют управляющую основу серверной части системы и существуют в единственном экземпляре для каждого серверного экземпляра системы. Прямой доступ

пользователей к этим серверам не предусмотрен (за исключением администраторов системы). Последние два типа серверов составляют рабочую часть системы и будут горизонтально масштабироваться при необходимости (развертывание системы планируется на базе так называемых облачных провайдеров, например, Amazon AWS [14], Windows Azure [15]). Пользователи первоначально подключаются к серверу аутентификации, выбор которого происходит автоматически клиентским приложением, затем для работы над конкретными проектами они подключаются к проектному серверу, который выбирается репозиториумом.

Сервер аутентификации позволяет пользователям:

1. Выполнять действия со своим аккаунтом (регистрация, аутентификация, смена учетных данных).
2. Управлять проектами (создание проектов, изменение информации о проектах, предоставление пользователям доступа к проекту, управление лицензиями и т.д.).

Проектный сервер – единственный тип серверов, который работает с серверной расширяемой объектной моделью предметной области. Расширение этой модели выполняется с помощью архитектурного решения «Plugin» [16] путем подключения модулей расширения (плагинов), в результате чего сервер должен удовлетворять следующим требованиям:

1. Возможность изоляции плагинов от проектного сервера и системы, на которой он запущен.
2. Возможность изоляции проектов друг от друга.
3. Возможность в рамках одного сервера иметь подключенными разные модули (в том числе одни и те же модули, но разных версий) для каждого запущенного проекта.

Для реализации этих требований используется изоляция на уровне доменов приложения .NET [17]. Для каждого проекта создается свой домен, в который загружаются только необходимые ему плагины с минимальными правами в системе. Вся обработка изменений выполняется в домене проекта, исключение составляет сетевое взаимодействие – им занимается сам сервер, передавая запросы домену с помощью технологии .NET Remoting [18].

10. Серверная объектная модель

Для серверной реализации расширяемой модели предметной области (рис. 6) необходимо реализовать следующие механизмы:

1. Механизм сохранения и загрузки объектов в данной модели.
2. Механизм транзакций для возможности выполнения сложных модификаций объектов данной модели в рамках транзакций с возможностью отката всех модификаций, если какие-то из них не могут быть применены.
3. Механизм контроля прав доступа к объектам модели.

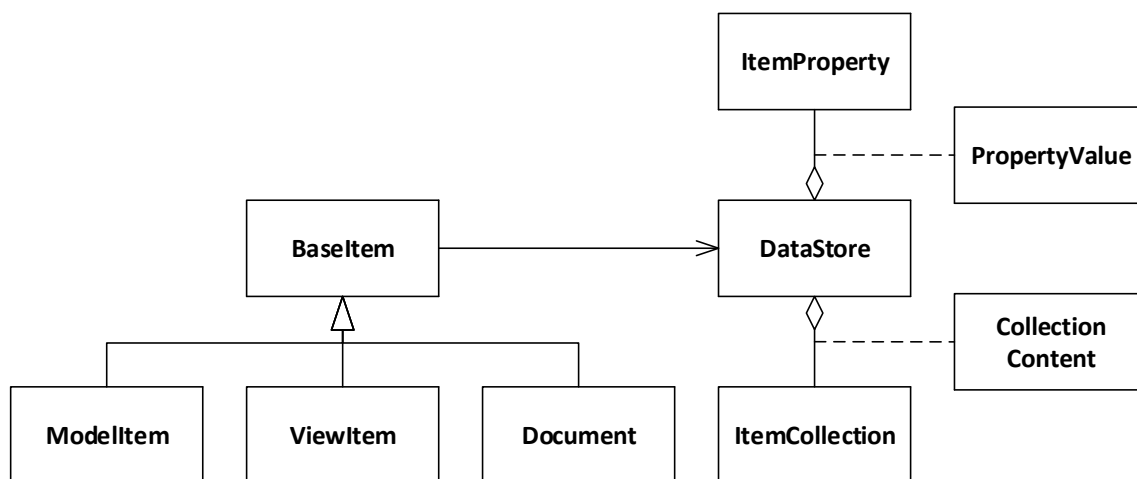


Рис. 6. Серверная объектная модель предметной области

Для реализации этих механизмов существует отдельное хранилище данных элемента (DataStore) (рис. 6), доступ к которому предоставляется через специальные объекты для свойств (ItemProperty) и коллекций (ItemCollection), которые полностью контролируются системой, что позволяет реализовать все указанные выше особенности.

Заключение

На данный момент система доработана до состояния архитектурного прототипа [19], т.е. реализует весь критический функционал на базовом уровне архитектуры. Помимо необходимой серверной инфраструктуры реализовано два клиентских приложения: на платформах Windows (Windows Store app) и Web (HTML + JavaScript). Дальнейшие исследования направлены в сторону интенсивного и экстенсивного расширения модели описания представления. Доступ к ознакомительной версии системы можно получить по адресу: <http://beta.obordesus.com/>

ЛИТЕРАТУРА

1. Моисеев А.Н., Политов А.М., Хомич М.О. Концепция системы поддержки командной разработки программного обеспечения с синхронизацией в режиме реального времени // Информационные технологии и математическое моделирование (ИТММ-2011) : материалы X Всерос. науч.-практ. конф. с междунар. участием (25–26 ноября 2011 г.). Томск, 2011. Ч. 1. С. 69–71.
2. An overview of Google Docs: Google Docs Help / Google. Service information. 2013. URL: <http://support.google.com/drive/bin/answer.py?hl=en&answer=49008> (retrieval date: 13.03.2013).
3. Status of Google Wave: Google Wave Help / Google. Service information. 2013. URL: <http://support.google.com/bin/answer.py?hl=en&answer=1083134> (retrieval date: 13.03.2013).
4. Flow Chart Maker & Online Diagram Software : Lucidchart / Lucid Software Inc. 2013. Service information. URL: <http://www.lucidchart.com/> (retrieval date: 13.03.2013).
5. Cacao: Create diagrams online Real time collaboration / Nulab inc. 2013. Service information. URL: <https://cacao.com/> (retrieval date: 13.03.2013).
6. UML, BPMN and Database Tool for Software Development / Visual Paradigm. 2011. Product information. URL: <http://www.visual-paradigm.com/> (retrieval date: 13.03.2013).
7. ISO/IEC 19505-2:2012(E). Information technology – Object Management Group Unified Modeling Language (OMG UML). Pt. 2: Superstructure. April 2012. International Organization for Standardization, 2012. 740 p.
8. Политов А.М., Рахимов В.В., Хомич М.О. Клиент-серверное взаимодействие в системе поддержки командной разработки с синхронизацией в реальном времени // Материалы 50-й Международной научной студенческой конференции «Студент и научно-технический прогресс»: Информационные технологии. Новосибирск, 2012. С. 42.
9. W3C SVG Working Group / W3C. 2010. Electronic data. URL: <http://www.w3.org/Graphics/SVG/> (retrieval date: 13.03.2013).
10. VML : the Vector Markup Language / J. Bowler, H. Cooperstein, B. Dister, A. Jindal, D. Lee, B. Mathews, T. Sandal, P. Wu. 1998. URL: <http://www.w3.org/TR/NOTE-VML> (retrieval date: 13.03.2013).
11. Mamta Dalal, Ashish Ghoda. XAML Developer Reference. Microsoft Press, December 2011. 342p.
12. Raffaele Garofalo. Building Enterprise Applications with Windows® Presentation Foundation and the Model View ViewModel Pattern. Microsoft Press, March 2011. 226 p.
13. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Design Patterns : Elements of Reusable Object-Oriented Software. Addison-Wesley Professional, 1994. 416 p.
14. Amazon Elastic Compute Cloud (Amazon EC2), Cloud Computing Servers // Amazon. 2013. Service information. URL: <http://aws.amazon.com/ec2/> (retrieval date: 13.03.2013).
15. Cloud Services on Windows Azure // Microsoft. 2013. Service information. URL: <http://www.windowsazure.com/en-us/home/scenarios/cloud-services/> (retrieval date: 13.03.2013).
16. Fowler M. Patterns of Enterprise Application Architecture. Addison-Wesley Professional, 2002. 560 p.
17. Programming with Application Domains and Assemblies // Microsoft. 2013. Technology overview. URL: <http://msdn.microsoft.com/en-us/library/dah4cwez.aspx> (retrieval date: 13.03.2013).
18. .NET Remoting / Microsoft. 2013. Technology overview. URL: [http://msdn.microsoft.com/en-us/library/vstudio/72x4h507\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/vstudio/72x4h507(v=vs.100).aspx) (retrieval date: 13.03.2013).
19. Политов А.М., Рахимов В.В., Хомич М.О. Прототип системы совместного редактирования диаграмм // Материалы XI Всероссийской научно-практической конференции с международным участием «Информационные технологии и математическое моделирование (ИТММ-2012)», Анжеро-Судженск, 23–24 нояб. 2012 г. Кемерово : Практика, 2012. Ч. 2. С. 60–62.

Моисеев Александр Николаевич, канд. техн. наук, доцент. E-mail: moiseev.tsu@gmail.com

Политов Арсентий Михайлович. E-mail: a.m.politov@gmail.com

Рахимов Валентин Валентинович. E-mail: rakhimovvv@gmail.com

Хомич Максим Олегович. E-mail: mohomich@gmail.com

Томский государственный университет

Поступила в редакцию 18 апреля 2014 г.

Moiseev Alexander N., Politov Arseny M., Rakhimov Valentin V., Khomich Maxim O. (Tomsk State University, Tomsk, Russian Federation).

Real-time collaborative diagram editor "Obordesus": concept of unified presentation of elements.

Keywords: collaborative editing; diagram, graphic element; client-server architecture; real-time synchronization

The software for collaborative editing with the real-time synchronization is currently spreading, and therefore the requirements to the functionality of such systems are rising.

Obordesus is a centralized system for collaborative diagram editing with the real-time synchronization. Its main features are the following:

1. Real-time synchronization of currently active diagrams.
2. Extensibility, i.e. an ability to expand functionality without altering the system.
3. Stability of the client to changes in system functional (functional expansion should not affect client code).
4. Remote server, i.e. the main logic, project information and diagrams are stored remotely.

The basic domain area elements include the model entities, their visual images, diagrams (sets of visual images) and projects (sets of model entities and diagrams).

A goal of the system client-server interaction is to allow functional extension without the need to alter the clients. Targeting multiple platforms makes developing extensions for each client time-consuming. Moreover, the communication is held through network, which creates a delay between user's action and server's reaction.

To sum up, the client-server interaction mechanism should allow:

1. Drawing of any arbitrary element.
2. Data synchronization between clients and server.
3. User-element and element-element interaction.

The following higher-level solution meeting the requirements is suggested: client object model is based not on the server object model directly, rather on some kind of intermediate presentation description model.

The presentation description model is based on the following principles:

1. All the information about elements is contained in their properties.
2. Presentation of the element is derived from the properties (of the element itself or of any other elements).
3. Any element consists of three parts: model, visual model and presentation.
4. Presentation of the element is described as a composition of primitive elements.

New presentation description language based on the XML has been developed as the implementation of the listed principles.

Server-side implementation of the extensible domain area model needs the following mechanisms to be implemented:

1. Saving and loading of any specific model objects.
2. Transactions for composite modification of the objects (including rollback support).
3. Access rights control for model objects.

Model extensibility is provided by the extension module (plugin) subsystem on project server. The following requirements apply:

1. Plugins should be isolated from the project server and its hosting system.
2. Plugins should be isolated from each other.
3. The server should allow loading of different plugins (including different plugins versions) for each loaded project.

To meet these requirements the application domain isolation technique is used. Each project is loaded into separate application domain along with the required plugins and with minimal permissions. Model objects operations are executed in the project application domain, while network communication operations are processed by the project server itself.

As of this paper, the basic prototype of the system has been implemented. In addition to critical server-side infrastructure, clients for Windows (Windows Store app) and Web (HTML + JavaScript) have been implemented. Further research is focused on intensive and extensive development of the presentation description model.

REFERENCES

1. Moiseev A.N., Politov A.M., Khomich M.O. [Collaborative real-time software development supply system concept]. *Informatsionnye tekhnologii i matematicheskoe modelirovanie (ITMM-2011): materialy X Vseros. nauch.-prakt. konf. s mezhdunar. uchastiem* [Information technologies and mathematical modelling (ITMM-2011): Proc. of the 10th Russian national research-to-practice conference with international participation]. Tomsk, 2011. Part 1, pp. 69-71. (In Russian).
2. *An overview of Google Docs*: Google Docs Help. Google. Service information, 2013. Available at: <http://support.google.com/drive/bin/answer.py?hl=en&answer=49008>. (Accessed: 13th March 2013).
3. *Status of Google Wave*: Google Wave Help. Google. Service information, 2013. Available at: <http://support.google.com/bin/answer.py?hl=en&answer=1083134>. (Accessed: 13th March 2013).

4. *Flow Chart Maker & Online Diagram Software: Lucidchart*. Lucid Software Inc, 2013. Service information. Available at: <http://www.lucidchart.com/>. (Accessed: 13th March 2013).
5. *Cacoo: Create diagrams online Real time collaboration*. Nulab inc, 2013. Service information. Available at: <https://cacoo.com/>. (Accessed: 13th March 2013).
6. *UML, BPMN and Database Tool for Software Development*. Visual Paradigm, 2011. Product information. Available at: <http://www.visual-paradigm.com/>. (Accessed: 13th March 2013).
7. *ISO/IEC 19505-2:2012(E)*. Information technology – Object Management Group Unified Modeling Language (OMG UML). Part 2: Superstructure. April 2012. International Organization for Standardization, 2012. 740 p.
8. Politov A.M., Rakhimov V.V., Khomich M.O. [Client-server interaction in collaborative real-time software development supply system]. *Materialy 50-y Mezhdunarodnoy nauchnoy studencheskoy konferentsii "Student i nauchno-tehnicheskiy progress": Informatsionnye tekhnologii* [Proc. of the 50th International Students Scientific Conference "Students and Progress in Science and Technology". Information Technologies]. Novosibirsk: Novosibirsk State University Publ., 2012, p. 42. (In Russian).
9. *W3C SVG Working Group*. W3C. 2010. Available at: <http://www.w3.org/Graphics/SVG/>. (Accessed: 13th March 2013).
10. Bowler J., Cooperstein H., Dister B., Jindal A., Lee D., Mathews B., Sandal T., Wu P. *VML: the Vector Markup Language*. 1998. Available at: <http://www.w3.org/TR/NOTE-VML>. (Accessed: 13th March 2013).
11. Mamta Dalal, Ashish Ghoda. *XAML Developer Reference*. Microsoft Press, December 2011. 342p.
12. Garofalo R. *Building Enterprise Applications with Windows® Presentation Foundation and the Model View ViewModel Pattern*. Microsoft Press, 2011. 226 p.
13. Gamma E., Helm R., Johnson R., Vlissides J. *Design Patterns: Elements of reusable object-oriented software*. Addison-Wesley Professional, 1994. 416 p.
14. *Amazon Elastic Compute Cloud (Amazon EC2), Cloud Computing Servers*. Amazon, 2013. Available at: <http://aws.amazon.com/ec2/>. (Accessed: 13th March 2013).
15. *Cloud Services on Windows Azure*. Microsoft, 2013. Available at: <http://www.windowsazure.com/en-us/home/scenarios/cloud-services>. (Accessed: 13th March 2013).
16. Fowler M. *Patterns of enterprise application architecture*. Addison-Wesley Professional, 2002. 560 p.
17. *Programming with Application Domains and Assemblies*. Microsoft, 2013. Available at: <http://msdn.microsoft.com/en-us/library/dah4cwez.aspx>. (Accessed: 13th March 2013).
18. *NET Remoting*. Microsoft, 2013. Available at: [http://msdn.microsoft.com/en-us/library/vstudio/72x4h507\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/vstudio/72x4h507(v=vs.100).aspx). (Accessed: 13th March 2013).
19. Politov A.M., Rakhimov V.V., Khomich M.O. [Collaborative diagram editing system prototype]. *Materialy XI Vserossiyskoy nauchno-prakticheskoy konferentsii s mezhdunarodnym uchastiem "Informatsionnye tekhnologii i matematicheskoe modelirovanie (ITMM-2012)"* [Information technologies and mathematical modelling (ITMM-2012): Proc. of the 11th Russian national research-to-practice conference with international participation]. Kemerovo: Praktika Publ., 2012. Pt 2, pp. 60-62. (In Russian).

ДИСКРЕТНЫЕ ФУНКЦИИ И АВТОМАТЫ

УДК 681.3.06

И.Б. Бурдонов, А.А. Косачев

ИССЛЕДОВАНИЕ ГРАФА ВЗАИМОДЕЙСТВУЮЩИМИ АВТОМАТАМИ

Исследование графа автоматами – корневая задача при верификации программно-аппаратных систем на основе формальных моделей, сводимых к графу переходов. При непрерывном росте используемых систем один компьютер уже не справляется с этой задачей по времени или памяти. Возникает задача параллельной и распределённой верификации, формализуемая как исследование графа взаимодействующими автоматами. Предлагается алгоритм такого исследования.

Ключевые слова: исследование графа; взаимодействующие автоматы; параллельная обработка; распределённые системы; верификация.

Задача обхода неизвестного ориентированного графа автоматом используется во многих приложениях. В данной статье подразумевается тестирование детерминированных систем: граф – это граф автомата тестируемой системы, автомат на графе (автомат-обходчик) – тестирующая система, а проход по дуге – это тестовое воздействие и наблюдение результата [1]. В качестве практического примера можно привести работу [2], где выполнялось функциональное тестирование различных подсистем модели процессора: кэш третьего уровня, управление прерываниями и пр. Модельные графы содержали от нескольких тысяч до нескольких миллионов узлов и несколько миллионов дуг. Тест выполнялся максимально на 150 компьютерах.

Автомат-обходчик выполняется на одной машине (процессор с памятью), а наличие нескольких автоматов на разных машинах позволяет существенно распараллелить работу. Автомат-обходчик начинает работу с начальной вершины графа, и за один такт создаётся не более одного автомата-обходчика.

Нижняя оценка времени обхода для одного или ограниченного числа автоматов равна $\Omega(nm)$, где n – число вершин графа, а m – число дуг [3]. Если число автоматов не ограничено, нижняя оценка $\Omega(m)$.

Для того чтобы автомат мог обходить любой конечный граф, требуется доступ по чтению / записи к неограниченной рабочей памяти, в которой накапливается информация о пройденной части графа. Если эта память – часть памяти автомата (машины), автомат не конечен на классе всех графов. Если автомат один, то существуют алгоритмы с оценкой $\Theta(nm)$ [Там же]. Если число автоматов больше одного, но ограничено, то распараллеливание ускоряет обход, но не меняет порядок времени обхода в наихудшем случае [4]. Если число автоматов не ограничено и все вычисления автоматов и передачи сообщений между ними выполняются не дольше (по порядку), чем проход дуги графа, то существуют алгоритмы с оценкой $\Theta(m)$.

Проблема возникает, когда граф не помещается в память машины, что эквивалентно конечности автомата. Есть два подхода.

Первый подход применим, когда рабочая память существует отдельно от памяти автоматов и реализуется на вершинах графа: автомат может писать / читать из текущей вершины символы конечного алфавита. Такой подход может применяться, например, для сети Интернета, когда вершина – это узел сети, а проход по дуге – передача сообщения между узлами.

Для одного конечного автомата известен алгоритм с оценкой $\Theta(nm+n^2\log\log n)$ [5], а при повторном обходе $\Theta(nm+n^2l(n))$, где $l(n)$ – число логарифмирований, при котором достигается соотно-

шение $1 \leq \log(\log \dots (n) \dots) < 2$ [6]. Отличие от нижней оценки $\Omega(nm)$ объясняется тем, что автомату бывает нужно «вернуться» в начало только что пройденной дуги.

Если конечных автоматов несколько, каждый из них может читать пометки в вершинах, оставленные другими автоматами, и обмениваться с ними сообщениями. Для двух автоматов оценка равна уже $\Theta(nm)$. Эти автоматы двигаются синхронно, кроме случая прохода по новой дуге. В этом случае один автомат (первый) идёт по дуге, а второй остаётся на месте, ожидая сообщения от первого автомата. В этом сообщении указывается, нужно ли второму автомату оставаться на месте, поджидая первый автомат (для возвращения первого автомата в начало только что пройденной дуги), или, наоборот, двигаться вперёд, догоняя первый автомат (возвращения по дуге не требуется).

Второй подход применяется при тестировании, когда вершина графа – это состояние тестируемой системы и автомат ничего не может в неё писать. Разные автоматы могут находиться в разных вершинах графа, т.е. в разных состояниях тестируемой системы. Это означает, что каждый автомат работает со своим клоном тестируемой системы. Будем считать, что за один такт создаётся не более одного клона тестируемой системы с одним автоматом-обходчиком. Тогда рабочая память – это суммарная память коллектива конечных автоматов, обменивающихся сообщениями. Для k машин можно обходить графы, в k раз большие, чем для одной машины. Если размер графа не ограничен, число автоматов в коллективе также должно быть не ограничено.

Этот подход впервые был применён в работе [7], где предложен алгоритм обхода с оценкой $O(m+n^2)$. В настоящей статье мы предлагаем алгоритм с улучшенной оценкой.

1. Формализация автоматов на графе

1.1. Номер выходящей дуги

Автомат на графе, начиная с начальной вершины графа, двигается по дугам, проходя некоторый маршрут. Обход выполнен, если по каждой дуге прошёл хотя бы один автомат. Когда автомат находится в вершине и хочет пройти по выходящей из неё дуге, он эту дугу должен указать. Для этого используется нумерация дуг, выходящих из вершины. Для детерминированных систем разные дуги, выходящие из одной вершины, имеют разные номера. Дуги, выходящие из вершины v , нумеруются от единицы до полустепени выхода вершины (числа выходящих дуг): $1 \dots d_{\text{out}}(v)$. Автомат указывает номер выходящей дуги. Однако для того чтобы выходной алфавит автомата оставался конечным, полустепень выхода вершин графа должна быть ограничена сверху некоторой константой r .

Это ограничение легко снимается с помощью преобразования графа, когда каждая вершина v , в которой полустепень выхода больше r , заменяется на v -цикл вершин. Добавляется $k = \lceil d_{\text{out}}(v)/(r-1) \rceil + 1$ вершин, каждая из которых имеет $r-1$ выходящих дуг, кроме последней вершины, из которой может выходить меньше дуг. Эти вершины связываются дополнительными *внутренними* дугами в цикл, который будем называть v -циклом. Тем самым, автомату, находящемуся в некоторой вершине v -цикла, нужно идентифицировать дугу только в пределах тех r дуг, которые выходят из этой вершины, включая внутреннюю дугу, или меньшего числа дуг для последней вершины. Дуги, заканчивавшиеся в вершине v , теперь будут заканчиваться в первой вершине v -цикла, а остальные вершины v -цикла отличаются тем, что в каждую из них входит ровно одна (внутренняя) дуга. В результате такого преобразования получается граф, в котором полустепень выхода каждой вершины v ограничена r . В дальнейшем для простоты изложения мы будем рассматривать только такие графы, в которых из каждой вершины выходит не более r дуг.

Заметим, что замена вершины v на v -цикл вершин можно понимать как создание одним автоматом, оказавшимся в вершине v , цепочки автоматов – по одному на каждые $r-1$ выходящих дуг или меньшего числа дуг для последнего автомата в цепочке.

1.2. Идентификатор вершины

Для того чтобы автоматы могли обходить граф, автомат, попавший в вершину, должен узнать число выходящих из неё дуг в предположении, что эти дуги занумерованы так, как описано в предыдущем подразделе. Кроме того, чтобы учитывать, какие выходящие дуги в каких вершинах пройдены, необходимо различать вершины. Для этого они снабжаются уникальными идентификаторами. Автомат, попавший в вершину, может узнать её идентификатор. Исключение может быть сделано только для терминальных вершин и вершин, в которые входит только одна дуга; идентификатор этих вершин может быть *пустым*. В первом случае уникальность идентификатора вершины не требуется, поскольку терминальная вершина не имеет выходящих дуг. Во втором случае вершина уникально идентифицируется непустым идентификатором вершины, из которой в данную вершину можно попасть единственным путём, и последовательностью номеров дуг этого пути. Для того чтобы такая вершина с непустым идентификатором всегда существовала, начальная вершина снабжается непустым идентификатором. Заметим, что дополнительные вершины v -цикла, о которых шла речь в предыдущем подразделе, имеют по одной входящей дуге и поэтому могут снабжаться пустым идентификатором.

Поскольку идентификатор вершины является входным символом автомата, для того чтобы автомат был конечным, алфавит идентификаторов должен быть конечным, что означает ограниченное число вершин графа с непустыми идентификаторами.

Для удобства изложения будем рассматривать расширенный автомат (EFSM), память которого хранит управляющее состояние, пробегающее конечный алфавит, и конечное число ячеек памяти, в которых могут храниться идентификаторы вершин и только они: в каждой ячейке по одному идентификатору. Эти ячейки будем называть ячейками идентификаторов (вершин).

Для идентификаторов вершин определены следующие операции и только они:

- скопировать идентификатор из указанной ячейки идентификаторов в сообщение, посылаемое другому автомату, как его параметр;
- скопировать идентификатор как параметр сообщения, полученного от другого автомата, в указанную ячейку идентификаторов;
- сравнить на равенство идентификаторы в двух указанных ячейках идентификаторов;
- проверить, является ли идентификатор в указанной ячейке пустым.

1.3. Адрес автомата и среда связи автоматов

Будем считать, что среда связи автоматов позволяет передавать сообщение от любого автомата любому указанному автомату за время, ограниченное сверху константой. При посылке сообщения автомат должен указать получателя сообщения как уникальный адрес автомата. Для конечного автомата число таких получателей должно быть конечным и ограниченным: адрес автомата должен пробегать конечный алфавит. Последнее ограничение приводит к тому, что число автоматов ограничено.

Мы будем считать, что в памяти расширенного автомата имеется конечное число ячеек памяти для хранения адресов автоматов. Эти ячейки будем называть ячейками адресов (автоматов). Тем самым, граф динамических связей автоматов – это конечный ориентированный граф с ограниченной полустепенью выхода вершин (автоматов). Будем считать, что выделен один *пустой* адрес, который не может быть адресом никакого созданного автомата.

Для адресов автоматов определены следующие операции и только они:

- послать сообщение автомату, адрес которого находится в указанной ячейке адресов;
- скопировать адрес из указанной ячейки адресов в посылаемое сообщение как его параметр;
- скопировать адрес как параметр полученного сообщения в указанную ячейку адресов;
- проверить, является ли адрес в указанной ячейке пустым.

Будем предполагать выполнение следующих правил обмена сообщениями:

- для передачи сообщения есть один примитив «послать сообщение», обязательным параметром которого является адрес получателя;
- для получения сообщений есть один примитив «принять сообщение от любого отправителя»;
- сообщения не теряются и не искажаются в среде передачи;
- при передаче нескольких сообщений от одного и того же отправителя одному и тому же получателю нет обгона сообщений.

1.4. Сообщение

Сообщение, передаваемое от одного автомата другому, является как входным, так и выходным символом автомата. Будем считать, что для расширенного автомата сообщение также «расширено»: оно состоит из *тега*, пробегающего конечный алфавит тегов, и ограниченного конечного числа параметров, которыми могут являться только идентификаторы вершин или адреса автоматов.

1.5. Порождение автоматов

Самый первый автомат порождается некоторым *внешним автоматом* и начинает свою работу с ожидания сообщения от этого внешнего автомата. Для порождения остальных автоматов предлагается использовать сообщение *создай автомат*, посылаемое внешнему автомату. В ответном сообщении *автомат создан* указывается адрес порождённого автомата.

Порождаемый автомат будет начинать работать с начальной вершины графа. Это требование объясняется тем, что мы рассматриваем тестирование системы, которая не допускает произвольное клонирование, а допускает только создание нового экземпляра системы в её начальном состоянии. Исключение составляет создание автомата, которому передаётся граф от другого автомата вместе с текущей вершиной. Передатчик теряет связь с графом.

1.6. Взаимодействие с графом

Предлагается рассматривать граф, точнее его экземпляр для каждого взаимодействующего с ним автомата, как отдельный автомат графа. Такой автомат графа создаётся внешним автоматом, когда ему посылается сообщение *создай граф*. В ответном сообщении *граф создан* содержатся адрес (автомата) графа, идентификатор начальной вершины и число дуг, выходящих из начальной вершины. Созданный автомат графа находится в состоянии, которое соответствует начальной вершине графа.

Далее потребуются только одна операция по взаимодействию с графом, которая реализуется как сообщение, посылаемое автомату графа, и ответное сообщение:

1. *Проход по дуге* графа с параметром: номер выходящей дуги.
2. Ответное сообщение *ответ на проход* с параметрами: идентификатор вершины и число выходящих дуг. Имеются в виду идентификатор конечной вершины указанной дуги и число дуг, выходящих из этой вершины.

1.7. Замечание о ячейках автомата и параметрах сообщений

Если идентификатор вершины занимает a битов памяти, а число вершин в графе с непустым идентификатором равно n_1 , то $2^a \geq n_1$. Если адрес автомата занимает b битов памяти, то сгенерировано может быть не более 2^b автоматов. В описываемом ниже алгоритме генерируется не более $2n_2 + m$ автоматов, где n_2 — число нетерминальных вершин графа, а m — число дуг графа. Тем самым, $2^b \geq 2n_2 + m$. Огрубляя, достаточно ограничений $2^a \geq n$ и $2^b \geq 2n + m$, где n — число вершин графа.

Для удобства изложения будут использоваться дополнительные ячейки и параметры фиксированного размера, кроме тех, что хранят идентификаторы вершин и адреса автоматов.

2. Идея алгоритма

2.1. Граф

Рассматривается обход ориентированного графа с выделенной начальной вершиной и пронумерованными выходящими из каждой вершины дугами. Все вершины снабжены идентификаторами, как было описано в п.1.2. *Началом* и *концом* дуги для краткости будем называть соответственно начальную и конечную вершины дуги. *Терминальной дугой* будем называть дугу, заканчивающуюся в терминальной вершине.

После инициализации в каждый момент времени имеются:

- *Пройденный граф* – подграф графа, определяемый всеми пройденными дугами.
- *Пройденное дерево* – ориентированный от корня, которым является начальная вершина, остов подграфа, получаемого из пройденного графа удалением терминальных дуг. Такое пройденное дерево содержит все нетерминальные пройденные вершины пройденного графа.

В начале работы алгоритма, после инициализации, пройденный граф совпадает с пройденным деревом и состоит из одной начальной вершины графа.

В процессе работы алгоритма пройденный граф и пройденное дерево увеличиваются (добавляются новые дуги и вершины). Пройденное дерево, получающееся в конце работы, будем называть *законченным деревом*. Под *хордой* будем понимать нетерминальную дугу, не принадлежащую законченному дереву; пройденная хорда является хордой пройденного дерева. Для данной вершины *входящей* дугой будем называть дугу законченного дерева, заканчивающуюся в этой вершине. У начальной вершины нет входящей дуги, а в любую другую вершину входит ровно одна дуга.

В конце работы алгоритма получается разметка графа: для каждой дуги в её начале указывается её тип: дуга законченного дерева, терминальная дуга или хорда.

2.2. Работа автоматов

Автомат работает в трёх режимах: генератор, регулятор, движок. Обход графа, т.е. движение по его дугам, выполняют движки, регуляторы предназначены для управления перемещением движков по дугам, а генератор создаёт новые движки и связанные с ними копии графов.

Каждый создаваемый генератором движок предназначен для того, чтобы, начиная с начальной вершины графа, пройти путь в пройденном дереве, после чего пройти непройденную дугу. С каждой пройденной нетерминальной вершиной связан ровно один регулятор, который направляет движки, приходящие в эту вершину, по тем или иным выходящим из вершины дугам. Для этого движок, оказавшийся в данной вершине, спрашивает у регулятора этой вершины (посылает ему сообщение) *куда идти*, а регулятор в ответном сообщении *иди по дуге* сообщает номер выходящей дуги. В некоторой ситуации (описанной ниже) движок оказывается «лишним» и останавливается, не пройдя ни одной непройденной дуги.

Автомат (копии) графа, связанный с движком, «помнит» текущую вершину графа, в которой находится движок. Для прохода по дуге движок посылает автомату графа сообщение *проход по дуге*, задавая номер выходящей дуги, и получает в сообщении *ответ на проход* идентификатор конца дуги и число выходящих из него дуг.

Вначале извне создаётся генератор. После инициализации генератор одновременно становится регулятором начальной вершины, которая в этот момент времени является единственной пройденной вершиной и единственной вершиной пройденного дерева, а также создаёт первый движок, находящийся в начальной вершине графа.

Для управления обходом графа используются сообщения *запрос* и *конец*. Такое сообщение посылается «по пройденной дуге» в обратном направлении: от автомата, находящегося в конце дуги, регулятору начала дуги. Сообщение *запрос* посылается регулятором вершины по входящей дуге и означает, что по этой дуге нужно послать один движок, когда будет такая возможность. Следующий

движок по этой дуге можно посылать после того, как будет получен новый **запрос**. Сообщение **конец**, посылаемое по пройденной дуге, означает, что по этой дуге больше никогда не нужно посылать движки. Сообщение **конец** посылает по хорде или терминальной дуге движок, который эту дугу проходит. Кроме того, сообщение **конец** посылает регулятор вершины по входящей дуге, если в пройденном дереве выше этой вершины нет вершин, из которых выходят непройденные дуги. Это происходит тогда, когда по всем дугам, выходящим из вершины, получено сообщение **конец**. У начальной вершины нет входящей дуги, поэтому регулятор начальной вершины посылает сообщение **конец** внешнему автомату, что означает конец работы алгоритма. **Запросы** посылаются с некоторым опережением, поэтому может оказаться, что движку, пришедшему в вершину, некуда двигаться. Такой движок становится *ждущим* в вершине (запоминается регулятором вершины). Далее либо по выходящей дуге придёт **запрос**, и движок будет послан по этой дуге, либо по всем выходящим дугам будут получены сообщения **конец**, и тогда движок оказывается «лишним» – он должен остановиться.

Каждая дуга имеет в регуляторе её начала три состояния:

- **Активная** = дуга, по которой нужно посылать движки. Это может быть либо непройденная дуга (такие дуги всегда активны), либо дуга пройденного дерева, по которой пришёл **запрос**, но движок по дуге ещё не послан.
- **Пассивная** = пройденная дуга, по которой пока не нужно посылать движки. Это дуга, по которой был послан движок, но после этого по дуге не приходили сообщения **запрос** или **конец**.
- **Законченная** = пройденная дуга, по которой больше никогда не нужно посылать движки. Это дуга, по которой пришло сообщение **конец**.

На рис. 1 изображена схема изменения состояния дуги под влиянием сообщений. В начале все дуги не пройдены, т.е. они активны.

Регулятор посылает движки по активным выходящим дугам «по кругу». Для этого у регулятора есть номер текущей выходящей дуги i . Это дуга, по которой последний раз посылался движок, или $i = 0$ в самом начале. Движок, находясь в вершине пройденного дерева, спрашивает у регулятора вершины, по какой дуге ему идти, посылая сообщение **куда идти**. Регулятор корректирует номер i текущей дуги: новой текущей дугой становится следующая после дуги i активная дуга в цикле выходящих дуг, а если активных дуг больше нет, то $i = 0$. Если после коррекции $i > 0$, регулятор посылает движку сообщение **иди по дуге** с номером дуги i , а сам по входящей дуге посылает **запрос**. У начальной вершины нет входящей дуги, поэтому сообщение **запрос** не посылается, а генератор продолжает генерацию движков. Если $i = 0$, регулятор запоминает адрес движка, движок становится *ждущим*, **запрос** не посылается. Если *ждущий* движок образуется в начальной вершине, генератор приостанавливает генерацию движков.

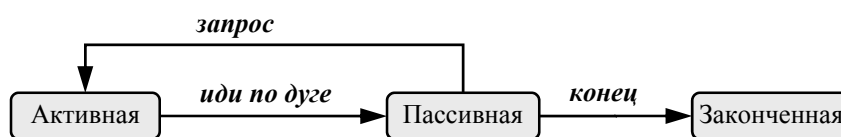


Рис. 1. Схема изменения состояния дуги

После того как появляется *ждущий* движок, возможны два случая: 1. Регулятор в дальнейшем получает сообщение **запрос** с указанием дуги, по которой регулятор направляет *ждущий* движок (сообщение **иди по дуге**) и сам посылает сообщение **запрос** регулятору начала входящей дуги. У начальной вершины нет входящей дуги, поэтому сообщение **запрос** не посылается, а генератор возобновляет генерацию движков. 2. Регулятор обнаруживает, что все выходящие дуги стали законченными (получено сообщение **конец** по последней незаконченной выходящей дуге). Регулятор посылает *ждущему* движку сообщение **иди по дуге** с нулевым номером дуги (это «лишний» движок и он останавливается) и сообщение **конец** регулятору начала входящей дуги. У начальной вершины нет входящей дуги, поэтому сообщение **конец** посылается внешнему автомату и генератор прекращает генерацию движков.

Как создаются регуляторы? Рассмотрим подробнее работу автоматов, связанную с проходом движком непройденной ранее дуги $a-i \rightarrow b$, ведущей из вершины a , имеющей в ней номер i и ведущей в вершину b .

- Если вершина b терминальная, для неё не создаётся регулятор. Движок посылает регулятору вершины a сообщение **конец** с признаком «терминальная вершина» и номером дуги i , а сам останавливается.

- Если вершина b нетерминальная, то движок должен узнать по идентификатору вершины, пройдена она или ещё нет, т.е. имеется ли регулятор этой вершины. Вершина b заведомо не пройдена, если её идентификатор пуст (в такую вершину входит только одна дуга). В противном случае производится опрос регуляторов, описанный ниже. Итак:

- Если вершина b уже пройдена, то движок прошёл по хорде. Движок посылает регулятору вершины a сообщение **конец** с признаком «хорда», номером дуги i и адресом регулятора вершины b , а сам останавливается.
- Если вершина b ещё не пройдена, то движок сам становится регулятором вершины b и создаёт новый движок в текущей вершине b , передавая ему граф. После того как этот новый движок будет создан и послан по выходящей дуге, новый регулятор вершины b посылает регулятору вершины a сообщение **запрос** с указанием номера дуги i и своим адресом как адресом регулятора вершины b .

Опрос регуляторов выполняется тогда, когда движок проходит по непройденной дуге и оказывается в нетерминальной вершине с непустым идентификатором. В этом случае ему нужно найти регулятор той вершины, в которой он оказался, или самому стать регулятором, если поиск не увенчался успехом. Этот поиск выполняется по идентификатору вершины: две вершины не могут иметь один и тот же непустой идентификатор. Для опроса регуляторов используется односторонний список регуляторов вершин с непустыми идентификаторами, регулятор начальной вершины (он же генератор) всегда находится в голове списка (напомним, что идентификатор начальной вершины всегда не пустой). Движок посылает по списку сообщение **опрос**, в котором указываются (непустой) идентификатор вершины и адрес движка. Если регулятор вершины уже есть, он, получив такое сообщение, посылает **ответ на опрос**, указывая движку свой адрес. Если регулятора вершины ещё нет, то сообщение **опрос** доходит до последнего в списке регулятора, который пересылает его движку, а сам ставит у себя ссылку по списку на этот движок. Получив обратно своё сообщение **опрос**, движок сам становится регулятором – последним в списке, и создаёт новый движок, передавая ему свой адрес (как адрес регулятора текущей вершины) и адрес графа.

3. Оценка сложности алгоритма

Будем считать, что времена передачи сообщения, прохода движка по дуге и срабатывания автомата ограничены сверху и снизу ненулевым положительным значением.

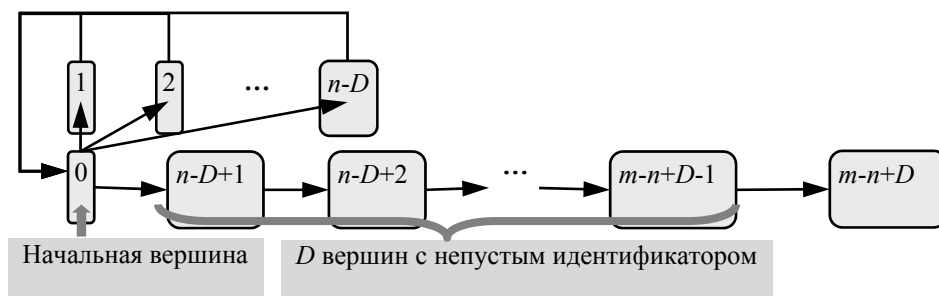


Рис. 2. Пример графа, где алгоритм работает время $\Omega(m+nD)$

Тогда алгоритм заканчивает работу за время $O(m+n_0D)$, где m – число дуг в исходном графе, n_0 – число нетерминальных вершин с непустым идентификатором, D – максимальное число нетерми-

нальных вершин с непустым идентификатором на пути в графе от начальной вершины, не считая самой начальной вершины. Поскольку $n_0 \leq n$, верна также оценка $O(m+nD)$. Эта оценка достигается на графе, изображённом на рис. 2. Поэтому суммарная оценка $\Theta(m+nD)$.

Заключение

Дальнейшие исследования обхода графов коллективом автоматов возможны по нескольким направлениям.

Первое направление: недетерминированные графы. Граф недетерминирован, если одному номеру выходящей дуги соответствует несколько дуг с общей начальной вершиной и разными конечными вершинами. Обход такого графа может пониматься двояко: полный обход, когда требуется проход по каждой дуге графа, или частичный обход, когда нужно пройти хотя бы по одной дуге с данной начальной вершиной и данным номером выходящей дуги. Частичный обход применяется, когда интересующие нас свойства графа могут быть исследованы при таком обходе. Для того чтобы гарантировать возможность обхода недетерминированного графа, нужны дополнительные гипотезы. Например, гипотеза об ограниченном недетерминизме, которая утверждает, что, проходя t раз из данной вершины по выходящей дуге с данным номером, мы пройдем все выходящие из этой вершины дуги с этим номером, где t – заранее известная константа. Существуют алгоритмы такого обхода одним автоматом [8]. Можно поставить задачу разработки нового алгоритма такого обхода коллективом взаимодействующих автоматов. Для частичного обхода может использоваться так называемая Δ -достижимость вершин графа из начальной вершины [9]. Неформально это означает, что существует алгоритм, который гарантированно обеспечивает достижение заданной вершины при любом недетерминированном выборе выходящей дуги в рамках множества дуг с тем же началом и тем же номером дуги.

Второе направление: использование внешней памяти. Предложенный в данной статье алгоритм обходит граф только в том случае, когда регуляторы всех нетерминальных вершин помещаются в суммарной памяти машин. Если это не так, становится актуальным вопрос об использовании дополнительной внешней памяти. Обход графа коллективом автоматов с использованием внешней памяти лежит на стыке двух направлений: методы эффективной работы с внешней памятью, в частности стратегий страничной и сегментной подкачки, и алгоритмы обхода графов коллективом автоматов.

Третье направление: графы специального вида. Большой интерес представляет разработка новых алгоритмов исследования графов, принадлежащих тому или иному подклассу графов. Такие подклассы имеет смысл рассматривать при сочетании двух условий: эти подклассы имеют практическое значение (графы переходов исследуемых систем и сетей относятся к такому подклассу) и на этих подклассах алгоритмы могут работать быстрее.

ЛИТЕРАТУРА

1. Бурдонов И.Б., Косачев А.С., Кулямин В.В., Петренко А.К. Подход UniTesK к разработке тестов // Программирование. 2003. № 6. С. 25–43.
2. Demakov A., Kamkin A., Sortov A. High-Performance Testing: Parallelizing Functional Tests for Computer Systems Using Distributed Graph Exploration. In Open Cirrus Summit 2011. Moscow, 2011.
3. Бурдонов И.Б., Косачев А.С., Кулямин В.В. Избыточные алгоритмы обхода ориентированных графов. Детерминированный случай // Программирование. 2003. № 5. С. 59–69.
4. Бурдонов И.Б., Грошев С.Г., Демаков А.В., Камкин А.С., Косачев А.С., Сортвов А.А. Параллельное тестирование больших автоматных моделей // Вестник ННГУ. 2011. № 3(2). С. 187–193.
5. Бурдонов И.Б. Обход неизвестного ориентированного графа конечным роботом // Программирование. 2004. № 4. С. 11–34.
6. Бурдонов И.Б. Проблема отката по дереву при обходе неизвестного ориентированного графа конечным роботом // Программирование. 2004. № 6. С. 6–29.
7. Бурдонов И.Б., Косачев А.С. Обход неизвестного графа коллективом автоматов // Труды Международной конференции «Научный сервис в сети Интернет: все грани параллелизма». М.: Изд-во МГУ, 2013. С. 228–232.
8. Бурдонов И.Б., Косачев А.С. Полное тестирование с открытым состоянием ограниченно недетерминированных систем // Программирование. 2009. № 6. С. 3–18.
9. Бурдонов И.Б., Косачев А.С., Кулямин В.В. Избыточные алгоритмы обхода ориентированных графов. Недетерминированный случай // Программирование. 2004. № 1. С. 2–17.

Бурдонов Игорь Борисович, д-р физ.-мат. наук. E-mail: igor@ispras.ru
Косачев Александр Сергеевич, канд. физ.-мат. наук. E-mail: kos@ispras.ru
Института системного программирования РАН (г. Москва)

Поступила в редакцию 25 января 2014 г.

Burdonov Igor B., Kossatchev Alexander S. (Institute for System Programming of the RAS, Moscow, Russian Federation).

Graph learning by interacting automata.

Keywords: graph learning; interacting automata; parallel processing.

The paper describes the learning an unknown oriented graph by automata traversing the arcs of the graph and exchanging messages. The learning completes when each arc has been traversed by at least one automaton.

Section 1 discusses the problems of identification of arcs and nodes in the graph, ways of exchanging messages between automata and interaction of the automata with the graph. The arcs outgoing from the same node are considered enumerated. When traversing the graph, each automaton indicates the number of the arc outgoing from the current node. The nodes have unique identifiers. The exceptions are sinks and nodes with only one incoming arc: these nodes can have empty identifier. Automaton can recognize the identifier of the current node and the number of outgoing arcs. The automata interaction media provides message transfer from one automaton to another in a limited time. The following primitives are recognized: "send message" indicating the receiver address and "receive message from any sender".

Section 2 describes the idea of the algorithm. Automata are divided into *regulators* and *runners*. Runners traverse the arcs of the graph. When in a new (yet untraversed) node, runner becomes its regulator. The node regulator controls the runners travel along the arcs outgoing from this node. The regulator of the initial node, in addition, generates new runners. During execution, the graph description is created in the cumulative memory of the regulators as a spanning tree of the already traversed part of the graph oriented from the initial node.

To navigate the runners, the following message is used: "*where to*" from runner to regulator and response message "*traverse the arc*" indicating the arc number and the address of the regulator of the arc terminal node if this arc has already been traversed. Regulator sends runners either along the arcs of the tree or along untraversed arcs. After traversing an untraversed arc, runners learn the identifier of its terminal node, but it needs to know whether this node has been already traversed. To get this information, the "*poll*" message is sent iterating all regulators linked into a list of addresses. If the node has been traversed, the node regulator sends the response message to the runner indicating its address, and the runner resends this address to the regulator of the initial node of the traversed chord in the "*finish*" message. This message is also used to indicate the traversal end: the regulator of the tree arc terminal node sends "*finish*" to the regulator of the arc initial node if it has received "*finish*" along all outgoing arcs. The traversal is over when regulator of the initial node receives "*finish*" along all outgoing arcs.

Section 3 gives the estimate of the traversal time $\Theta(m+nD)$, where n is a number of nodes, m is a number of arcs, and D is the maximal number of nodes with nonempty identifiers on the graph path from the initial node.

The conclusion describes possible directions of further work: learning of nondeterministic graphs, use of external memory, learning graphs of special kind.

REFERENCES

1. Bourdonov I.B., Kossatchev A.S., Petrenko A.K., Kuliain V.V. The UniTesK approach to designing test suites. *Programming and Computer Software*, 2003, vol. 29, no. 6, pp. 310-322. DOI: 10.1023/B:PACS.0000004131.54590.fb.
2. Demakov A., Kamkin A., Sortov A. *High-performance testing: Parallelizing functional tests for computer systems using distributed graph exploration*. In Open Cirrus Summit. Moscow, 2011.
3. Bourdonov I.B., Kossatchev A.S., Kuliain V.V. Irredundant algorithms for traversing directed graphs: The deterministic case. *Programming and Computer Software*, 2003, vol. 29, no. 5, pp. 245-258. DOI: 10.1023/A:1025733107700.
4. Bourdonov I.B., Groshev S.G., Demakov A.V., Kamkin A.S., Kossatchev A.S., Sortov A.A. Parallel testing of large automata models. *Vestnik NNGU – Vestnik UNN*, 2011, no. 3(2), pp. 187-193. (In Russian).
5. Bourdonov I.B. Obkhod neizvestnogo orientirovannogo grafa konechnym robotom [Backtracking problem in the traversal of an unknown directed graph by a finite robot]. *Programmirovaniye*, 2004, vol. 30, no. 6, pp. 305-322.
6. Bourdonov I.B. Problema otkata po derevu pri obkhode neizvestnogo orientirovannogo grafa konechnym robotom [Traversal of an unknown directed graph by a finite robot]. *Programmirovaniye*, 2004, vol. 30, no. 4, pp. 188-203.
7. Bourdonov I.B., Kossatchev A.S. [Unknown graph traversing by automata group]. *Trudy Mezhdunarodnoy superkomp'yuternoy konferentsii "Nauchnyy servis v seti Internet: vse grani parallelizma"* [The proceeding of Russian Supercomputer conference "Scientific service of Internet"]. Moscow: MSU Publ., 2013, pp. 228-232. (In Russian).
8. Bourdonov I.B., Kossatchev A.S. Polnoe testirovaniye s otkrytym sostoyaniem ogranichenno nedeterminirovannykh sistem [Complete open state testing of limitedly nondeterministic systems]. *Programmirovaniye*, 2009, vol. 35, no. 6, pp. 301-313.
9. Bourdonov I.B., Kossatchev A.S., Kuliain V.V. Neizbytochnye algoritmy obkhoda orientirovannykh grafov. Nedeterminirovanny sluchay [Irredundant algorithms for traversing directed graphs: The Nondeterministic case]. *Programmirovaniye*, 2004, vol. 30, no. 1, pp. 2-17.

ПРОЕКТИРОВАНИЕ И ДИАГНОСТИКА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

УДК 681.518.5:004.052.32

В.В. Сапожников, Вл.В. Сапожников, Д.В. Ефанов

ОБ ИСПОЛЬЗОВАНИИ СВОЙСТВ КОДОВ С СУММИРОВАНИЕМ ПО ОБНАРУЖЕНИЮ МОНОТОННЫХ ОШИБОК В СИСТЕМАХ ФУНКЦИОНАЛЬНОГО КОНТРОЛЯ КОМБИНАЦИОННЫХ СХЕМ

Исследуются свойства классических, модифицированных и модульно модифицированных кодов с суммированием единичных информационных разрядов по обнаружению ошибок в системах функционального контроля комбинационных логических схем. Особое внимание уделяется свойствам кодов с суммированием по обнаружению монотонных (однонаправленных) ошибок в информационных разрядах кодовых слов. Показано, что модифицированные и модульно модифицированные коды с суммированием не обнаруживают только такие монотонные искажения в информационных векторах кодовых слов, которые имеют кратность $d = M, 2M, \dots, iM$ ($iM \leq m$, m – длина информационного вектора). Установлено, что при заданном значении величины M доля монотонных необнаруживаемых ошибок кратности d от общего числа необнаруживаемых ошибок той же кратности не зависит от числа m и одинакова для всех модульно модифицированных кодов.

Ключевые слова: система функционального контроля; комбинационная схема; схема с монотонно независимыми выходами; код Бергера; модифицированный код Бергера; модульно модифицированные коды с суммированием; информационные разряды; необнаруживаемая ошибка; монотонная необнаруживаемая ошибка.

В системе функционального контроля (рис. 1) заданное логическое устройство $f(x)$, вычисляющее систему рабочих булевых функций $f_1(x), f_2(x), \dots, f_m(x)$, дополняется блоком контрольной логики $g(x)$, вычисляющим ряд контрольных функций $g_1(x), g_2(x), \dots, g_k(x)$. Между сигналами, формируемыми на выходах блоков $f(x)$ и $g(x)$, устанавливается однозначное соответствие, контролируемое самопроверяемым тестером [1–3]. В случае нарушения данного соответствия при возникновении дефектов в структурах блоков $f(x)$ и $g(x)$ или при наличии отказов в блоке тестера (это могут быть как сбои и устойчивые отказы, так и неисправности типа «временная задержка» передачи сигналов [4, 5]) на контрольных выходах устанавливается непарафазный сигнал $\langle 00 \rangle$ или $\langle 11 \rangle$.

Основной задачей организации функционального контроля является обеспечение проверки правильности работы блока $f(x)$ в режиме реального времени (без его отключения от объекта управления). При этом необходимо на выходе тестера получать информацию о наличии неисправности в любой из составляющих системы функционального контроля [6].

Часто при организации систем функционального контроля используются коды с суммированием единичных информационных разрядов [7–11]. Данные коды принадлежат к классу систематических, т.е. их кодовые слова разделимы и состоят из информационных и контрольных векторов. Значения разрядов контрольного вектора вычисляются по значениям разрядов информационного вектора по определенным правилам. В системе функционального контроля значения выходных сигналов блока $f(x)$ ставятся в соответствие значениям информационных разрядов кода с суммированием, а значениям выходных сигналов блока $g(x)$ – значениям контрольных разрядов.

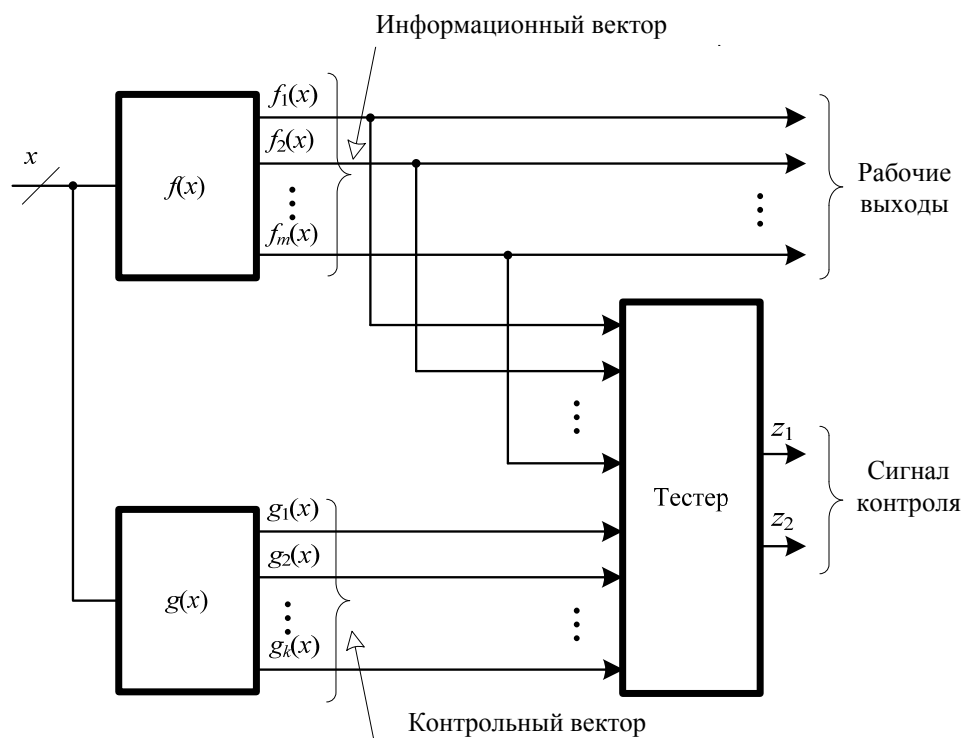


Рис. 1. Структура системы функционального контроля

В структуре, изображенной на рис. 1, блоки $f(x)$, $g(x)$ и тестер реализуются отдельно, поэтому в произвольный момент времени допускается возникновение только одного дефекта (одновременное появление неисправностей в нескольких блоках маловероятно). Если в структуре блока $f(x)$ один логический элемент связан с несколькими его выходами, то при возникновении одиночной неисправности данного элемента возникает одновременное искажение значений нескольких выходов, что соответствует искажению нескольких информационных разрядов кодового слова. Поэтому можно рассматривать свойства кодов с суммированием по обнаружению искажений в информационных векторах при условии отсутствия искажений в контрольных векторах, определяя тем самым свойства системы функционального контроля по обнаружению ошибок в блоке $f(x)$.

Данная работа посвящена исследованию свойств кодов с суммированием по обнаружению монотонных (однонаправленных) искажений в информационных векторах кодовых слов. На практике результаты исследования позволяют определить условия применимости различных кодов с суммированием при организации контроля логических устройств и синтезе надежных дискретных систем.

1. Коды Бергера в системах функционального контроля

Классический код с суммированием, или код Бергера [12], образуется путем приписывания к информационному вектору контрольного вектора, представляющего собой двоичное число $\langle g_1 g_2 \dots g_k \rangle$, равное сумме числа единичных информационных разрядов (весу r информационного вектора). Число контрольных разрядов в коде Бергера зависит от числа информационных разрядов и равно $k = \lceil \log_2(m+1) \rceil$ (запись $\lceil x \rceil$ обозначает целое сверху от x). Обозначим классический код с суммированием как $S(m,k)$ -код, где m – длина информационного вектора кода, а k – длина контрольного вектора.

$S(m,k)$ -код обнаруживает любые ошибки в информационных разрядах, которые изменяют значение веса информационного вектора (рис. 2, а). К необнаруживаемым ошибкам относятся разнонаправленные ошибки в информационных векторах, содержащие группу искажений $\{0 \rightarrow 1 \text{ и } 1 \rightarrow 0\}$ (рис. 2, б).

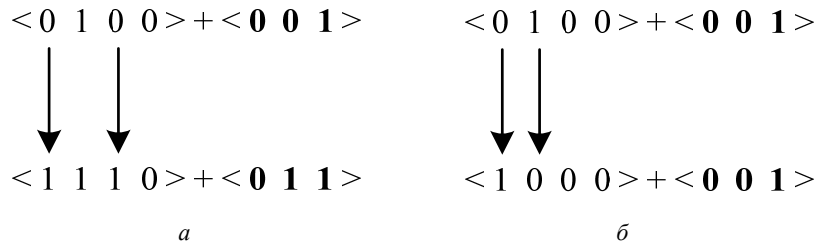


Рис. 2. Примеры ошибок в информационных векторах $S(m,k)$ -кодов:
 a – обнаруживаемые; b – необнаруживаемые

Свойства $S(m,k)$ -кодов по обнаружению ошибок в информационных векторах удобно определять, задавая код в табличной форме, где все информационные векторы распределены на контрольные группы по весу r [13]. Например, в табл. 1 дается $S(4,3)$ -код.

Таблица 1

$S(4,3)$ -код

r							
0	1	2	3	4	5	6	7
Контрольный вектор							
000	001	010	011	100	101	110	111
0000	0001	0011	0111	1111			
	0010	0101	1011				
	0100	0110	1101				
	1000	1001	1110				
		1010					
		1100					

Необнаруживаемая $S(m,k)$ -кодом ошибка соответствует переходу между информационными векторами, принадлежащими одной контрольной группе. Для $S(4,3)$ -кода, приведенного в табл. 1, существуют по 12 необнаруживаемых ошибок в контрольных группах с $r = 1$ и $r = 3$ и 30 необнаруживаемых ошибок в контрольной группе с $r = 2$ (всего 48 двукратных и 6 четырехкратных необнаруживаемых искажений).

Коды Бергера имеют довольно большое число необнаруживаемых ошибок в информационных векторах (обозначим данную величину как N_m), что объясняется неравномерностью распределения информационных векторов на контрольные группы [14, 15]. В каждой контрольной группе содержится по C_m^r информационных векторов. С увеличением значения r до величины $r = \left\lfloor \frac{m}{2} \right\rfloor$ (запись $\lfloor x \rfloor$

обозначает целое снизу от x) число информационных векторов в контрольной группе возрастает, а с дальнейшим приближением к величине m аналогичным образом уменьшается. Минимальным количеством необнаруживаемых ошибок обладает такой (m,k) -код, у которого все 2^m информационных векторов равномерно распределены по 2^k контрольным группам [15, 16]. Этот код является оптимальным (m,k) -кодом по критерию минимума числа необнаруживаемых искажений в информационных векторах. Число необнаруживаемых ошибок в нем равно

$$N_{m,k}^{\min} = 2^m (2^{m-k} - 1). \quad (1)$$

Оптимальный $(4,3)$ -код имеет всего 16 необнаруживаемых ошибок в информационных векторах, что более чем в три раза меньше, чем у $S(4,3)$ -кода.

«Близость» кода с суммированием к оптимальному (m,k) -коду определяется коэффициентом эффективности ξ :

$$\xi = \frac{N_{m,k}^{\min}}{N_m}. \quad (2)$$

Чем ближе значение ξ к 1, тем эффективнее код с позиции обнаружения искажений в информационных векторах при заданных значениях длин информационных и контрольных векторов. $S(4,3)$ -код, например, имеет значение $\xi = 0,2963$. У других кодов Бергера значение ξ , как правило, не превышает величины 0,5 [15, 16].

Таким образом, коды Бергера далеки от оптимальных (m,k) -кодов. Более того, $S(m,k)$ -коды имеют низкую эффективность в области обнаружения ошибок малой кратности. В [13] показано, что в независимости от длины информационного вектора доля необнаруживаемых ошибок четной кратности d в информационных векторах $S(m,k)$ -кодов от общего числа искажений в информационных векторах данной кратности является постоянной величиной:

$$\beta_d = 2^{-d} C_d^{\frac{d}{2}}. \quad (3)$$

Из (3) следует, что $S(m,k)$ -кодом не обнаруживается 50% двукратных и 37,5% четырехкратных искажений в информационных векторах.

Достоинством классических кодов с суммированием является то, что они обнаруживают 100% монотонных ошибок в информационных векторах. При таких ошибках в информационном векторе искажаются только нули либо только единицы, что нарушает значение веса r информационного вектора (см. рис. 2, а).

Коды Бергера являются оптимальными кодами по обнаружению монотонных искажений в информационных векторах [17, 18], что используется при организации систем функционального контроля. Решение задачи 100%-го обнаружения одиночных неисправностей в блоке $f(x)$ достигается несколькими способами. Первый такой способ заключается в дублировании исходной комбинационной схемы $f(x)$ и сравнении между собой одноименных функций с применением блоков сжатия парафазных сигналов *TRC* (*two-rail checker*) [11, 19, 20]. Второй способ основан на выделении на множестве выходов блока $f(x)$ групп независимых и монотонно независимых выходов (Н- и МН-групп) [18, 21 – 23]. Множество выходов блока $f(x)$ образует Н-группу, если при возникновении любой неисправности в его внутренней структуре искажается значение только одного выхода из группы. К МН-группе относятся такие выходы блока $f(x)$, для которых любое одиночное искажение в его внутренней структуре приводит к возникновению ошибки только на одном выходе, либо монотонной ошибки на нескольких выходах. Третий способ состоит в преобразовании блока $f(x)$ путем внутреннего резервирования логических элементов в контролепригодную структуру, в которой все выходы образуют одну Н-группу [18] или МН-группу [21–23]. Контроль таких устройств осуществляется на основе использования $S(m,k)$ -кода.

2. Модифицированные коды Бергера в системах функционального контроля

Для повышения эффективности обнаружения искажений в информационных векторах $S(m,k)$ -кодов в [15, 16] разработан алгоритм преобразования контрольного вектора в слове кода Бергера в контрольный вектор нового (модифицированного) кода (обозначим его как $RS(m,k)$ -код).

Алгоритм. Правила построения модифицированного кода Бергера.

1. Фиксируется модуль $M = 2^{\lceil \log_2(m+1) \rceil - 1}$.
2. Подсчитывается вес информационного вектора r .
3. Число r представляется по модулю M (другими словами, определяется вычет числа r по заданному модулю): $V = (r) \bmod M$.
4. Определяется поправочный коэффициент α , равный сумме по модулю два произвольного (но заранее установленного) числа любых информационных разрядов.

5. Формируется число $W = V + \alpha M$.

6. Полученное число W представляется в двоичном виде и записывается в контрольный вектор.

Применение алгоритма иллюстрируется на примере $RS(4,3)$ -кода, для которого поправочный коэффициент рассчитывается по формуле $\alpha = x_2 \oplus x_3$ (табл. 2).

Таблица 2

Векторы $RS(4,3)$ -кода

№	Информационные разряды				r	$V = r(mod 4)$	α	$W = V + \alpha M$	Контрольные разряды		
	x_1	x_2	x_3	x_4					y_1	y_2	y_3
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	1	0	1	0	0	1
2	0	0	1	0	1	1	1	5	1	0	1
3	0	0	1	1	2	2	1	6	1	1	0
4	0	1	0	0	1	1	1	5	1	0	1
5	0	1	0	1	2	2	1	6	1	1	0
6	0	1	1	0	2	2	0	2	0	1	0
7	0	1	1	1	3	3	0	3	0	1	1
8	1	0	0	0	1	1	0	1	0	0	1
9	1	0	0	1	2	2	0	2	0	1	0
10	1	0	1	0	2	2	1	6	1	1	0
11	1	0	1	1	3	3	1	7	1	1	1
12	1	1	0	0	2	2	1	6	1	1	0
13	1	1	0	1	3	3	1	7	1	1	1
14	1	1	1	0	3	3	0	3	0	1	1
15	1	1	1	1	4	0	0	0	0	0	0

При модификации кода Бергера по приведенному выше алгоритму происходит перераспределение информационных векторов по контрольным группам (рис. 3).

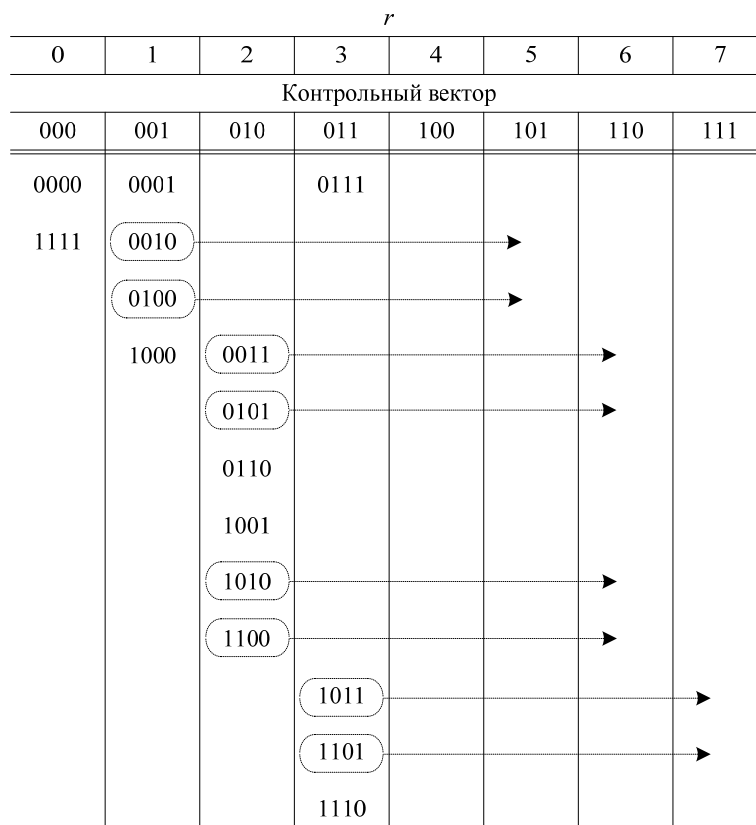


Рис. 3. Перераспределение информационных векторов в $RS(4,3)$ -коде с $\alpha = x_2 \oplus x_3$

Информационные векторы $RS(4,3)$ -кода более равномерно распределены на контрольные группы, чем информационные векторы $S(4,3)$ -кода (табл. 3). $RS(4,3)$ -код имеет 24 необнаруживаемые ошибки, что в 2,25 раза меньше, чем у $S(4,3)$ -кода. Однако $RS(4,3)$ -код не является оптимальным и имеет коэффициент эффективности $\xi = 0,6667$. В среднем модифицированные коды Бергера обнаруживают примерно в два раза больше искажений в информационных векторах, чем классические коды с суммированием, и более близки к оптимальным (m,k) -кодам [16].

Таблица 3

RS(4,3)-код

R							
0	1	2	3	4	5	6	7
Контрольный вектор							
000	001	010	011	100	101	110	111
0000	0001	0110	0111		0010	0011	1011
1111	1000	1001	1110		0100	0101	1101
						1010	
						1100	

В зависимости от формулы вычисления поправочного коэффициента α для данного значения m можно построить $\sum_{j=1}^{m-1} C_m^j = 2^m - 2$ модифицированных кодов с суммированием [16]. От этой же формулы зависят и свойства $RS(m,k)$ -кода по обнаружению искажений в информационных векторах. При этом существенным является только количество информационных разрядов в линейной сумме α , но не конкретное положение разрядов в информационном векторе (обозначим число различных информационных разрядов в формуле поправочного коэффициента α как t). Поэтому число модифицированных кодов с суммированием с различными характеристиками по обнаружению искажений в информационных векторах ограничивается не числом сочетаний информационных разрядов в α , а величиной, равной длине информационного вектора. Более того, как показывают исследования, $RS(m,k)$ -коды при заданном значении m обладают симметричными распределениями чисел необнаруживаемых ошибок по всем кратностям относительно величины $t = \frac{m}{2}$. К примеру, на рис. 4 представлены зависимости между числом необнаруживаемых ошибок по кратностям d для различных $RS(8,4)$ -кодов при различных значениях t .

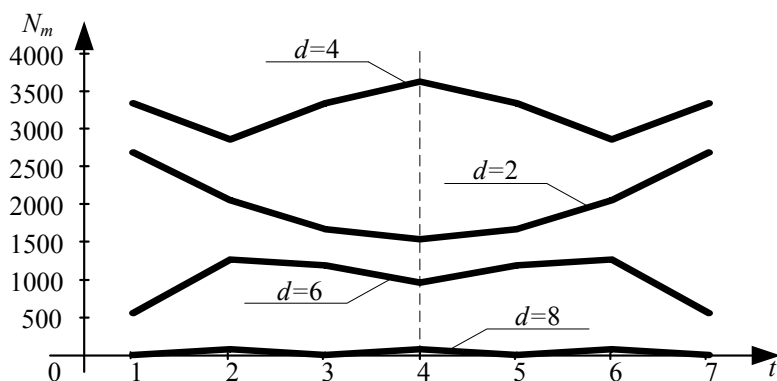


Рис. 4. Распределение необнаруживаемых ошибок по кратностям в $RS(8,4)$ -кодах

Для определения свойств $RS(m,k)$ -кодов по обнаружению ошибок в информационных векторах была составлена компьютерная программа, реализующая алгоритм распределения информационных

векторов на контрольные группы и подсчитывающая число необнаруживаемых ошибок по кратностям d . С ее помощью были установлены следующие особенности $RS(m,k)$ -кодов:

1. Распределения необнаруживаемых ошибок по кратностям d в $RS(m,k)$ -кодах являются одинаковыми при одинаковом значении t .

2. Всего существует $\left\lfloor \frac{m}{2} \right\rfloor$ разновидностей $RS(m,k)$ -кодов, каждый из которых имеет отличное от других распределение необнаруживаемых ошибок по кратностям d .

3. $RS(m,k)$ -коды с числом информационных разрядов в формуле поправочного коэффициента α , равным t и $m-t$, имеют одно и то же распределение необнаруживаемых ошибок (рис. 4).

4. Для $RS(m,k)$ -кодов с $M \geq 8$ минимальным общим количеством необнаруживаемых ошибок обладает код, в котором для определения α использовалась сумма $\left\lfloor \frac{m}{2} \right\rfloor$ информационных разрядов.

5. Минимум необнаруживаемых искажений кратности $d = 2$ имеет модифицированный код, для определения α в котором использовалась сумма $\left\lfloor \frac{m}{2} \right\rfloor$ информационных разрядов.

6. При малых m в $RS(m,k)$ -кодах доминируют необнаруживаемые ошибки малых кратностей, с увеличением m последовательно приоритет по количеству необнаруживаемых ошибок переходит к ошибкам больших кратностей.

Однако модифицированные коды с суммированием имеют недостаток в сравнении с классическими кодами Бергера. При получении значений разрядов контрольного вектора в соответствии с п. 1 алгоритма фиксируется модуль M , по которому в дальнейшем представляется вес r информационного вектора. Это приводит к тому, что внутри одной контрольной группы могут оказаться информационные векторы с различным истинным весом r , но одинаковым по модулю M . Например, в табл. 3 в контрольной группе $\langle 000 \rangle$ $RS(4,3)$ -кода присутствуют два информационных вектора – $\langle 0000 \rangle$ и $\langle 1111 \rangle$. Взаимные переходы данных векторов друг в друга происходят при возникновении монотонной ошибки кратности $d = 4$.

$RS(4,3)$ -код имеет две необнаруживаемые монотонные ошибки в информационных векторах. В кодах с большими значениями m количество необнаруживаемых монотонных искажений намного больше. Данная особенность модифицированных кодов Бергера должна учитываться при синтезе систем функционального контроля комбинационных логических схем с выходами, образующими МН-группу.

Установим свойства модифицированных кодов с суммированием по обнаружению монотонных ошибок в информационных векторах.

Утверждение 1. $RS(m,k)$ -код не обнаруживает только те монотонные ошибки информационных векторов, которые имеют кратность $d = M = 2^{\lceil \log_2(m+1) \rceil - 1}$.

Справедливость утверждения 1 вытекает из следующих рассуждений. В соответствии с п. 6 алгоритма в одну и ту же контрольную группу $RS(m,k)$ -кода входят информационные векторы, для которых рассчитанное значение числа W является одним и тем же. В соответствии с п. 5 (так как $\alpha = 0$ или 1) минимальное значение числа $W_{\min} = V$, а максимальное значение – $W_{\max} = V + M$. Таким образом, в соответствии с п. 3 и 5 в одну контрольную группу кода $RS(m,k)$ входят информационные векторы с весами r , кратными установленному модулю M . Взаимные необнаруживаемые переходы между такими векторами связаны с монотонными ошибками кратности M .

Утверждение 2. Для $RS(m,k)$ -кодов число монотонных необнаруживаемых искажений зависит только от числа информационных разрядов в формуле поправочного коэффициента, но не зависит от того, какие именно это разряды.

Покажем справедливость утверждения 2. Мы рассматриваем $RS(m,k)$ -коды на всех информационных векторах – их количество равно 2^m . Число V будет поставлено в соответствие всем информационным векторам. Число векторов с весом $W \neq V$ определяется поправочным коэффициентом α , но

этот коэффициент рассчитывается на всем множестве информационных векторов, а значит (учитывая свойства функции сложения по модулю два принимать равное количество нулевых и единичных значений на всех наборах), увеличивает значение V до величины W на 2^t векторах. Количество векторов со значением суммарного веса W определяется только числом информационных разрядов в сумме α .

В табл. 4 представлены распределения необнаруживаемых ошибок по кратностям d в $RS(10,4)$ -кодах с различным числом t информационных разрядов в сумме поправочного коэффициента α . В левой части таблицы даются распределения необнаруживаемых ошибок по кратностям d : для каждого значения t число в верхней клетке определяет количество монотонных необнаруживаемых ошибок, а в нижней клетке – общее количество необнаруживаемых ошибок данной кратности; в правой части таблицы приведены величины v_d , характеризующие доли монотонных необнаруживаемых ошибок кратности d от общего числа необнаруживаемых ошибок той же кратности.

Т а б л и ц а 4

Характеристики $RS(10,4)$ -кодов

t	Число необнаруживаемых ошибок кратности d						v_d					
	2	4	6	8	10	2–10	2	4	6	8	10	2–10
1	0	0	0	72	0	72	0	0	0	0,0277778	–	0,0007478
	18 432	48 384	26 880	2 592	0	96 288						
2	0	0	0	232	0	232	0	0	0	0,0277778	0	0,0025091
	14 848	37 632	31 360	8 352	272	92 464						
3	0	0	0	168	0	168	0	0	0	0,0277778	–	0,0018299
	12 288	37 632	35 840	6 048	0	91 808						
4	0	0	0	168	0	168	0	0	0	0,0277778	0	0,0018321
	10 752	40 704	33 920	6 048	272	91 696						
5	0	0	0	200	0	200	0	0	0	0,0277778	–	0,0021815
	10 240	42 240	32 000	7 200	0	91 680						

Анализ таблиц, аналогичных табл. 4, для $RS(m,k)$ -кодов с различными длинами информационных векторов позволил установить следующие свойства данного класса кодов по обнаружению монотонных ошибок в информационных векторах:

1. Минимум монотонных необнаруживаемых ошибок для данного m имеет $RS(m,k)$ -код со значением $t = 1$.

2. Значение доли монотонных необнаруживаемых ошибок кратности d от общего числа необнаруживаемых ошибок той же кратности при $d = M$ для любого $RS(m,k)$ -кода является постоянной величиной и не зависит от значения t .

3. У $RS(m,k)$ -кодов, для которых m есть степень числа 2, а t является четным числом, существует всего 2 монотонные необнаруживаемые ошибки максимальной кратности $d = m$; если t является нечетным числом, такой код обнаруживает все монотонные ошибки максимальной кратности $d = m$. Любые другие $RS(m,k)$ -коды обнаруживают все монотонные ошибки максимальной кратности.

Свойства $RS(m,k)$ -кодов по обнаружению монотонных ошибок в информационных векторах можно использовать на практике при построении систем функционального контроля схем с МН-выходами.

Обозначим через b максимальное четное число выходов схемы $f(x)$, с которыми связан выход внутреннего логического элемента при помощи путей с одинаковым по четности числом инверсий. Например, на рис. 5 приведена произвольная схема с шестью выходами. На схеме пунктирными линиями показаны пути от выхода элемента 2 к выходами схемы. Анализ путей схемы показывает, что с выходами $f_1(x)$, $f_2(x)$ и $f_6(x)$ элемент 2 связан путями с четным количеством инверсий, а с выходами $f_3(x)$, $f_4(x)$ и $f_5(x)$ – путями с нечетным количеством инверсий. Поэтому для данной схемы $b = 2$.

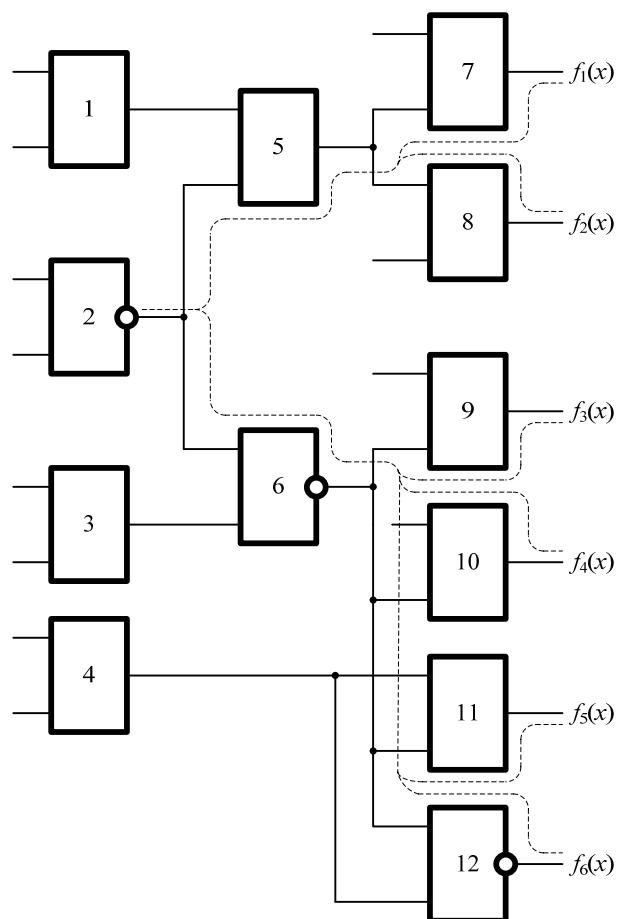


Рис. 5. Произвольная комбинационная схема

Справедливо следующее положение.

Утверждение 3. В схеме с МН-выходами обнаруживаются все одиночные ошибки на основе применения в системе функционального контроля $RS(m,k)$ -кода, если характеристика схемы

$$b < 2^{\lceil \log_2(m+1) \rceil - 1}. \quad (4)$$

В самом деле, так как в схеме с МН-выходами возможны только монотонные искажения сигналов на выходах, то для 100%-го обнаружения всех искажений с применением модифицированного кода Бергера необходимо исключить возникновение ошибок кратности $d = M$ (см. утверждение 1). Значение $M = 2^{\lceil \log_2(m+1) \rceil - 1}$. Поэтому если структура схемы $f(x)$ такова, что в ней не существует путей с одинаковым по четности числом инверсий от произвольного логического элемента к M выходам, то данная схема может контролироваться $RS(m,k)$ -кодом. Величина b в формуле (4) как раз определяет число выходов схемы, имеющих подобные пути.

Организовать контроль схемы, приведенной на рис. 5, можно с применением $RS(6,3)$ -кода, так как значение b для данной схемы равно двум.

3. Модульно модифицированные коды с суммированием

В [16] рассмотрен класс модульно модифицированных кодов с суммированием. Контрольные векторы кодовых слов модульно модифицированных кодов получают по алгоритму, приведенному в п. 2, но с тем условием, что значение M выбирается из множества $M \in \{2, 4, \dots, 2^{\lceil \log_2(m+1) \rceil - 2}\}$. Модульно модифицированные коды, или $RSM(m,k)$ -коды, по сравнению с $S(m,k)$ и $RS(m,k)$ -кодами, имеют важ-

ное преимущество – они имеют меньшее число контрольных разрядов. Это позволяет строить более простые с точки зрения технической реализации системы функционального контроля, так как уменьшается сложность блоков контрольной логики и тестера (см. рис. 1).

$RSM(m,k)$ -коды являются обобщением модифицированных кодов Бергера, поэтому обладают всеми их особенностями. Кроме того, модульно модифицированным кодам присущи следующие свойства [16]:

1. В $RS2(m,k)$ - и $RS4(m,k)$ -кодах в независимости от длины информационного вектора и способа определения коэффициента α общее число необнаруживаемых ошибок имеет одно и то же значение.
2. При одинаковых значениях t $RS4(m,k)$ -код не обнаруживает ровно вдвое меньше ошибок информационных разрядов, чем $RS2(m,k)$ -код с тем же значением m .
3. $RSM(m,k)$ -коды с модулями $M \geq 4$ имеют одинаковое число двукратных необнаруживаемых ошибок информационных разрядов при одинаковом t для заданной длины информационного слова m .
4. Коды с модулем $M = 2$ имеют вдвое больше необнаруживаемых ошибок информационных разрядов кратности $d = 2$ при одинаковых значениях t , чем $RSM(m,k)$ -коды с модулями $M \geq 4$.

В табл. 5 и 6 для примера приводятся характеристики всех возможных модульно модифицированных кодов с суммированием со значением $m = 10$.

Т а б л и ц а 5

Характеристики $RS2(10,2)$ -кодов

t	Число необнаруживаемых ошибок кратности d						v_d					
	2	4	6	8	10	2–10	2	4	6	8	10	2–10
1	18 432	16 128	2 688	72	0	37 320	0,5	0,125	0,03125	0,0078125	–	0,1429228
	36 864	129 024	86 016	9 216	0	261 120						
2	14 848	12 544	3 136	232	2	30 762	0,5	0,125	0,03125	0,0078125	0,0019531	0,1178079
	29 696	100 352	100 352	29 696	1024	261 120						
3	12 288	12 544	3 584	168	0	28 584	0,5	0,125	0,03125	0,0078125	–	0,1094669
	24 576	100 352	114 688	21 504	0	261 120						
4	10 752	13 568	3 392	168	2	27 882	0,5	0,125	0,03125	0,0078125	0,0019531	0,1067785
	21 504	108 544	108 544	21 504	1024	261 120						
5	10 240	14 080	3 200	200	0	27 720	0,5	0,125	0,03125	0,0078125	–	0,1061581
	20 480	112 640	102 400	25 600	0	261 120						

Т а б л и ц а 6

Характеристики $RS4(10,3)$ -кодов

t	Число необнаруживаемых ошибок кратности d						v_d					
	2	4	6	8	10	2–10	2	4	6	8	10	2–10
1	0	16 128	0	72	0	16 200	0	0,25	0	0,015625	–	0,1240809
	18 432	64 512	43 008	4 608	0	130 560						
2	0	12 544	0	232	0	12 776	0	0,25	0	0,015625	0	0,0978554
	14 848	50 176	50 176	14 848	512	130 560						
3	0	12 544	0	168	0	12 712	0	0,25	0	0,015625	–	0,0973652
	12 288	50 176	57 344	10 752	0	130 560						
4	0	13 568	0	168	0	13 736	0	0,25	0	0,015625	0	0,1052083
	10 752	54 272	54 272	10 752	512	130 560						
5	0	14 080	0	200	0	14 280	0	0,25	0	0,015625	–	0,109375
	10 240	56 320	51 200	12 800	0	130 560						

Утверждение 4. $RSM(m,k)$ -код не обнаруживает только те монотонные ошибки информационных векторов, которые имеют кратность $d=M, 2M, \dots, iM (iM \leq m)$.

Утверждение 4 обобщает положение утверждения 1 на случай $RSM(m,k)$ -кода. В соответствии с п. 3–6 алгоритма в одну контрольную группу $RSM(m,k)$ -кода размещаются информационные векторы с одним и тем же значением веса, подсчитанным по модулю M . Поэтому в контрольных группах возможны только такие ошибочные переходы между информационными векторами, которые связаны с монотонными искажениями разрядов кратностей $M, 2M, \dots, iM (iM \leq m)$.

Сравнивая табл. 4 с табл. 5 и 6, отметим, что модульно модифицированные коды сохраняют важное свойство модифицированных кодов с суммированием. При установленном модуле M для $RSM(m,k)$ -кодов значения величин v_d (долей монотонных необнаруживаемых ошибок кратности d от общего числа необнаруживаемых ошибок той же кратности) при любом m являются постоянными величинами и не зависят от значения t .

Из утверждения 4 вытекают условия применения $RSM(m,k)$ -кодов при организации систем функционального контроля устройств с МН-выходами.

Утверждение 5. Контроль схемы с МН-выходами может быть организован на основе применения $RSM(m,k)$ -кода в том случае, если характеристика схемы

$$b < 2^M. \quad (5)$$

Утверждение 5 доказывается аналогично утверждению 3.

Из утверждений 3 и 5 следует, что выбор $RS(m,k)$ -кода (или $RSM(m,k)$ -кода) определяется только параметром b схемы и не зависит от числа ее выходов.

Заключение

В данной статье рассмотрены свойства кодов с суммированием единичных информационных разрядов по обнаружению монотонных ошибок в информационных векторах. Установлено, что возникновение монотонных необнаруживаемых ошибок в модифицированных и модульно модифицированных кодах с суммированием связано с модулем M , по которому вычисляется вес информационного вектора. Результаты данной работы позволяют на практике устанавливать критерии применения кодов с суммированием при организации контроля комбинационных схем, любые одиночные неисправности элементов в которых могут приводить только к монотонным искажениям на выходах [24].

ЛИТЕРАТУРА

1. Nelson V.P. Fault-Tolerant Computing: Fundamental Concepts // Journal Computer. 1990. V. 23. Issue 7. P. 19–25.
2. Toubia N.A., McCluskey E.J. Logic Synthesis of Multilevel Circuits with Concurrent Error Detection // IEEE Trans. Computer-Aided Design of Integrated Circuits and System. 1997. V. 16. P. 783–789.
3. Nicolaidis M., Zorian Y. On-Line Testing for VLSI – A Compendium of Approaches // Journal of Electronic Testing: Theory and Applications. 1998. No.12. P. 7–20.
4. Matrosova A., Lipsky V., Melnikov A., Singh V. Path Delay Faults and ENF // Proceedings of 8th IEEE East-West Design & Test Symposium (EWDTS'2010), St. Petersburg, Russia, September 17–20. 2010. P. 164–167.
5. Матросова А.Ю., Останин С.А., Сингх В. Обнаружение несущественных путей логических схем на основе совместного анализа И-ИЛИ деревьев и SSBD-графов // Автоматика и телемеханика. 2013. № 7. С. 126–142.
6. Пархоменко П.П., Согомонян Е.С. Основы технической диагностики (оптимизация алгоритмов диагностирования, аппаратные средства). М. : Энергоатомиздат, 1981. 320 с.
7. Lo J.-C. An SFS Berger Check Prediction ALU and Its Application to Self-Checking Processor Designs // Computer-Aided Design of Integrated Circuits and Systems. 1992. V. 11. Issue 4. P. 525–540.
8. Gorshe S.S. A Self-Checking ALU Design with Efficient Codes // Proceedings of 14th VLSI Test Symposium. Princeton, NJ, USA, 1996. P. 157–161.
9. Das D., Toubia N.A. Synthesis of Circuits with Low-Cost Concurrent Error Detection Based on Bose-Lin Codes // Journal of Electronic Testing: Theory and Applications. 1999. V. 15. Issue 1–2. P. 145–155.
10. Piestrak S.J. Design of Self-Testing Checkers for Unidirectional Error Detecting Codes. Wrocław : Oficyna Wydawnicza Politechniki Wrocławskiej, 1995. 111 p.
11. Fujiwara E. Code Design for Dependable Systems: Theory and Practical Applications. John Wiley & Sons, 2006. 720 p.
12. Berger J.M. A Note on Error Detecting Codes for Asymmetric Channels // Information and Control. 1961. V. 4. Issue 1. P. 68–73.

13. Ефанов Д.В., Сапожников В.В., Сапожников Вл.В. О свойствах кода с суммированием в схемах функционального контроля // Автоматика и телемеханика. 2010. № 6. С. 155–162.
14. Сапожников В.В., Сапожников Вл.В., Ефанов Д.В. Предельные свойства кода с суммированием // Известия Петербургского университета путей сообщения. 2010. № 3. С. 290–299.
15. Блюдов А.А., Ефанов Д.В., Сапожников В.В., Сапожников Вл.В. Построение модифицированного кода Бергера с минимальным числом необнаруживаемых ошибок информационных разрядов // Электронное моделирование. 2012. Т. 34, № 6. С. 17–29.
16. Efanov D., Sapozhnikov V., Sapozhnikov Vl., Blyudov A. On the Problem of Selection of Code with Summation for Combinational Circuit Test Organization // Proceedings of 11th IEEE East-West Design & Test Symposium (EWDTS'2013). Rostov-on-Don, Russia, September 27–30, 2013. P. 261–266.
17. Freiman C.V. Optimal Error Detection Codes for Completely Asymmetric Binary Channels // Information and Control. 1962. V. 5. Issue 1. P. 64–71.
18. Согомонян Е.С., Слабаков Е.В. Самопроверяемые устройства и отказоустойчивые системы. М. : Радио и связь, 1989. 208 с.
19. Сапожников В.В., Сапожников Вл.В. Самопроверяемые дискретные устройства. СПб. : Энергоатомиздат, 1992. 224 с.
20. Lala P.K. Self-Checking and Fault-Tolerant Digital Design. University of Arkansas, 2001. 216 p.
21. Saposhnikov V.V., Saposhnikov Vl.V., Morosov A., Goessel M. Design of Self-Checking Unidirectional Combinational Circuits with Low Area Overhead // Proceedings of 2nd IEEE International On-Line Testing Workshop (IOLTW). Biarritz, France, 1996. P. 56–67.
22. Morosow A., Saposhnikov V.V., Saposhnikov Vl.V., Goessel M. Self-Checking Combinational Circuits with Unidirectionally Independent Outputs // VLSI Design. 1998. V. 5. Issue 4. P. 333–345.
23. Saposhnikov V.V., Morosov A., Saposhnikov Vl.V., Gõessel M. A New Design Method for Self-Checking Unidirectional Combinational Circuits // Journal of Electronic Testing: Theory and Applications. 1998. V. 12. Issue 1–2. P. 41–53.
24. Сапожников В.В., Сапожников Вл.В., Гёссель М., Морозов А.А. Метод построения комбинационных самопроверяемых устройств с обнаружением всех одиночных неисправностей // Электронное моделирование. 1998. Т. 20, № 6. С. 70–80.

Сапожников Валерий Владимирович, д-р техн. наук, профессор. E-mail: kat@pgups.edu

Сапожников Владимир Владимирович, д-р техн. наук, профессор. E-mail: kat@pgups.edu

Ефанов Дмитрий Викторович. E-mail: TrES-4b@yandex.ru

Петербургский государственный университет путей сообщения (г. Санкт-Петербург)

Поступила в редакцию 2 апреля 2014 г.

Sapozhnikov Valery V., Sapozhnikov Vladimir V., Efanov Dmitry V. (Petersburg State Transport University, Petersburg, Russian Federation).

On the use of the properties of sum code for unidirectional error detection in concurrent error detection (CED) systems of combinational circuits

Keywords: concurrent error detection (CED); combinational circuit; circuits with unidirectionally-independent outputs; Berger code; modified Berger code; unit-modified sum codes; data bits; undetectable error; unidirectional undetectable error.

For the CED systems organization of logical units, codes with summation of one data bits are frequent in use. Sum code data vector is assigned to operating outputs of logical unit $f(x)$, and check vector – to the outputs of reference logic block $g(x)$. A unique correspondence is set between the signals, formed at the outputs of $f(x)$ и $g(x)$ blocks, and this correspondence is checked by self-checking checkers. The primal problem of on-line testing organization is to provide the operating accuracy control of $f(x)$ block on a real time basis (without its disconnection from controlled object). It is also necessary to get the information about faults existence in any of CED system components at the checker outputs.

When all blocks of CED system are presented independently at any given time, the occurrence of one defect only (simultaneous occurrence of faults in several blocs is unlikely) is considered acceptable. If one logical element connects within $f(x)$ block structure with the number of its outputs, the simultaneous distortion of several output values occurs while a single fault occurs of this component, that corresponds the distortion of the number of codeword data bits. Hence, it is possible to consider sum codes properties for distortion detection in data vectors under condition of nonexistence of distortions in check vectors, that defines CED system properties for error detection in $f(x)$ block.

This article includes the analysis of classic, modified and unit-modified one data bits sum codes properties for error detection in the CED systems of combinational logic circuits. Special focus is put on sum codes capability for detection of unidirectional (one-directional) errors in codeword data bits. It has been known that classic sum code (Berger code) detects any unidirectional distortions in data vectors, that allows to make the best use of this code for combinational circuits checking, any single faults of components in it may result only to unidirectional distortions at the outputs. Modified and unit-modified sum codes can also be implemented for these purposes. This study shows that modified and unit-modified sum codes do not detect only such unidirectional distortions in codeword data vectors that has multiplicity $d = M, 2M, \dots, iM$ ($iM \leq m$, where m – data vector length). It has been established that for given M value the fraction of unidirectional undetectable errors of d multiplicity from the total number of undetectable error of the same multiplicity does not depend on m and is the same for all unit-modified codes. The results of this study allow to set the practical criteria of application of modified and unit-modified sum codes by organization of concurrent error detection of combinational circuits.

REFERENCES

1. Nelson V.P. Fault-tolerant computing: Fundamental concepts. *J. Computer*, 1990, vol. 23, issue 7, pp. 19-25. DOI: 10.1109/2.56849.
2. Touba N.A., McCluskey E.J. Logic synthesis of multilevel circuits with concurrent error detection. *IEEE Trans. Computer-Aided Design of Integrated Circuits and System*, 1997, vol. 16, pp. 783-789. DOI: 10.1109/43.644041.
3. Nicolaidis M., Zorian Y. On-line testing for VLSI – A compendium of approaches. *J. of Electronic Testing: Theory and Applications*, 1998, no.12, pp. 7-20. DOI: 10.1023/A:1008244815697.
4. Matrosova A., Lipsky V., Melnikov A., Singh V. Path Delay Faults and ENF. *Proceedings of 8th IEEE East-West Design & Test Symposium (EWDTS'2010)*. St. Petersburg, Russia, September 17-20, 2010, pp. 164-167.
5. Matrosova A.Yu., Ostanin S.A., Singh V. Detection of false paths in logical circuits by joint analysis of the AND/OR trees and SSBDD-Graphs. *Automation and Remote Control*, 2013, vol. 74, issue 7, pp. 126-142. DOI: 10.1134/S0005117913070084.
6. Parkhomenko P.P., Sogomonyan E.S. *Osnovy tekhnicheskoy diagnostiki (optimizatsiya algoritmov diagnostirovaniya, apparatnyye sredstva)* [The foundation of technical diagnosis (optimization of diagnosis algorithms, hardware). Moscow: Energoatomizdat Publ., 1981. 320 p.
7. Lo J.-C. An SFS Berger check prediction ALU and its application to self-checking processor designs. *Computer-Aided Design of Integrated Circuits and Systems*, 1992, vol. 11, issue 4, pp. 525-540. DOI: 10.1109/43.125100.
8. Gorshe S.S. A self-checking ALU design with efficient codes. *Proceedings of 14th VLSI Test Symposium*. Princeton, NJ, USA, 1996, pp. 157-161.
9. Das D., Touba N.A. Synthesis of circuits with low-cost concurrent error detection based on Bose-Lin codes. *J. of Electronic Testing: Theory and Applications*, 1999, vol. 15, issue 1-2, pp. 145-155. DOI: 10.1023/A:1008344603814.
10. Piestrak S.J. *Design of self-testing checkers for unidirectional error detecting codes*. Wrocław: Oficyna Wydawnicza Politechniki Wrocławskiej, 1995. 111 p.
11. Fujiwara E. *Code design for dependable systems: Theory and practical applications*. John Wiley & Sons, 2006. 720 p.
12. Berger J.M. A note on error detecting codes for asymmetric channels. *Information and Control*. 1961, vol. 4, issue 1, pp. 68-73. DOI: 10.1016/S0019-9958(61)80037-5.
13. Efanov D.V., Sapozhnikov V.V., Sapozhnikov V.I.V. On summation code properties in functional control circuits. *Automation and Remote Control*, 2010, vol. 71, issue 6, pp. 155-162. DOI: 10.1134/S0005117910060123.
14. Sapozhnikov V.V., Sapozhnikov V.I.V., Efanov D.V. Predel'nye svoystva koda s summirovaniem [Berger Code limit properties]. *Izvestiya Peterburgskogo universiteta putey soobshcheniya*, 2010, issue 3, pp. 290-299.
15. Blyudov A.A., Efanov D.V., Sapozhnikov V.V., Sapozhnikov V.I.V. Formation of the Berger Modified Code with minimum number of undetectable errors of informational bits. *Elektronnoe modelirovanie*, 2012, vol. 34, issue 6, pp. 17-29. (In Russian).
16. Efanov D., Sapozhnikov V., Sapozhnikov V.I., Blyudov A. [On the problem of selection of code with summation for combinational circuit test organization]. *Proceedings of 11th IEEE East-West Design & Test Symposium (EWDTS'2013)*. Rostov-on-Don, 2013, pp. 261-266. (In Russian).
17. Freiman C.V. Optimal error detection codes for completely asymmetric binary channels. *Information and Control*, 1962, vol. 5, issue 1, pp. 64-71. DOI: 10.1016/S0019-9958(62)90223-1.
18. Sogomonyan E.S., Slabakov E.V. *Samoproveryaemye ustroystva i otkazoustoychivye sistemy* [Self-checking devices and fault-tolerant systems]. Moscow: Radio & Communication Publ., 1989. 208 p.
19. Sapozhnikov V.V., Sapozhnikov V.I.V. *Samoproveryaemye diskretnye ustroystva* [Self-checking digital devices]. St. Petersburg: Energoatomizdat Publ., 1992. 224 p.
20. Lala P.K. *Self-checking and fault-tolerant digital design*. University of Arkansas, 2001. 216 p.
21. Saposhnikov V.V., Saposhnikov V.I.V., Morosov A., Goessel M. Design of self-checking unidirectional combinational circuits with low area overhead. *Proceedings of 2nd IEEE International On-Line Testing Workshop (IOLTW)*. Biarritz, France, 1996, pp. 56-67.
22. Morosov A., Saposhnikov V.V., Saposhnikov V.I.V., Goessel M. Self-checking combinational circuits with unidirectionally independent outputs. *VLSI Design*, 1998, vol. 5, issue 4, pp. 333-345. DOI: 10.1155/1998/20389.
23. Saposhnikov V.V., Morosov A., Saposhnikov V.I.V., Göessel M. A new design method for self-checking unidirectional combinational circuits. *J. of Electronic Testing: Theory and Applications*, 1998, vol. 12, issue 1-2, pp. 41-53. DOI: 10.1023/A:1008257118423.
24. Sapozhnikov V.V., Sapozhnikov V.I.V., Göessel M., Morosov A.A. Metod postroeniya kombinatsionnykh samoproveryaemykh ustroystv s obnaruzheniem vseh odinochnykh neispravnostey [Method of construction of combinational self-checking devices with detection of all single faults]. *Elektronnoe modelirovanie*, 1998, vol. 20, issue 6, pp. 70-80.

УДК 519.714

Л.Д. Черемисинова

ЛОГИЧЕСКИЙ СИНТЕЗ КОМБИНАЦИОННЫХ КМОП СХЕМ С УЧЕТОМ РАССЕИВАНИЯ МОЩНОСТИ

Рассматривается задача синтеза многоуровневых логических сетей в базе библиотечных элементов КМОП СБИС, оптимизированных по критерию площади и среднему значению рассеиваемой КМОП микросхемой мощности при ее реализации на кристалле СБИС. Предлагаются сравнительные прогнозные оценки вариантов оптимизации схем по критерию рассеивания мощности на этапах логического синтеза: минимизации двухуровневых и многоуровневых И-ИЛИ-схем, отображения И-ИЛИ-схем в технологический базис КМОП СБИС.

Ключевые слова: логический синтез; заказные КМОП СБИС; рассеивание мощности.

Прогресс в области микроэлектронных технологий обуславливает постоянное повышение степени интеграции и тактовой частоты, что позволяет, в свою очередь, создавать на одном кристалле все более быстродействующие и функционально сложные устройства. Однако наряду с огромными возможностями, которые открывают для электронных изделий эти успехи, они обуславливают также и серьезные проблемы, связанные с рассеянием мощности. Высокая степень интеграции привела к появлению устройств более чувствительных к рассеиванию мощности, к обострению проблемы надежности по сравнению с аналогичными устройствами меньшей степени интеграции. Эти проблемы не могут быть проигнорированы, так как проектируются все более сложные изделия для различных сфер применения, которые будут работать длительное время без подзарядки батареи питания и иметь наряду с этим низкую стоимость.

Проектирование с учетом энергопотребления до сих пор остается по большому счету искусством, что связано, в частности, и с отсутствием эффективных средств оценки эффекта используемых в процессе проектирования эвристик на энергопотребление схемы, реализуемой в дальнейшем на кристалле СБИС. Решением проблемы энергопотребления при проектировании СБИС занимаются многие специалисты и фирмы, например Cadence Design Systems (лидер в области автоматизации проектирования), Apache Design, Atrenta, Magma Design Automation, Synopsys, Mentor Graphics и др.

Снижение энергопотребления данной схемы может обеспечиваться на разных уровнях ее проектирования. При этом чем более ранним является этап, тем важнее получать на нем более качественные решения. В частности, на логическом уровне (за счет построения удачной логической структуры) можно достичь сокращения рассеивания мощности на 10–20% без ущерба для быстродействия и сложности схемы [2]. САПР микроэлектронных схем должны иметь средства, позволяющие оценивать и минимизировать энергопотребление схем уже в процессе их логического проектирования, чтобы избежать дорогостоящей процедуры их перепроектирования на стадии схемотехнического проектирования.

В настоящей работе рассматривается проблема снижения рассеивания мощности (энергопотребления) при логическом синтезе многоуровневых схем из библиотечных элементов, выполненных на основе технологии статических КМОП схем, которая в настоящее время является доминирующей технологией в области цифровых СБИС. Компоненты СБИС, выполненные по этой технологии, потребляют подавляющую часть необходимой для их функционирования энергии во время их переключения, соответственно, энергопотребление существенно зависит от переключательной активности узлов схемы, которая определяется, в свою очередь, динамикой функционирования и структурой схемы.

Рассматриваемый подход к синтезу позволяет минимизировать площадь и среднее значение мощности, рассеиваемой микросхемой, реализуемой на кристалле КМОП СБИС. Предлагаются оценки среднего значения рассеиваемой мощности многоуровневых комбинационных КМОП схем на всех этапах логического синтеза, когда 1) основные схемотехнические решения, такие как частота синхронизации и напряжение питания, фиксированы; 2) предполагается синхронная реализация схем; 3) для оценки энергопотребления в процессе синтеза схем используется статический метод, основанный на вероятностных характеристиках входных сигналов. Предполагается, что возможные входные воздействия на схему существенно определяются ее применением и могут быть спрогнозированы проектировщиком на вероятностном уровне, при этом, помимо функционального описания проектируемой схемы, задаются вероятностные оценки активности изменения сигналов на ее входах.

1. Оценка вариантов оптимизации схемы по прогнозируемому рассеиванию мощности

В общем случае мощность рассеивания энергии логической схемой является сложной функцией, зависящей от задержек распространения сигналов через схему, частоты синхронизации, технологических параметров изготовления, топологии микросхемы, а в случае КМОП технологии мощность рассеивания существенно зависит от последовательности прилагаемых к схеме входных воздействий. В типичных КМОП цепях от 60 до 80% всей рассеиваемой мощности приходится на ее динамическую составляющую [4], порождаемую нестационарным поведением узлов схемы. Согласно упрощенной модели энергия рассеивается КМОП микросхемой всякий раз, когда изменяется сигнал на ее выходе. Отсюда следует, что более активные в переключательном плане КМОП схемы рассеивают больший объем энергии. Таким образом, рассеивание мощности существенно зависит от переключательной активности элементов схемы, а она, в свою очередь, определяется последовательностью подаваемых входных воздействий на КМОП схему, т.е. динамикой функционирования.

На логическом уровне, когда схемы еще нет и часто неизвестен даже технологический базис, в котором она будет реализована, рассеивание мощности может быть снижено путем такого преобразования схемы, которое обеспечивает уменьшение ее переключательной активности без изменения функциональности [2, 5]. Для оценки предпочтительности вариантов оптимизации схемы на логическом уровне может быть использовано количественное изменение переключательной активности результирующей схемы при выборе этих вариантов. Такой подход к оценке рассеивания мощности дает возможность сравнивать варианты реализации схемы в процессе ее проектирования, что позволяет уже на логическом уровне проектировать схемы, потенциально имеющие низкое рассеивание мощности.

В основе методов оценки переключательной активности лежит подход, основанный на вероятностных характеристиках входных сигналов и функционально-структурных свойствах исследуемой схемы [6]. Подход предполагает задание вероятностей переключения сигналов на входных полюсах схемы, которые отражают частоту смены значений сигналов и используются для вычисления вероятностей переключения сигналов на выходах узлов схемы. В литературе предлагается множество вероятностных методов оценки энергопотребления логических схем [1–3, 5–7]. Для оценки вариантов оптимизации схемы достаточным представляется использование простых, быстро вычисляемых оценок изменения переключательной активности, в основе которых лежат следующие предположения: 1) изменения на входах схемы распространяются через все ее элементы мгновенно, а значит, все переходы в схеме происходят одновременно; 2) для каждого входного полюса узла имеет место временная независимость, предполагающая, что значение сигнала в любом такте синхронизации не зависит от его значений в предшествующих тактах; 3) входные полюсы узла пространственно независимы, что означает отсутствие корреляции значений сигналов на них (что может быть вызвано, например, наличием разветвлений на выходах элементов или обратных связей).

Различают вероятность появления сигнала 1 (или 0) на некотором полюсе схемы и вероятность смены значения сигнала на этом полюсе. Вероятность появления сигнала 1 на i -м полюсе схемы называется сигнальной вероятностью p_i и определяется средней долей тактов, на которых сигнал на i -м полюсе

имеет единичное значение. Вторая вероятность $p_i^{1 \rightarrow 0}$ (или $p_i^{0 \rightarrow 1}$) есть вероятность смены значения сигнала с 1 на 0 (или с 0 на 1) и определяется средней долей тактов, на которых сигнал на i -м полюсе меняет свое значение по сравнению со значением в предшествующем такте. При сделанных предположениях вероятность $p_i^{1 \rightarrow 0}$ ($p_i^{0 \rightarrow 1}$) равна произведению вероятности появления на нем сигнала 1 (0) в одном такте на вероятность того, что в следующем такте на нем появится 0 (1). Соответственно, переключательная активность полюса z_i схемы равна $E(z_i) = p_i^{1 \rightarrow 0} + p_i^{0 \rightarrow 1}$ или (в предположении, что $0 < p_i < 1$):

$$E(z_i) = 2p_i(1 - p_i). \quad (1)$$

Например, если сигнальная вероятность p_e полюса e равна 0,2, то переключательная активность этого полюса $E(e) = 0,32$; или если $p_e = 0,5$, то $E(e) = 0,5$.

Вероятность p_e появления сигнала 1 на выходе элемента e существенно зависит от вероятностных характеристик p_i сигналов на его входах и от функции, реализуемой этим элементом. Если сигналы на входах элемента не коррелируют в пространстве и времени, то сигнальные вероятности на выходе простых элементов типа инвертора и И, ИЛИ с $n(e)$ входными полюсами подсчитываются исходя из таблиц истинности реализуемых ими функций:

$$p_e^- = 1 - p_i; \quad p_e^{\wedge} = \prod_{i=1}^{n(e)} p_i; \quad p_e^{\vee} = 1 - \prod_{i=1}^{n(e)} (1 - p_i). \quad (2)$$

Если заданы сигнальные вероятности входных сигналов схемы, то они могут быть распространены на выходы элементов схемы и через всю схему на ее выходные полюсы. Таким образом, могут быть подсчитаны переключательные активности всех полюсов схемы и, соответственно, переключательная активность схемы в целом как сумма переключательных активностей всех ее полюсов. Следует заметить, что если даже требование пространственной (и временной) независимости выполняется для входных сигналов схемы, оно может не иметь места для входных сигналов внутренних элементов схемы (как результат наличия разветвлений на выходах элементов и линий обратной связи). В этом случае вычисленные по формулам (1), (2) вероятности имеют погрешность. Однако для сравнительной оценки вариантов оптимизации достаточно, как правило, ограничиться этими простыми оценками, не прибегая к более точным, но гораздо более трудоемко вычисляемым оценкам.

2. Логический синтез в базе библиотечных элементов с учетом рассеивания мощности

В процессе логического синтеза абстрактное описание поведения синтезируемой схемы (система булевых функций) представляется в базе элементов технологической библиотеки КМОП СБИС. Каждый из элементов характеризуется своей функцией и физическими характеристиками. Подход основан, как и большинство известных методов синтеза, на разбиении процесса логического синтеза на стадии технологически независимой оптимизации и технологического отображения. Первая стадия синтеза ориентирована на оптимизацию и декомпозицию логики, а вторая – на реализацию полученного функционального описания в заданном технологическом базисе. Цель первого этапа заключается в минимизации сложности многоуровневой схемы в технологически независимом базисе элементов. Последний состоит, как правило, из простых вентилях, выбор которых может быть не привязан ни к какому элементному базису или может выбираться исходя из реальной технологической библиотеки, как это делается в [8] и в предлагаемой работе. Сложность многоуровневой схемы измеряется числом вентилях, глубиной схемы и оценкой рассеивания мощности на логическом уровне.

Второй этап заключается в переводе многоуровневой схемы из вентилях в технологический базис на основе структурного покрытия соответствующей объектной сети подсхемами, реализующими библиотечные элементы. Такой подход не предполагает кардинальную перестройку схемы, полученной на этапе технологически независимой оптимизации, отсюда следует, что качество искомого покрытия существенно зависит от ее структуры. Просчеты, допущенные при ее синтезе, не могут быть компенсированы в полной мере на этапе технологического отображения, поэтому в существующих САПР большое внимание уделяется этапу технологически независимой оптимизации.

Технологически независимая оптимизация включает в себя в качестве первого этапа минимизацию функций реализуемых логических описаний в классе дизъюнктивных нормальных форм (ДНФ). Принимая во внимание специфику КМОП базиса, рационально проводить совместную минимизацию с учетом полярности функций, выбирая ту ее форму (ДНФ или ее инверсию), которая имеет меньшую сложность и энергопотребление. На втором этапе минимизированная система ДНФ, представляемая двухуровневой схемой, декомпозируется в многоуровневую объектную сеть из вентилях И, ИЛИ с ограниченным числом входов, на которые распадаются структуры основных элементов КМОП библиотеки, в которую планируется отобразить схему.

Сложность схемы и рассеивание ею мощности тесно связаны в том смысле, что уменьшение площади схемы имеет тенденцию обеспечивать также и снижение ее энергопотребления, а увеличение площади, как правило, ведет к увеличению энергопотребления [7]. Из этих соображений следует, что в процессе синтеза необходим компромисс между критериями минимизации энергопотребления и площади. Основная проблема оптимизации схемы заключается в том, что на этапе логического синтеза, еще не привязанного к конкретному технологическому базису, трудно оценить энергопотребление реальной схемы на достаточно достоверном уровне. В связи с этим минимизация динамической составляющей энергопотребления схемы (оцениваемой переключательной активностью схемы) на этапе логического синтеза в ущерб сложности схемы может в итоге привести к увеличению ее энергопотребления. Это происходит из-за возрастания значений других составляющих энергопотребления.

В настоящей работе решение проблемы оптимизации логических схем исходит из того, что основной путь энергосбережения при проектировании логических схем, прежде всего, подразумевает сокращение площади, занимаемой схемой на кристалле. На всех этапах логического синтеза при оценке вариантов оптимизации используется ранжированный критерий: в первую очередь количественная оценка изменения площади, затем количественная оценка изменения переключательной активности схемы.

3. Оптимизация двухуровневых логических схем

Для оптимизации двухуровневых схем с учетом рассеяния мощности используются методы минимизации функциональных описаний систем полностью и частично определенных булевых функций, которые представляют собой модификации известных методов минимизации булевых функций в классе ДНФ путем добавления в них эвристик, направляющих процесс минимизации к получению систем ДНФ, реализуемых схемами с наименьшим рассеиванием мощности. Практически все методы минимизации двухуровневых представлений булевых функций основаны на разделении множества искомым простым импликант на три подмножества: существенных, несущественных и условно существенных. Первые должны быть включены в любое безызбыточное решение, вторые не должны включаться ни в одно решение, а из третьих выбирается некоторое безызбыточное подмножество, покрывающее все интервалы единичных областей задания минимизируемых функций, которые оказались не покрытыми существенными импликантами.

Методы минимизации различаются способами построения простых импликант из числа условно существенных и критериями, которым они должны удовлетворять для включения их в решение. Различают методы: 1) последовательного построения простых импликант, включаемых в решение (например, метод конкурирующих интервалов [9]), путем укрупнения интервала булева пространства аргументов, представляющего импликанту, за счет включения в него непокрытых элементарных конъюнкций; 2) последовательного расширения интервала булева пространства аргументов, представляющего сначала конъюнкцию исходной ДНФ, а в перспективе простую импликанту, покрывающую эту конъюнкцию за счет исключения некоторых литералов, входящих в нее (например, методы, реализуемые ESPRESSO [10]). Простейший метод минимизации может использовать лишь одну операцию расширения условно существенных импликант. Наиболее просто с целью учета энергосбережения модифицируются методы минимизации, в которых в качестве кандидатов в искомое реше-

ние строится сразу несколько простых импликант, или методы, в которых найденное решение модифицируется методом последовательных улучшений.

Для того чтобы направить минимизацию к получению энергосберегающего решения, необходимо в процессе получения простых импликант и безызбыточных покрытий вычислять и учитывать переключаемые активности всех простых импликант, используя формулы (1), (2). Для пояснения процедур, направляющих процесс минимизации на получение нужного результата, рассмотрим операции, входящие практически во все методы минимизации.

Операция расширения интервала выполняется путем исключения его литералов. При расширении интервала принимаются во внимание две цели: уменьшить сложность этого интервала, т.е. максимально расширить его, и покрыть с его помощью (полностью или частично) как можно больше непокрытых еще интервалов. При минимизации энергопотребления желательно: 1) исключать не любые, а наиболее активные литералы и 2) покрывать не любые, а наиболее энергоемкие интервалы. В силу первого утверждения на предмет исключения проверяются сначала литералы с большей переключаемой активностью. В силу второго утверждения важен порядок расширения интервалов: слишком раннее расширение интервала может препятствовать ситуации, когда другой интервал покрывает рассматриваемый. Чтобы снизить энергопотребление, оценивается энергетический вклад каждого интервала и энергоемкие (с большой переключаемой активностью) интервалы расширяются последними в надежде на то, что некоторые другие интервалы расширятся и покроют их.

Операция нахождения безызбыточного покрытия заключается в приведении текущего покрытия ДНФ к безызбыточному виду. При поиске безызбыточного множества простых импликант выбирается минимальное число наименее активных (с меньшей переключаемой активностью) импликант. При этом из безызбыточных множеств простых импликант, представляющих собой покрытие исходного множества интервалов единичных областей минимизируемых булевых функций, отбираются минимальные по мощности или по суммарному числу литералов всех импликант. Каждое из отобранных множеств оценивается суммой переключаемых активностей (или плотностей переключений [6, 11]) входящих в него импликант, и в качестве решения выбирается безызбыточное множество с минимальной оценкой переключаемой активности.

Проведенное сравнение модифицированных методов минимизации [11] с исходными методами (без учета энергопотребления) показало, что минимизация булевых функций с учетом переключаемой активности сигналов позволяет получить уменьшение энергопотребления схем без увеличения их сложности. При этом вычислительные затраты на минимизацию увеличиваются незначительно.

4. Оптимизация многоуровневых логических схем

На этом этапе рассматривается задача построения многоуровневой логической схемы из вентилях И, ИЛИ, НЕ, реализующей систему булевых функций. Эта задача непосредственно следует за минимизацией булевых функций в классе ДНФ и предшествует синтезу логических схем из библиотечных элементов, выполненных по КМОП технологии. Цель этапа заключается в том, чтобы построить такой вариант представления схемы из вентилях, который мог бы служить хорошей отправной точкой для этапа технологического отображения в базис библиотечных элементов. В качестве количественных оценок эффективности проектирования на этом этапе используются суммарное число входных полюсов вентилях и суммарная переключаемая активность полюсов схемы.

Комбинационные элементы типичной библиотеки КМОП СБИС на логическом уровне можно представить древообразными структурами из вентилях НЕ, И и ИЛИ, реализующими инверсную логику. Например, двухъярусные И-НЕ, ИЛИ-НЕ, трехъярусные 2-2И-2ИЛИ-НЕ. Элементы, реализуемые многоярусными структурами, включают в себя вентили с двумя и тремя входными полюсами. Логическую эффективность элемента библиотеки можно оценивать отношением суммарного числа полюсов его структуры к числу транзисторов. Наиболее эффективными элементами являются сильно структурированные микросхемы из вентилях И и ИЛИ с числом входов 2 и 3, наименее эффективными – инвертор и двухвходовые вентили. Соответственно, наиболее привлекательными с точки зрения

покрытия библиотечными элементами являются сильно структурированные сети из вентилях И и ИЛИ с малым числом входных полюсов. Эти соображения положены в основу принятого метода синтеза многоуровневых схем.

Основным, используемым во всех САПР, методом решения задачи декомпозиции систем ДНФ является алгебраическая декомпозиция [8], в основе которой лежит построение факторизованных форм (или факторизованных ДНФ) путем поиска факторов – общих частей конъюнкций или дизъюнкций ДНФ системы. Факторизованная форма является алгебраической формой задания многоуровневого представления ДНФ. Преобразование исходной минимизированной системы ДНФ в факторизованную форму, которой соответствует многоуровневая реализация из вентилях с ограниченным числом входов, разбивается на два этапа [12].

1. *Совместная нетривиальная факторизация* системы ДНФ: выделяются факторы (конъюнкции или ДНФ), которые имеют длину (число литералов) не более максимальных чисел входов $n_{\max}$ и $m_{\max}$ вентилях И и ИЛИ и входят в не менее чем n_{dl} выражений. Ключевым вопросом при поиске факторов является вопрос оценки их стоимостного и энергосберегающего качества. Стоимостное качество T_s фактора s , входящего в выражения из его порождающего множества U_s , упрощенно оценивается площадью соответствующего ему минора булевой матрицы, задающей факторизуемое множество выражений

$$T_s = c(s) (|U_s| - 1),$$

где $c(s)$ – цена реализации выражения s по Квайну. Энергосберегающее качество фактора количественно оценивается тем выигрышем в переключательной активности искомой схемы, который дает выделение этого фактора. В факторизованном множестве выражений переключательные активности всех выражений не изменятся по сравнению с их значениями в исходном множестве, но изменяется нагрузка полюсов схемы, соответствующих литералам, входящим в фактор s : она уменьшится на $(|U_s| - 1)$, соответственно изменяется и суммарная переключательная активность полюсов схемы, реализующих факторизованное множество выражений. Энергосберегающее качество фактора $s = \{z_1, z_2, \dots, z_l\}$ оценивается как

$$P_s = (|U_s| - 1) \sum_{z_i \in s} E(z_i).$$

2. *Построение скобочных выражений ДНФ каждой из функций* системы основано на итеративном вынесении общих литералов конъюнкций заданной ДНФ D за скобки, т.е. на декомпозиции $D = k(A) + B$, где D , A и B – ДНФ (дизъюнкции некоторого множества конъюнкций), а k – конъюнкция, состоящая из некоторого множества литералов, общих для всех конъюнкций из A . Конъюнкция k выбирается следующим образом: ее ядром является некоторый «лучший» литерал x , вместе с этим литералом за скобки выносятся и другие литералы, общие для конъюнкций из ДНФ A . Лучшим считается тот литерал x , который входит в максимальное число l конъюнкций из D , а из равноценных по этому критерию – тот, что имеет максимальное значение переключательной активности. Такой выбор литерала обосновывается тем, что 1) энергонагрузка на полюс схемы, имеющий переключательную активность E_x и соответствующий литералу x , выносимому из l выражений, уменьшается на величину $(l - 1)E_x$; 2) наиболее активный сигнал будет подаваться на схему ближе к выходу, что позволит уменьшить суммарную переключательную активность схемы.

После окончания итеративного процесса вынесения общих литералов конъюнкций за скобки оставшиеся конъюнкции ранга, большего чем $n_{\max}$, факторизуются отдельно. При этом в фактор включаются в первую очередь литералы, соответствующие полюсам с наименьшей переключательной активностью, так как наиболее активные сигналы желательно подавать на входы элементов схемы как можно ближе к выходу.

Проведенные экспериментальные исследования [12] показали, что предложенный метод синтеза позволяет получить достаточно стабильный выигрыш в оценке рассеивания мощности проектируемой схемы по сравнению с аналогичным методом без учета энергопотребления.

5. Технологическое отображение

На этапе технологического отображения используется структурный метод покрытия (наиболее эффективный в практике проектирования сложных схем) многоуровневой схемы из вентилях И, ИЛИ библиотечными элементами. В процессе покрытия структурные схемы библиотечных элементов сравниваются с фрагментами покрываемой схемы, и в случае полного совпадения заменяют их. Соответственно, каждый библиотечный элемент должен быть представлен разными структурами, реализующими его функцию. При структурном покрытии схемы наряду со структурами КМОП элементов, которые реализуют инверсную логику, в библиотеку структурных описаний предлагается включать также двойственные структуры, получаемые путем переноса инверторов с выходов элементов на входы (и, соответственно, смены вентилях И на ИЛИ, а ИЛИ на И). Например, элемент 2И-2ИЛИ-НЕ (NOA), реализующий функцию $\overline{ab \vee c}$, порождает элемент 2ИЛИ-2И с инверторами на входах (AON), реализующий функцию $(\bar{a} \vee \bar{b}) \cdot \bar{c}$.

Покрываемая многоуровневая многовыходная логическая сеть из вентилях представляется ориентированным ациклическим графом $G = (V, U)$, называемым далее объектным. Вершины графа соответствуют базовым вентилям и входным полюсам схемы. Структурное описание библиотечного элемента представляет собой одновыходную многоуровневую логическую сеть из тех же базовых вентилях, что и покрываемая объектная сеть. Каждый библиотечный элемент представляется древовидным ориентированным графом, называемым далее модельным.

Метод покрытия основан на последовательном выделении подграфов G_k графа G , сравнении каждого из них на предмет покрытия его графами H_i и замене его графом H_i , дающим наибольшее значение выбранных критериев оптимизации. Таким образом, метод обеспечивает локальную оптимизацию и является приближенным. В процессе покрытия граф G сжимается за счет исключения из него покрытых подграфов. Одновременно наращивается (начиная с пустого) результирующий граф E , представляющий логическую сеть из библиотечных элементов, которая функционально эквивалентна исходной объектной сети.

Критерии оптимизации покрытия. В процессе покрытия необходимо получить такой вариант схемы из библиотечных элементов, который обеспечивал бы минимум площади и рассеивания мощности. В предлагаемом методе в качестве первого критерия оптимизации принимается площадь полученной в результате покрытия схемы, которая измеряется суммарным числом транзисторов всех библиотечных элементов и оценивается суммой стоимостей модельных графов, входящих в покрытие объектного графа. Соответственно, в качестве стоимостной оценки варианта покрытия подграфа графа $G = (V, U)$ принимается относительная стоимость покрытия одного полюса объектной сети, которая измеряется отношением числа покрытых полюсов фрагмента сети к стоимости покрывающего библиотечного элемента (которая измеряется числом транзисторов): чем больше это число, тем более желателен вариант покрытия.

Энергопотребление схемы, полученной в результате покрытия, оценивается суммой переключательных активностей всех полюсов схемы. Соответственно, в качестве энергосберегающей оценки варианта покрытия подграфа графа $G = (V, U)$ принимается сумма переключательных активностей вентилях, покрытых библиотечным элементом.

Если пренебречь задержкой сигналов в логических элементах, то при оценке рассеивания мощности можно пренебречь также и интенсивностью переключений сигналов на внутренних полюсах библиотечных элементов. Это допущение обосновывается тем, что наиболее существенный вклад в паразитную емкость схемы вносят соединительные линии и на перезарядку этой емкости в процессе переключений сигналов тратится основная доля расходуемой мощности источника питания. Отсюда схему, реализуемую по КМОП технологии, нужно покрывать библиотечными элементами так, чтобы как можно больше узлов схемы с наибольшей переключательной активностью оказалось внутри библиотечных элементов. Соответственно, за значение критерия энергоэффективности варианта покрытия принимается сумма переключательных активностей элементов, которые соответствуют покрытым вершинам, ставшим внутренними вершинами покрывающего образца.

Повышение быстродействия алгоритма покрытия. Основной перебор при выполнении покрытия сосредоточен в процедуре выбора пары: покрываемый подграф G_k графа G и покрывающий модельный граф H_j библиотечного элемента. Следующие процедуры позволяют существенно сократить этот перебор.

1. Поиск и покрытие подграфов G_k объектного графа, допускающих покрытие единственным образом (единственным модельным графом). В простейшем случае такими подграфами являются вершины с большими степенями захода (соответствующие элементам с большим числом входных полюсов), для которых существует единственный вариант покрытия (например, для многих серий КМОП библиотек это число больше трех).

2. Упорядочение модельных графов. Если упорядочить модельные графы H_j по убыванию их эффективности (измеряемой отношением числа полюсов его структуры к числу транзисторов), то, сравнивая их по порядку с выделяемым на некотором шаге алгоритма покрытия подграфом G_k , можно сократить в ряде случаев перебор: если для некоторого модельного графа H_i значение $H_i \setminus G_k = \emptyset$, то можно не производить сравнение подграфа G_k с другими графами H_i , так как они имеют меньшую эффективность и покрывают, следовательно, меньшую часть графа G . Можно ограничиться лишь поиском варианта покрытия подграфа G_k с наибольшим значением критерия энергоэффективности.

3. Топологическая сортировка объектного и модельных графов. Сортировка модельного графа H_i (он есть дерево) заключается в упорядочении ветвей дерева слева направо по убыванию их сложности. Ветвь P помещается левее ветви R , если P длиннее ветви R или (если они равны по длине) при просмотре вершин от корня в ветви P раньше встретится вершина с большей полустепенью захода, чем в ветви R . Аналогично производится и частичная сортировка графа G . Сортировка позволит быстрее давать ответ на вопрос, возможно ли покрыть фрагмент объектного графа заданным модельным графом.

Заключение

1. Предложены процедуры модификации методов, реализующих этапы синтеза и оптимизации многоуровневых логических сетей в базе библиотечных элементов КМОП СБИС, с целью оптимизировать при логическом синтезе среднее значение рассеиваемой мощности целевой КМОП микросхемой.

2. Предложены сравнительные прогнозные оценки вариантов оптимизации схем по критерию энергопотребления на этапах логического синтеза: минимизации двухуровневых и многоуровневых И-ИЛИ-схем, отображения И-ИЛИ-схем в технологический базис КМОП СБИС.

3. Программные реализации описанных методов, реализующих этапы синтеза многоуровневых логических сетей в базе библиотечных элементов КМОП СБИС, оптимизированных по критерию площади и среднему значению рассеиваемой мощности КМОП микросхемой при ее реализации на кристалле СБИС, включены в программный комплекс автоматизации проектирования интегральных микросхем с пониженным энергопотреблением, выполняемых по КМОП технологии [13].

ЛИТЕРАТУРА

1. Рабаи Ж.М., Чандракасан А., Николит Б. Цифровые интегральные схемы. Методология проектирования. М. : ООО «И.Д. Вильямс», 2007. 912 с.
2. Benini L., De Micheli G. Logic Synthesis for Low Power // Logic Synthesis and Verification / eds. S. Hassoun, T. Sasao, R.K. Brayton. Boston ; Dordrecht ; London : Kluwer Academic Publ., 2002. P. 197–223.
3. Roy K., Prasad S.C. Low Power CMOS VLSI Circuit Design. N.Y. : John Wiley and Sons Inc., 2000. 376 p.
4. Power Compiler. Automatic Power Management within Galaxy™ Implementation Platform. URL: http://pdf.aminer.org/000/285/870/power_compiler_a_gate_level_power_optimization_and_synthesis_system.pdf (дата обращения: 1.02.14).
5. Черемисинова Л.Д. Оценка энергопотребления КМОП схем на логическом уровне // Информационные технологии. 2010. № 8. С. 27–35.
6. Najm F.N. A survey of Power Estimation Techniques in VLSI Circuits // IEEE Trans. on VLSI. 1994. No. 12. P. 446–455.

7. Pedram M. Power Minimization in IC Design: Principles and Applications // ACM Trans. Design Automation Electronic Systems. 1996. V. 1. P. 3–56.
8. Черемисинова Л.Д. Синтез комбинационных КМОП схем с учетом энергосбережения // Информатика. 2010. № 4. С. 112–122.
9. Торопов Н.Р. Минимизация систем булевых функций в классе ДНФ // Логическое проектирование / под ред. А.А. Закревского. Минск: Ин-т техн. кибернетики НАН Беларуси, 1999. С. 4–19.
10. Brayton R.K., Hachtel G.D., McMullen C., Sangiovanni-Vincentelli A.L. Logic minimization algorithms for VLSI synthesis. Boston, Massachusetts : Kluwer Academic Publ., 1984. 193 p.
11. Черемисинов Д.И., Черемисинова Л.Д. Минимизация двухуровневых КМОП схем с учетом энергопотребления // Информационные технологии. 2011. № 5. С. 17–23.
12. Черемисинова Л.Д., Кириенко Н.А. Синтез многоуровневых логических схем с учетом энергопотребления // Информационные технологии. 2013. № 3. С. 8–14.
13. Бибило П.Н., Черемисинова Л.Д., Кардаш С.Н., Кириенко Н.А., Романов В.И., Черемисинов Д.И. Автоматизация логического синтеза КМОП схем с пониженным энергопотреблением // Программная инженерия. 2013. № 8. С. 35–41.

Черемисинова Людмила Дмитриевна, д-р техн. наук. E-mail: cld@newman.bas-net.by
Объединенный институт проблем информатики НАН Беларуси (г. Минск)

Поступила в редакцию 1 февраля 2014 г.

Cheremisinova Ludmila D. (United Institute of Informatics Problems of NAS of Belarus, Minsk, Belarus).

Lower-power logic synthesis of the combinational CMOS circuits.

Keywords: logic design; CMOS VLSI; power dissipation.

In the paper, the task of synthesis of multilevel circuits in the basis of the static CMOS cell library is considered. The circuits are optimized according to area and mean value of the dissipated power of microcircuit implemented on the CMOS VLSI chip. The case of synthesis of the combinational CMOS circuits is discussed when 1) key schematic solutions, such as clock frequency and supply voltage, are assigned; 2) synchronous implementation of the CMOS circuits is supposed; 3) for the purposes of energy estimation during the synthesis process the static method, based on probabilistic properties of input signals, is used. The estimates of mean value of the dissipated power of the multilevel combinational CMOS circuits on all steps of logic synthesis are proposed.

The dynamic power dissipation dominates in the CMOS digital circuits: it accounts for 60%–80% of the overall power dissipation. The dynamic power dissipation is related to charging and discharging node capacitors which is caused by switching signals on circuit nodes. At the stage of logic optimization the majority of the overall energy savings can be achieved by minimizing the switching activities in the circuit. Based on this fact, to estimate variants of circuit optimization on logic level, the quantitative change of switching activity of the resulting (after the variant choosing) circuit is used. It is supposed that a circuit input sequence depends essentially upon its application and can be forecasted by designer on probabilistic level. At that, in addition to functional description of circuit under design, the probabilistic estimates of switching activities of circuit input signals should be specified.

Here, the solving problem of logic circuit optimization is based on the assumption that the basic way of power saving during logic design, first of all, reduces the chip area of the circuit implementation. So, in all techniques at all stages of logic design, a ranged criterion is used: the chip area as the first one, and only then a quantitative criterion of circuit switching activity change.

In the process of logic synthesis, an abstract form of desired circuit behavior (system of Boolean functions) is turned into a design implementation in terms of the logic gates of the CMOS cell library. The considered approach is based, as overwhelming majority of known synthesis methods, on design process partition into two stages: the technology independent phase, where logic minimization and decomposition is performed on the Boolean functions with no regard to physical properties and the technology dependent phase, where the mapping logic network into a physical CMOS cell library is performed.

The first step of technology independent optimization is driven minimization of the Boolean functions in class of disjunctive normal forms (DNFs). In the second step, the two-level network realizing the minimized DNF system is decomposed in the multi-level network of primitive AND and OR gates with restricted numbers of inputs that are predefined by the structures of the basic gates of the target CMOS library. The technology mapping consists in transformation of the multi-level AND-OR network in the target CMOS cell library basis, the mapping is based on structural replacing sub-networks of the initial AND-OR network with cell library instances.

In the paper, the modifications of methods, realizing the basic stages of synthesis and optimization of multi-level logic circuits in CMOS cell library basis, are considered. They are minimization of the two-level and multi-level AND-OR networks and mapping the networks into the technological CMOS VLSI basis. To optimize variants for a circuit under design, the comparative look-ahead estimates are proposed which targeted to reducing the average power dissipation of the CMOS circuit and based on expected change of its switching activity.

REFERENCES

1. Rabaey J.M., Chandrakasan A., Nikolic B. *Tsifrovyye integral'nye skhemy. Metodologiya proektirovaniya* [Digital integrated circuits. A design perspective]. Moscow: OOO I.D. Vilyams Publ., 2003. 761 p.
2. Benini L., De Micheli G. *Logic synthesis for low power*. In: Hassoun S., Sasao T., Brayton R.K. (eds.) *Logic synthesis and verification*. Boston, Dordrecht, London: Kluwer Academic Publ., 2002, pp. 197–223.
3. Roy K., Prasad S.C. *Low power CMOS VLSI circuit design*. New York: John Wiley and Sons Inc., 2000. 376 p.

4. *Power Compiler. Automatic Power Management within Galaxy™ Implementation Platform*. Available: http://pdf.aminer.org/000/285/870/power_compiler_a_gate_level_power_optimization_and_synthesis_system.pdf. (Accessed: 1st February 2014).
5. Cheremisinova L.D. Logic level estimation of power dissipation in CMOS circuits. *Informatsionnye tekhnologii – Information Technologies*, 2010, no. 8, pp. 27-35. (In Russian).
6. Najm F.N. A survey of power estimation techniques in VLSI circuits. *IEEE Transactions on VLSI*, 1994, no. 12, pp. 446-455. DOI: 10.1109/92.335013.
7. Pedram M. Power minimization in IC design: principles and applications. *ACM Transactions Design Automation Electronic Systems*, 1996, vol. 1, pp. 3-56. DOI: 10.1145/225871.225877.
8. Cheremisinova L.D. Sintez kombinatsionnykh KMOP skhem s uchetom energosberezheniya [Low-power synthesis of combinational CMOC networks]. *Informatika*, 2010, no. 4, pp. 112-122.
9. Toropov N.R. *Minimizatsiya sistem bulevykh funktsiy v klasse DNF* [Minimization of a system of Boolean in DNF class]. In: Zakrevskiy A.A. *Logicheskoe proektirovanie* [Logic Design]. Minsk: Institute of Technical Cybernetics of NAS of Belarus Publ., 1999, pp. 4-19.
10. Brayton R.K., Hachtel G.D., McMullen C., Sangiovanni-Vincentelli A.L. *Logic minimization algorithms for VLSI synthesis*. Boston, Massachusetts: Kluwer Academic Publ., 1984. 193 p.
11. Cheremisinov D.I., Cheremisinova L.D. Low power driven minimization of two-level CMOS-circuits. *Informatsionnye tekhnologii – Information Technologies*, 2011, no. 5, pp.17-23. (In Russian).
12. Cheremisinova L.D., Kirienko N.A. Low power driven synthesis of multi-level logical circuits. *Informatsionnye tekhnologii – Information Technologies*, 2013, no. 3, pp. 8-14. (In Russian).
13. Bibilo P.N., Cheremisinova L.D., Kardash S.N., Kirienko N.A., Romanov V.I., Cheremisinov D.I. Low-Power Logical Synthesis of CMOS Circuits Automation. *Programmnaya inzheneriya – Software engineering*, 2013, no. 3, pp. 8-14. (In Russian).

Raimund Ubar**FAULT EFFECT REASONING IN DIGITAL SYSTEMS BY TOPOLOGICAL VIEW
ON LOW- AND HIGH-LEVEL DECISION DIAGRAMS****To memory of my opponent Arkadi Zakrevski**

In order to cope with complexity of diagnostic reasoning of today's digital systems, hierarchical multi-level approaches should be used. In this paper, the possibilities of using Decision Diagrams (DD) for uniform diagnostic modeling of digital systems at different levels of abstraction are discussed. Binary Decision Diagrams (BDD) have become the state-of-the-art data structure in VLSI CAD. A special class of BDDs is presented called Structurally Synthesized BDDs (SSBDD). The idea of SSBDDs is to establish one-to-one mapping between the nodes of SSBDDs and signal paths in gate-level networks, which allows to investigate and solve with SSBDDs test and diagnosis problems directly associated with structural aspects of circuits, like fault modeling, fault collapsing, fault masking, delays, hazards. The main concept of SSBDDs lays on exploiting the topology of graphs for fault reasoning which allows to generalize the methods of test synthesis and fault analysis from the Boolean level to higher register-transfer or behavior levels of hierarchy by introducing a novel High-level DDs (HLDD).

Keywords: Binary Decision Diagrams (BDD); structurally synthesized BDD (SSBDD); shared SSBDD (SSSBDD); high level DD (HLDD); test generation and fault diagnosis.

Within the last two decades BDDs have become the state-of-the-art data structure in VLSI CAD for representation and manipulation of Boolean functions. BDDs were first introduced for logic simulation in 1959 [1], and for logic level diagnostic modeling in [2, 3]. In 1986, Bryant proposed a new data structure called reduced ordered BDDs (ROBDDs) [4]. He showed the simplicity of the graph manipulation and proved the model canonicity that made BDDs one of the most popular representations of Boolean functions [5, 6]. Different types of BDDs have been proposed and investigated during decades such as shared or multi-rooted BDDs [7], ternary decision diagrams (TDD), or in more general, multi-valued decision diagrams (MDD) [8], edge-valued BDDs (EVBDD) [7], functional decision diagrams (FDD) [9], zero-suppressed BDDs (ZBDD) [10], algebraic decision diagrams (ADD) [11], Kronecker FDDs [12], binary moment diagrams (BMD) [13], free BDDs [14], multiterminal BDDs (MTBDD) and hybrid BDDs [15], Fibonacci decision diagrams [16] etc. Overviews about different types of BDDs can be found in [5, 6, 17].

Traditional use of BDDs has been functional, i.e. the target has been to represent and manipulate the Boolean functions by BDDs as efficiently as possible. Less attention has been devoted to representing with BDDs the structural aspects of circuits. Such a goal was first set up in [2, 18] and realized by introducing a possibility for one-to-one mapping between the nodes of BDDs and signal paths in the related circuit. A special class of BDDs was introduced, called initially alternative graphs (AG) [2]. Later AG was renamed as structurally synthesized BDD (SSBDD) [18, 19] in accordance to the way how they were synthesized - directly from the gate-level network structure of logic circuits.

The direct mapping between SSBDDs and circuits allows to model different test related objectives and relations of gate level networks like signal paths, faults in gates or connections, delays on paths, fault masking, fault equivalence and dominance, etc. These issues are difficult to model and simulate explicitly with "classical" BDDs.

Whereas logic level test and diagnosis methods are well developed, this is not the case for higher level test approaches based on abstract execution graphs, system graphs, instruction set architecture (ISA)

descriptions, flowcharts, hardware description languages (HDL, VHDL, Verilog, SystemC), or Petri nets for complex digital systems. Most of these modeling tools are not well suited for reverse simulation and effect-cause reasoning in digital systems. They also need specialized and dedicated for the given language fault processing and reasoning algorithms, which makes it difficult to develop hierarchical approaches for test synthesis, fault analysis and diagnosis. HDL based modeling methods, which are efficient for simulation purposes, do not support analytical reasoning and analysis that is needed in test generation and fault diagnosis.

To overcome this gap, High-Level DDs [HLDD] were introduced as extension of BDDs [18,19]. The topological basis of test algorithms developed for SSBDDs allows to extend these algorithms in a rather straightforward way from logic level to high level. The class of node variables in DDs was extended from Boolean to Boolean vectors and to integer variables, whereas the class of Boolean functions was extended to data manipulation operations typically used in high-level descriptions of digital systems.

The rest of the paper is organized as follows. Section 2 presents the SSBDD as a structural model for gate-level circuits. In Section 3, the main ideas of test generation and fault reasoning based on the topological view on SSBDDs are discussed. Section 4 demonstrates the possibility of using SSBDDs for reasoning multiple faults and for detection of fault masking. An extension of SSBDD model by introducing shared SSBDDs is discussed in Section 5. An overview of generalization of logic level DDs for higher abstraction levels to represent complex digital systems is given in Section 6. Finally, Section 7 discusses the ideas of using HLDDs for automatization of hierarchical test program synthesis for digital systems at RTL and ISA levels, and Section 8 concludes the paper.

1. S SBDD – a model for gate-level circuits

Let us have a gate level combinational circuit with fan-outs only at inputs. Consider the maximum *fan-out free region* (FFR) of the circuit with inputs at the fan-out branches and fan-out free inputs. Let the number of the inputs of FFR be n . For such a *tree-like sub-circuit* we can create by *superposition* of BDDs of gates an SSBDD with n nodes [19].

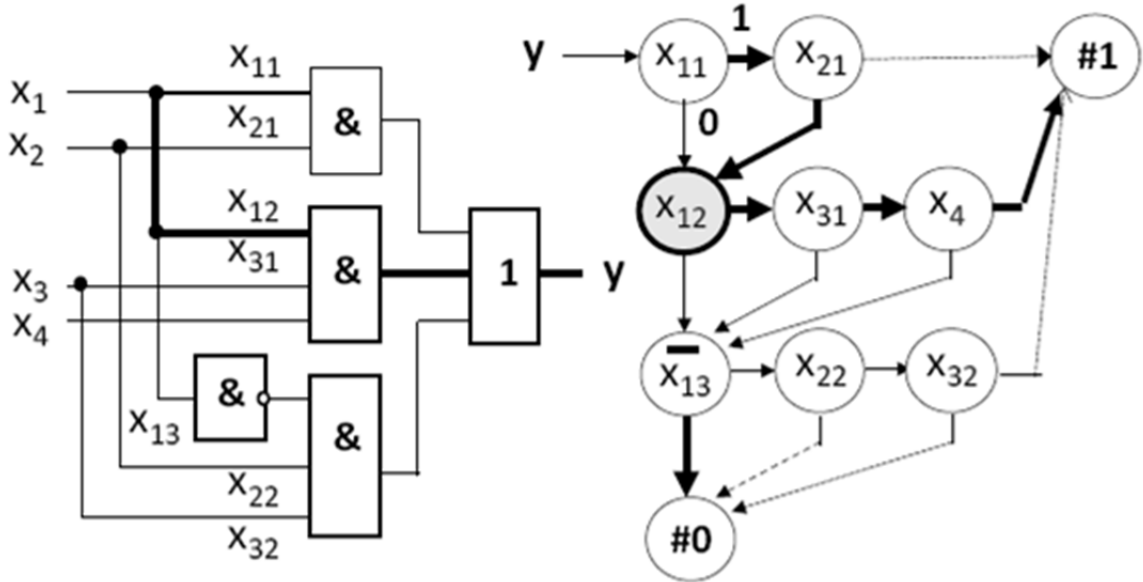


Fig. 1. Combinational circuit and its SSBDD

Example 1. In Fig. 1 we have a circuit with a FFR-module which can be represented by a Boolean expression:

$$y = (x_{11}x_{12}) \vee x_{12}x_{31}x_4(\bar{x}_{13}x_{22}x_{32})$$

and the SSBDD for the FFR part of the circuit. The literals with two indexes in the formula and in the SSBDD denote the branches of fan-out stems, and represent *signal paths* in the circuit. In this example, there are only two branches for each fan-out, the second index 1 is for the upper branch in the circuit, and the second index 2 is for the lower branch. For instance, the SSBDD node labeled by the variable x_u represents the bold signal path in the circuit. The variables in the nodes of SSBDD, in general, may be inverted. They are inverted when the number of invertors on the corresponding signal path in the circuit is odd. The two terminal nodes of the SSBDD are labeled by Boolean constants #1(truth) and #0(false). Let us agree that the right hand edges are always labeled by value 1, and lower hand edges by 0.

Every combinational circuit can be regarded as a network of modules, where each module represents an FFR of maximum size. This way of modeling the circuit by

BDDs allows to keep the complexity of the model (the total number of nodes in all graphs) linear to the number of gates in the circuit. In Table 1, a comparison of numbers of nodes for representing ISCAS'85 circuits by ROBDD [20], FBDD [21], and SSBDD models is presented.

Table 1

Comparison of sizes of different BDDs

Circuit	#Gates	ROBDD	FBDD	SSBDD
c432	232	30200	1063	308
c499	618	49786	25866	601
c880	357	7655	3575	497
c1355	514	39858	N/A	809
c1908	718	12463	5103	866
c2670	997	N/A	1815	1313
c3540	1446	208947	21000	1648
c5315	1994	32193	1594	2712
c6288	2416	N/A	N/A	3872
c7552	2978	N/A	2092	3552

As a side effect of the synthesis of SSBDDs, we build up a strict one-to-one mapping between the nodes in SSBDDs and the signal paths in the modules (FFRs) of the circuit. The algorithm for synthesis of SSBDDs is presented in [18, 19]. The optimization issues of SSBDDs are discussed in [22]. Different properties of SSBDD which facilitate efficient processing of SSBDDs for test generation and fault diagnosis purposes are discussed in [23].

Since all the *stuck-at faults* (SAF) on the inputs of an FFR form a collapsed fault set of the FFR, and since all these faults are represented by the faults at the nodes of the related SSBDD, it follows that the synthesis of an SSBDD, described in [18, 19] is equivalent to the fault collapsing procedure similar to fault folding [24].

Direct relation of nodes to signal paths allows to handle with SSBDDs easily such problems like fault modeling, fault collapsing, and fault masking.

2. Fault reasoning using topology of S SBDD

Consider an FFR-module of a circuit which implements a function $y = f(X)$ where X is the set of input variables of the module, and is represented by SSBDD with a set of nodes M . Let $x(m) \in X$ the variable at the node $m \in M$, and let m^0 and m^1 be the neighbors of the node m for the assignments $x(m) = 0$ and $x(m) = 1$, respectively.

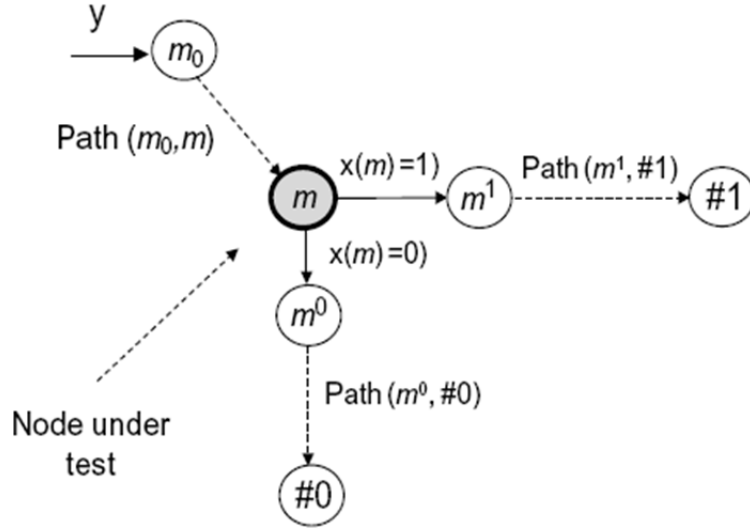


Fig. 2. Topological view on testing of nodes on the SSBDD

Activation of SSBDD paths. Let T_t be a pattern applied at the moment t on the inputs X of the module. The edge (m, m^e) in SSBDD, where $e \in \{0, 1\}$, is called *activated* by T_t if $x(m) = e$. A path (m, n) is called activated by T_t if all the edges which form the path are activated. To activate a path (m, n) means to assign by T_t proper values to the node variables along this path.

Test generation with SSBDD. A test pattern T_t will detect a *single stuck-at-fault* (SSAF) $x(m) \equiv e$, $e \in \{0, 1\}$, if it activates in the SSBDD three paths (Fig.2): a path (m_0, m) from the root node to the node under test, two paths $(m^0, \#0)$, $(m^1, \#1)$ for fault-free and faulty cases, and satisfies the fault activation condition $x(m) = e \otimes 1$.

Assume $e = 1$. To simulate the test experiment for T_t generated for the fault $x(m) \equiv 1$, first, the path (m_0, m) will be traced up to the node m which will "serve as a switch". If the fault is missing, the path $(m^0, \#0)$ will be traced, and if the fault is present, the path $(m^1, \#1)$ will be activated.

Note, that a test pattern T_t for a node fault $x(m) \equiv e$ detects single SAFs on all the lines of the signal path in the circuit, which is represented by the node m in SSBDD.

Example 2. Consider the fault $x_{12} \equiv 0$ in the circuit of Fig.1, represented by the fault $x(m) = x_{12} \equiv 0$ in the SSBDD. To generate a test pattern T_t for $x_{12} \equiv 0$, we have to activate three paths: $(m_0, m) = (x_{11} = 1, x_{21} = 0)$, $(m^1, \#1) = (x_{11} = 1, x_{31} = 1, x_4 = 1, \#1)$, and $(m^0, \#0) = (x_{13} = 1, \#0)$. For the node under test we take $x_{12} = 1$ which means that the expected value of test will be $y = 1$. From this, the test pattern $X' = (x_1, x_2, x_3, x_4)' = 1011$ results. If the fault $x_{12} \equiv 0$ will be present, the path $(x_{11}, x_{12}, x_{13}, \#0)$ will be activated, and the value $\#0$ in the terminal node will indicate the presence of the fault.

Fault simulation and diagnosis with SSBDDs. For fault simulation of a test pattern X' on the SSBDD first, the path $l = l(m_0, m^T)$ from the root node m_0 to a terminal node $m^T \in \{\#0, \#1\}$, activated by the pattern X' , is determined. Then, for each node $m \in l$, whose successor m^* does not belong to the path l , the path $l_{m^*} = l(m^*, m^{T*})$ will be simulated for the pattern X' . If $m^{T*} \neq m^T$ then the fault of the node m is detected by X' , otherwise not. Special properties called "direction rule" allow to define the nodes $m \in l$, which can be excluded from analysis to speed up the simulation [25].

Example 3. Consider a test pattern $X' = (x_1, x_2, x_3, x_4)' = 1011$ which activates the path $l = (x_{11}, x_{21}, x_{12}, x_{31}, \#1)$ shown bold in Fig. 1. According to the algorithm described, we can find that by the given test pattern the faults $x_{12} \equiv 0$, $x_{31} \equiv 0$, $x_4 \equiv 0$ can be detected.

The fault simulation procedure described can be used in fault diagnosis based on the effect-cause fault location concept to locate the fault candidates.

Fig. 3 illustrates average speed-up achieved by using SSBDDs for different simulation algorithms like logic simulation, fault simulation [26, 27], timing simulation [28] and multi-valued simulation [29] compared to gate-level simulation algorithms. The fault simulation shows the most noticeable acceleration. Other simulation algorithms vary in decreasing the runtime by 2,5 up to almost 4 times compared to algorithms working on the gate-level netlist model. This effect is possible due to shift from lower gate level to a higher macro level when working with SSBDD model (as macros, the FFRs are considered).

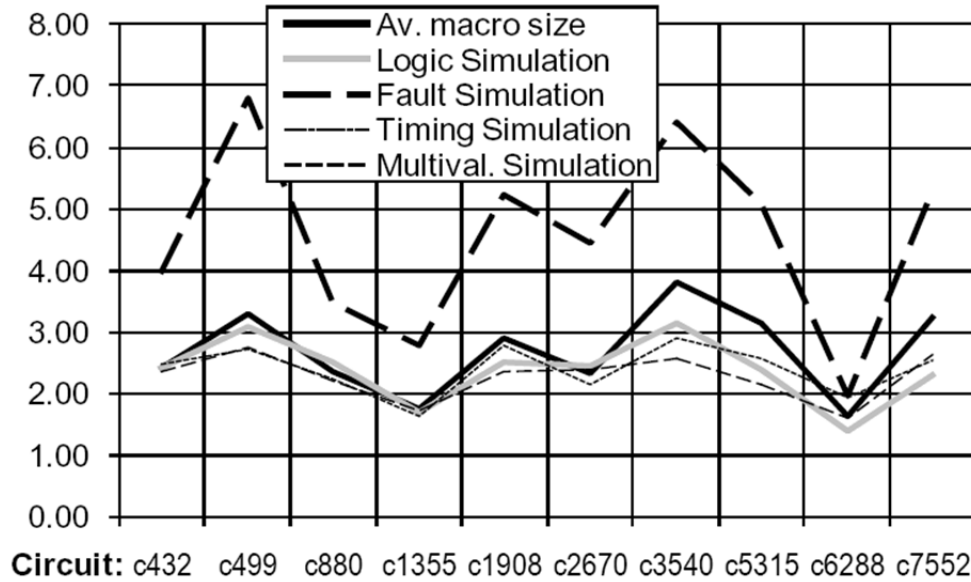


Fig. 3. Logic level simulation speed-up for different algorithms

SSBDDs facilitate well parallel processing of paths. This gave a possibility to develop extremely fast exact parallel critical path tracing algorithms which exceed the speed of commercial fault simulators, used in the industry [26, 27].

3. Multiple fault reasoning with SSBDDs

SSBDDs serve as an efficient tool for reasoning masking relationships in the presence of multiple faults. The basis of such a reasoning is the topological analysis of changes in the activation of paths because of faults.

A well known concept of test pairs [30–32] has been introduced for test generation which avoids fault masking in case of possible multiple faults. Let us call two test patterns $TP = (T_0, T_1)$ a *test pair* for testing a node $x(m)$, where T_0 is for testing $x(m) \equiv 0$, T_1 is for testing $x(m) \equiv 1$, and both patterns differ from each other only in the value of $x(m)$.

Consider a topology of an SSBDD in Fig.4a with the highlighted root node m_0 , two terminal nodes #0, #1, and two faulty nodes $a \equiv 0$, $c \equiv 1$. The dotted lines represent activated paths during a test pair $TP = \{T_0, T_1\}$ which has the goal to test the node a . T_0 is for activating the correct path $L_1 = (m_0, a, \#1)$ for detecting the fault $a \equiv 0$ with expected test result #1. If the fault is present, instead of L_1 , a “faulty” path $L_0 = (m_0, a \equiv 0, c, \#0)$ should be activated with faulty result #0. Thus, the fault $a \equiv 0$ should be detected.

In case of the masking fault $c \equiv 1$ on L_0 , a masking path $L_M = (m_0, a \equiv 0, c \equiv 1, \#1)$ will be activated, and the fault under test $a \equiv 0$ will not be detected by T_0 . The role of the second pattern T_1 , according to the test pair concept, is to activate the path $L_0 = (m_0, a, \#0)$ with two goals: (1) to detect $a \equiv 1$ (in the single fault case), or (2) to detect the masking fault $c \equiv 1$ (if the masking took place at T_0). At T_1 the path L_M should remain activated because of the masking fault $c \equiv 1$, and the wrong test result #1 will indicate the presence of a fault in the circuit.

The added value of a test pair is that if both patterns will pass then the wire under test (and the whole related signal path in the circuit) is proved to be correct at any present multiple fault.

Unfortunately, the described properties of the test pair will not always be sufficient for detecting multiple faults. The topological view on SSBDDs allows to develop efficient algorithms to avoid fault masking in multiple fault cases.

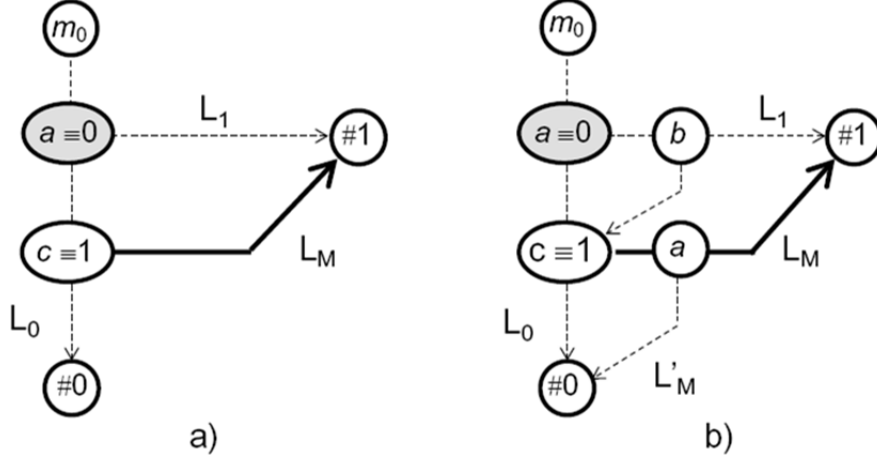


Fig. 4. Topological view on testing of nodes on the SSBDD

First, we show by simple reasoning of the topology of activated paths in SSBDDs why the test pairs are not always working as expected. Consider once more in Fig.4b the same test pair case $TP = \{T_0, T_1\}$ in the presence of the multiple fault $\{a \equiv 0, c \equiv 1\}$. Compared to Fig.4a, we have now additional node a on the masking path L_M , labeled by the same variable as the node under test $a \equiv 0$ (in grey). T_0 will again show the correct value $\#1$ because of fault masking. At T_1 the value of a was changed from 1 to 0 compared to T_0 . Because of this change, instead of the masking path $L_M = (m_0, a \equiv 0, c \equiv 1, \#1)$, a new “demasking” path $L'_M = (m_0, a \equiv 0, c \equiv 1, a, \#0)$ will be activated. Hence, both patterns will pass, and the multiple fault remains undetected.

The main idea of the test pair concept is to keep the activation of the masking path stable during both patterns. In Fig.4b, this condition is not fulfilled, since all the three paths, L_1 for correct case, L_0 for single fault case ($a \equiv 0$), and L_M for the multiple fault case ($a \equiv 0, c \neq 1$), involve the variable a under test, and the changing value of a will make the activation of the masking path L_M unstable.

In [33], a new method of test groups was developed and discussed which extends the method of test pairs and removes its drawbacks. The method of test groups essentially is based on the topological analysis of SSBDDs.

Example 4. Let us add, as an example, to TP a third pattern T_2 for testing the node b (for $b \equiv 1$) on L_1 . The goal of T_2 is to keep the masking path $L_M = (m_0, a = 1, b = 0, c \equiv 1, a = 1, \#1)$ again activated. The three patterns T_0 (for testing $a \equiv 0$ and $b \equiv 0$), T_2 (for testing $a \equiv 1$) and T_3 (for testing $b \equiv 1$) can be regarded as a *test group* for testing two nodes a and b . In this example, the first two patterns will pass, however, the pattern T_2 will not pass, and detect the given multiple fault.

The test groups are targeting not a single SSBDD node, rather a selected subset of SSBDD nodes [33] which represent a subcircuit of the given gate-level network. The main property and the main importance of test groups is that they are robust with respect to multiple faults. In other words, from passing of a given test group the correctness of the related subcircuit results. The test pair can be considered as a special case of the test group which works in “special” cases.

4. Shared SSBDDs

In [34], a new type of SSBDDs called *Structurally Synthesized Multiple Input BDDs* or Shared SSBDDs (SSSBDD) was introduced. The goal was to further compress the SSBDD model by exploiting the effect of superposition of SSBDDs.

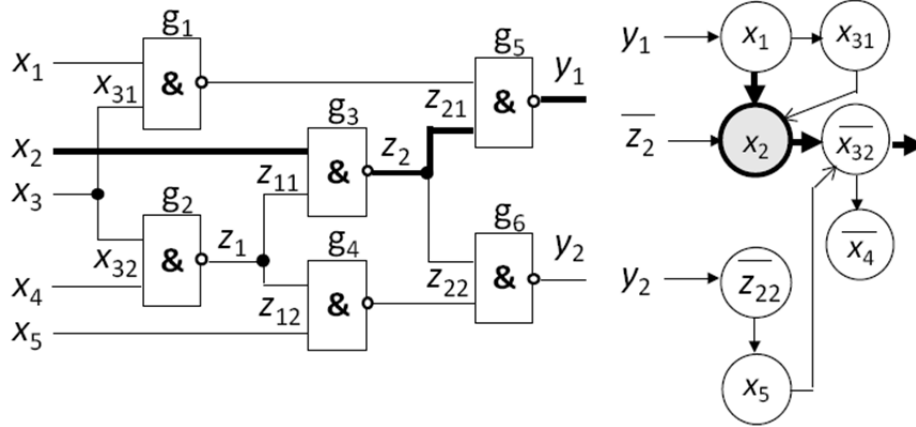


Fig. 5. Digital circuit c17 and its SSSBDD

Example 5. An example of a combinational circuit and its SSSBDD is presented in Fig.5. For simplicity we omit the terminal nodes, and agree that leaving the graph to the right (down) means entering the terminal node #1 (#0). The graph with 7 nodes represents only 14 collapsed stuck-at faults as targets for test generation instead of 32 faults in the lines of the original gate-level circuit. The graph joins three subgraphs for outputs y_x and y_2 , and for internal node z_2 with the shared subgraph with nodes $\neg x_{32}$ and $\neg x_4$. The nodes x_1 and x_5 represent signal paths in the circuit from x_1 to y_1 and from x_5 to y_2 , respectively. The node x_{31} represents the path from the lower input of g_1 to y_x . The node x_2 represents the path from the upper input of g_1 to y_1 . The nodes $\neg x_{32}$ and $\neg x_4$ represent the paths from the inputs of g_2 to both outputs y_1 and y_2 . And, finally, the node $\neg x_{22}$ represents the path from the upper input of g_6 to y_2 . By bold lines in SSSBDD a path is shown, which leads to assignments $x_1 = 0$ and $x_3 = 0$ for testing the bold path in the circuit from x_2 to y_1 . More detailed discussion of using SSSBDDs where the graphs for different output functions are merged into the same graph can be found in [34].

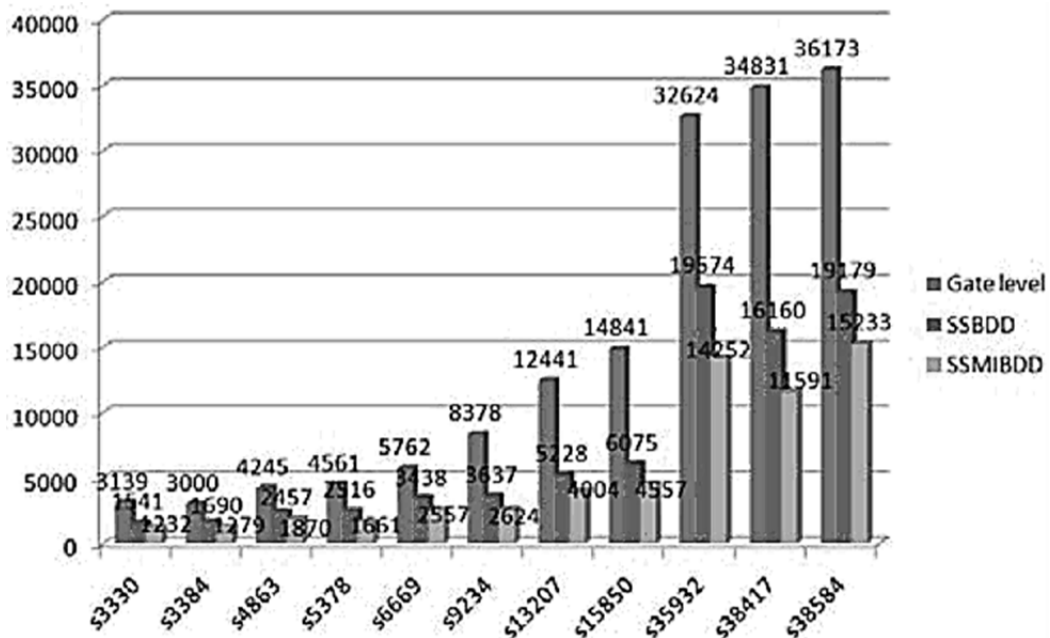


Fig. 6. Comparison of complexities of SSBDDs and SSSBDDs

A comparison of the reduction of complexities for gate-level circuits, SSBDDs and SSSBDDs in terms of fault collapsing was carried out for ISCAS'89 circuits. The differences in the number of nodes are shown

in Fig. 6. The average minimization gained for SSSBDDs in the number of nodes (and also in the size of collapsed fault sets) is up to 2.4 times compared to the gate level, and up to 1.4 times compared to the SSBDD model. The results prove that the SSSBDD model is more compact than the previously discussed SSBDD or gate level models, and as the result allows better fault collapsing which in its turn has the influence on the efficiency and speed of test generation and fault simulation.

5. Overview of high level DDs

The most important impact of the high-level DDs (HLDD) is the possibility of generalization and extension of the methods for test generation, fault simulation and diagnosis, developed for logic level circuits, to higher abstraction levels of digital systems using the uniform graph topology based formalism. For this purpose, the class of variables was extended from Boolean ones to Boolean vectors or integer variables, and the class of Boolean functions was extended to the data manipulation operations typically used in high-level descriptions of digital systems.

Example 6. In Fig. 7, an example of a RTL data-path and its HLDD is presented. The variables R_1 and R_2 represent registers, IN denotes the input bus, the integer variables y_1, y_2, y_3, y_4 represent control signals, M_1, M_2, M_3 are multiplexers, and the functions $R_1 + R_2$ and $R_1 * R_2$ represent the adder and multiplier, respectively. Each node in the DD represents a subcircuit of the system (e.g. the nodes y_1, y_2, y_3, y_4 represent multiplexers and decoders). The whole DD describes the behavior of the input logic of the register R_2 . To test a node in the DD means to test the corresponding to the node component or subcircuit.

Depending on the class of the system (or its representation level), we may have various HLDDs where the nodes have different interpretations and relationships to the system structure. In the RTL descriptions, we usually partition the system into control and data paths. In this case, the non-terminal nodes in the HLDDs correspond to the control path, and they are labeled by state or output variables of the control part, interpreted as addresses or instruction words. On the other hand, the terminal nodes in the HLDDs correspond to the data path, and they are labeled by the data words or functions of data words, which correspond to buses, registers, or data manipulation blocks. The state transfer and output functions of control circuits are represented as well by HLDDs. When using HLDDs for describing complex digital systems, we have to represent the system by a suitable set of interconnected components (combinational or sequential subcircuits). Thereafter, we have to describe the components by their functions which can be represented as HLDDs.

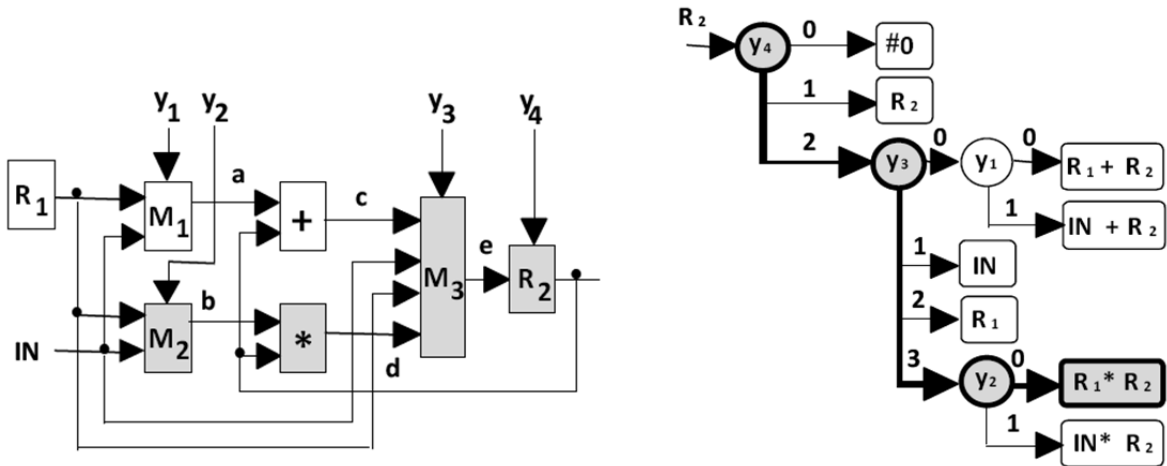


Fig. 7. Representing a register transfer level data path by a HLDD

Two methods for synthesis of HLDDs for representing digital systems were described in [19, 35]. The first one is based on symbolic execution of procedural descriptions, which corresponds to the functional representation of systems. The method can be used in cases when the system is given functionally as a

procedure in a hardware description language. The second method is based on iterative superposition of HLDDs, and the created model corresponds to the high-level structural representation of the system. The method can be used in cases when the system is given structurally as a network of components (subsystems), and for each component its HLDD is already given. The second method can be regarded as a generalization of the superposition procedure for BDDs [19].

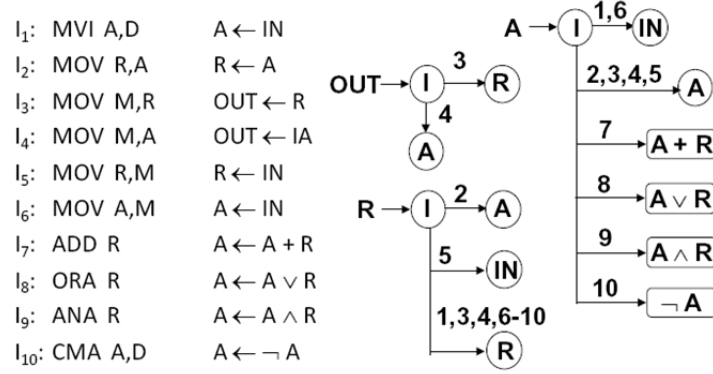


Fig. 8. HLDDs for a hypothetical microprocessor on the ISA level

Example 7. An example of behavior level HLDDs is shown in Fig. 8 for representing a hypothetical microprocessor given at the Instruction Set Architecture (ISA) level by its instruction set list. The model consists of three DDs: OUT, A, and R for representing the processor's output behavior, accumulator A, and register R, respectively.

6. Diagnostic modeling of digital systems with HLDDs

The methods for test generation and fault simulation developed for SSBDDs can be easily generalized for using at higher abstraction levels of systems [36]. The possibility of generalization results from the topological similarity of DDs at lower and higher levels (Fig. 9). In case of SSBDDs, each node has two output edges, and the graph has two terminal nodes $m^{T,0}$ and $m^{T,1}$ with constants 0 and 1, respectively. HLDDs differ from SSBDDs in having more edges from nodes and more terminal nodes $m^{T,1}, m^{T,2}, \dots, m^{T,n}$, whereas the terminal nodes in general case may be labeled by constants, register variables or functional expressions. Both graphs represent a mapping into the structure of the system they describe.

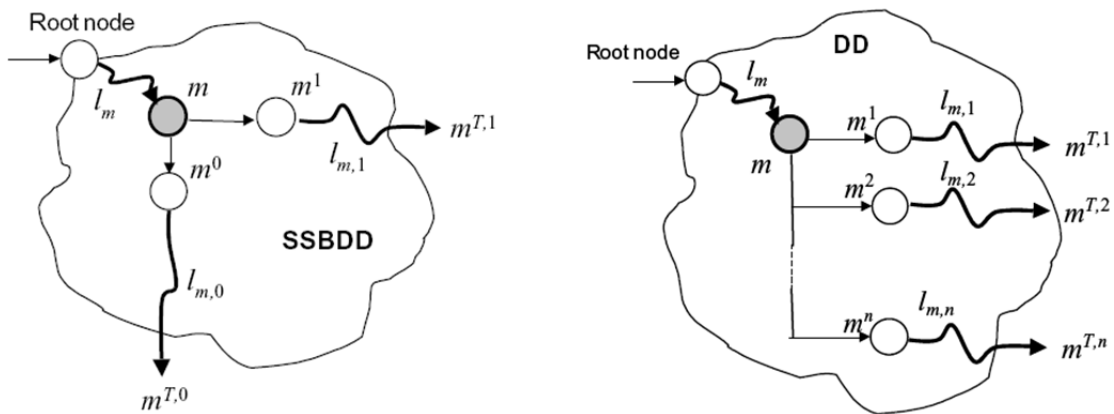


Fig. 9. Topological similarities of SSBDDs and HLDDs

In both cases, the faults in the system can be modeled similarly by errors at the nodes, and for both types of graphs, test generation for a given node re is carried out by activating a path from the root node to re and

from all successor nodes of re to corresponding terminal nodes. It is easy to see that the SSBDD can be regarded as a special case of HLDD. Similarly, as we defined the operations of logic simulation and path activation for SSBDDs we can do the same for HLDDs.

Example 8. In test pattern simulation, a path is traced in the graph, guided by the values of system variables until a terminal node is reached, similarly as in the case of SSBDDs. For example, in Fig.7, the result of simulating the vector $X' = (y_1, y_2, y_3, y_4, R_1, R_2, IN) = -, 0, 3, 2, 10, 6, -$ is $R_2 = R_1 * R_2 = 60$ (here "-" means *don't care*, the bold arrows in Fig.7 highlight the simulated path, and the grey node $R_1 * R_2$ is reached by simulation).

The advantage of HLDDs compared to the traditional methods of simulation of systems, lays in the fact that instead of processing of all the components in the RTL network for the given input pattern, in the HLDD only 3 control variables y_4, y_3, y_2 , were visited in this particular case during simulation, and only a single data manipulation operation $R_2 = R_1 * R_2$ was carried out.

Fault model on HLDDs. Each path in the HLDD describes the behavior of the system in a specific working mode. The faults having effect on the behavior can be associated with nodes along the path. A fault causes incorrect leaving the path activated by a test. From this point of view the following abstract fault model for nodes re with node variables $x(m)$ in HLDDs can be defined:

D1: the output edge for $x(m) = i$ of a node m is always activated, $x(m) = i$ (analog to logic level stuck-at-1);

D2: the output edge for $x(m) = i$ of a node m is broken (analog to logic level stuck-at-0);

D3: instead of the given edge for $x(m) = i$ of a node m , another edge for $x(m) = j$, or a set of edges $\{j\}$ is activated (analog to logic level multiple stuck-at-fault).

The fault model is directly related to the nodes m , and is an abstract one. It will have a semantic meaning only when the node has a particular physical interpretation. As an example, in Table 2 the correspondence of the HLDD-based fault model to different microprocessor fault classes [37], and RTL fault classes [38, 39] is shown.

Table 2

Comparison of High-Level fault models

Microprocessor faults [37]	DD faults	
F1: No source is selected F4: No destination selected	D1, D2	Internal nodes
F3: More than one source is selected F5: Instead of, or in addition to the selected destination, one or more other destinations are selected	D3	Internal nodes
F2: A wrong source is selected; F5: One or more other destinations are selected	D3	Internal nodes
F9-F14: Data storage, communication or manipulation faults	D3	Terminal nodes
RTL faults [38,39]	DD faults	
F15-F20: Control, addressing, timing, condition, decoding faults)	D1,D2,D3	Internal nodes
F21-F23: Data storage, communication or manipulation faults	D3	Terminal nodes

Test generation. Without going into details regarding fault handling, consider the following simplified idea of test generation for the nodes of HLDD.

To generate a test pattern for testing an internal node re in HLDD, $(n + 1)$ paths are to be activated: first, a path (m_0, m) , and second, n paths $l_e = (m^e, m^{T,e})$ for all values e of the variable $x(m)$, so that

$$x(m^{T,1}) \neq x(m^{T,2}) \neq \dots \neq x(m^{T,n}).$$

All paths should be activated consistently by the same test pattern (or sequence) T' . The test T' includes as well the data found by solving the inequality.

The test program for an internal node m (*conformity test* of the control part), consists of n experiments to exercise all the possible n values of $x(m)$ [40].

To test a terminal node $m^{T,i}$ a path $(m_0, m^{T,i})$ is activated. The test program (*scanning test*) generated for $m^{T,1}$ will be repeated for all local test patterns for testing the module with function $x(m^{T,i})$. The local test patterns may be generated at lower level using for instance SSBDs [40].

Example 9. As an example, consider test generation for testing the multiplexer M_3 represented by the node y_3 in the HLDD in Fig. 7. We activate, first, the path from the root node y_4 to the node y_3 under test by assigning $y_4 = 2$. Second, we activate 4 paths from the successors of y_3 , for each value $e = 0, 1, 2, 3$ of y_3 . Two of the paths, l_1, l_2 , for values $e = 1$ and $e = 2$, respectively, are activated “without action”, since the successors of y_3 for these values are terminal nodes. Other two paths l_0 and l_3 may be activated, for example, by $y_1 = 0$ and $y_2 = 0$, respectively. The test data $R_1 = D_1, R_2 = D_2, IN = D_3$ are found by satisfying the inequality $R_1 + R_2 \neq IN \neq R_1 * R_2$.

Note, by the described procedure, a test pattern is created for earring out the test for a selected HLDD node (a structural unit of a system) at the given state of the system (i.e. content of the system registers). In the full test sequence, the needed load operations as well as the operations for reading out the test result should be included. These operations can be formally generated as well from the HLDD model of the system [19].

From above, the following test program results:

Test program for control part:

Fore = 1,2,3,4

BEGIN

Load the data registers:

$R_1 = D_1, R_2 = D_2$;

Carry out the tested working mode at:

$y_3 = e, y_1 = 0, y_2 = 0, y_4 = 2$ and $IN = D_3$;

Read the test response $R_{2,e}$

END.

Example 10. As another example, consider test generation for testing the multiplier M_3 represented by the node $R_1 * R_2$ in Fig. 7. By activating the path to this node (shown in bold in Fig. 7) we generate a control word $(y_2, y_3, y_4) = (0, 3, 2)$. To find the proper values of R_1 and R_2 we need to descend to the lower abstraction level of hierarchy (e.g. to the gate level) and generate test patterns by a low level ATPG for the low level implementation of the multiplier. Let us have got a low level test set of n data patterns $(D_{1,1}, D_{2,1}; D_{1,2}, D_{2,2}; D_{1,1}, \dots, D_{1,n}, D_{2,n})$ generated for the multiplier with input registers R_1 and R_2 .

From above, the following test program results:

Test program for data part:

For all the values of $t = 1, 2, \dots, n$

BEGIN

Load the data registers:

$R_1 = D_{1,t}, R_2 = D_{2,t}$,

Carry out the tested working mode at the

 control values $(y_2, y_3, y_4) = (0, 3, 2)$;

Read the test response of $R_{2,t}$

END.

HLDDs have been used in different fields of high-level and hierarchical test and verification. As the result, new promising algorithms, techniques and prototype tools have been developed, which allowed to improve the efficiency of RTL cycle based simulation [41, 42], hierarchical test program automated synthesis [40, 43], hierarchical fault simulation [44], high-level verification [46], fault diagnosis [47, 48], and automated design error correction [48, 49].

Conclusion

An overview was given about two types of Decision Diagrams - SSBDDs and HLDDs for diagnostic modeling of digital systems, particularly for fault simulation and test generation. The main focus of both models is on the topological view on the graphs and on representing in DDs besides the functions the implementation details of the structure of the system as well.

Acknowledgment: The work has been supported by FP7 1ST project DIAMOND, and Research Centre CEBE funded by EU Structural Funds.

REFERENCES

1. Lee C.Y. Representation of Switching Circuits by Binary Decision Programs // The Bell System Technical Journal. 1959. P. 985-999.
2. Ubar R. Test Generation for Digital Circuits with Alternative Graphs // Proceedings of Tallinn Technical University. 1976. No. 409. P. 75-81.
3. Akers S.B. Functional Testing with Binary Decision Diagrams // J. of Design Automation and Fault-Tolerant Computing. 1978. V. 2. P. 311-331.
4. Bryant R.E. Graph-based algorithms for Boolean function manipulation // IEEE Trans on Comp. 1986. V. C-35, No. 8. P. 667-690.
5. Sasao T., Fujita M. (eds.). Representations of Discrete Functions. Kluwer Academic Publishers. 1996.
6. Drechsler R., Becker B. Binary Decision Diagrams. Kluwer Academic Publ., 1998.
7. Minato S., Ishiura N. Shared binary decision diagrams with attributed edges for efficient Boolean function manipulation // Proc. 27th IEEE/ACM ICCAD. 1990. P. 52-57.
8. Srinivasan A., Kam T., Malik S., Bryant R.E. Algorithms for discrete function manipulation // Proc. of Informations Conference on CAD-ICCAD. 1990. P. 92-95.
9. Kebschull U., Schubert E., Rosenstiel W. Multilevel logic synthesis based on functional decision diagrams. IEEE EDAC, 1992.
10. Minato S. Zero-suppressed BDDs for set manipulation in combinational problems // Proc. 30th DAC. 1995. P. 272-277.
11. Bahar R., Frohm E., Gaona C., Hachtel G., Macii E., Pardo A., Somenzi F. Algebraic decision diagrams and their applications // Int. Conf. on CAD. 1993. P. 188-191.
12. Drechsler R., Sarabi A., Theobald M., Becker B., Perkowski M.A. Efficient representation and manipulation of switching functions based on Ordered Kronecker Functional Decision Diagrams // Proc. DAC. 1994.
13. Bryant R.E., Chen Y.-A. Verification of arithmetic functions with binary moment diagrams // 32nd ACM/IEEE DAC. 1995.
14. Bern J., Meinel C., Slobodova A. Efficient OBDD-based manipulation in CAD beyond current limits // 32-nd DAC. 1995. P. 408-413.
15. Clarke E., Fujita N., Zhao X. Multi-terminal binary decision diagrams and hybrid decision diagrams, In: T. Sasao, M. Fujita (eds.), Representations of Discrete Functions. Kluwer Academic Publishers, 1996. P. 93-108.
16. Stankovic R., Astola J., Stankovic M., Egiazarjan K. Circuit synthesis from Fibonacci decision diagrams // VLSI Design, Special Issue on Spectral Techniques and Decision Diagrams. 2002. V. 14. P. 23-34.
17. Karpovsky M.G., Stankovic R.S., Astola J.T. Spectral Logic and Its Applications for the Design of Digital Devices. Wiley-Interscience, 2008.
18. Ubar R. Test Synthesis with Alternative Graphs // IEEE Design&Test of Computers. Spring, 1996. P. 48-57.
19. Ubar R., Raik J., AJutman, Jenihhin M. Diagnostic Modeling of Digital Systems with Multi-Level DDs // Design and Test Technology for Dependable SoC, R. Ubar, J. Raik, H.Th. Vierhaus (Eds.). 2011. P. 92-118.
20. Brace K.S., Rudell R.L., Bryant R.E. Efficient Implementation of a BDD Package // Proc. of the 27th DAC. June 1990. P. 40-45.
21. Giinther W., Drechsler R. Minimization of Free BDDs // Proc. of Asia and South Pacific Design Automation Conf. Hong Kong, Jan 1999. P. 323-326.
22. Ubar R., Vassiljeva T., Raik J., AJutman, Tombak M., Peder A. Optimization of Structurally Synthesized BDDs // IASTED Conf. on Modelling, Simulation and Optimization, Kauai, Hawaii, USA, August 17-19. 2004. P. 234-240.
23. Peder A., Nestra H., Raik J., Tombak M., Ubar R. Linear algorithms for testing and parsing superpositional graphs // Facta Universitatis (Nis) Ser.: Elec. Energ. 2011. V. 24, No. 3. P. 325-339.
24. To K. Fault Folding for Irredundant and Redundant Combinational Circuits // IEEE Trans, on Computers. 1973. No. 11. P. 1008-1015.
25. Ubar R. Overview about Low-Level and High-Level Decision Diagrams for Diagnostic Modeling of Digital Systems // Facta Universitatis (Nis) Ser.: Elec. Energ. 2011. V. 24, No. 3. P. 303-324.
26. Ubar R., Devadze S., Raik J., AJutman. Fast Fault Simulation in Digital Circuits with Scan Path // IEEE Proc. of 13th Asia and South Pacific Design Automation Conference - ASP-DAC. 2008. P. 667-672.
27. Ubar R., Devadze S., Raik J., AJutman. Parallel X-Fault Simulation with Critical Path Tracing Technique // IEEE Conf. Design, Automation & Test in Europe – DATE. 2010. P. 1-6.
28. AJutman, Ubar R., Peng Z. Algorithms for Speeding-Up Timing Simulation of Digital Circuits // DATE. 2001. P. 460-465.

29. Ubar R. Multi-Valued Simulation of Digital Circuits with Structurally Synthesized Binary Decision Diagrams. OPA, N.V. Gordon and Breach Publishers, Multiple Valued Logic. 1998. V. 4. P. 141-157.
30. Cox H., Rajski J. A Method of Fault Analysis for Test Generation and Fault Diagnosis // IEEE Trans. on CAD. 1988. V. 7. No. 7. P. 813-833.
31. Kajihara S., Nishigaya R., Sumioka T., Kinoshita K. Efficient Techniques for Multiple Fault Test Generation // 3rd ATS. 1994. P. 52-56.
32. Agrawal A., Saldanha A., Lavagno L. Compact and Complete Test Set Generation for Multiple Stuck-at Faults // ICCAD'93. 1996. P. 212-219.
33. Ubar R., Kostin S., Raik J. Multiple Stuck-at-Fault Detection Theorem // The 15th IEEE Symposium on Design and Diagnostics of Electronic Circuits and Systems – DDECS. Tallinn, Estonia, April 18-20. 2012.
34. Ubar R., Mironov D., Raik J., AJutman. Structural Fault Collapsing by Superposition of BDDs for Test Generation // ISQED. 2010. P. 250-257.
35. Ubar R., Raik J., Karputkin A., Tombak A. Synthesis of High-Level Decision Diagrams for Functional Test Pattern Generation // 16th Int. Conference MIXDES. 2009. P. 519-524.
36. Raik J., Ubar R. Fast Test Pattern Generation for Sequential Circuits Using Decision Diagram Representations. Journal of Electronic Testing: Theory and Applications. Kluwer Academic Publishers. 2000. V. 16, No. 3. P. 213-226.
37. Thatte S.M., Abraham J.A. Test Generation for Microprocessors // IEEE Trans. On Comp. 1980. V. C-29, No. 6. P. 429-441.
38. Gupta A.K., Armstrong J.R. Functional Fault modelling and Simulation for VLSI Devices // 22nd Design Automation Conference. 1985. P. 720-726.
39. Ubar R., Raik J., AJutman, Instenberg M., Wuttke H.-D. Modeling Microprocessor Faults on High-Level Decision Diagrams. International Conference on Dependable Systems & Networks: Anchorage, USA, Alaska, June 24-27. 2008. P. 17-22.
40. Jervan G., Ubar R., Peng Z., Eles P. Test Generation: A Hierarchical Approach // System-level Test and Validation of HW/SW Systems. by M.Sonza Reorda, Z.Peng, M.Violante. Springer Series in Advanced Microelectronics. 2005. V. 17. P. 63-77.
41. Xeveugle R., Ubar R. Modeling VHDL Clock-Driven Multi-Processes by Decision Diagrams // J. of Electron Technology. 1999. V. 32, No. 3. P. 282-287.
42. Ubar R., Morawiec A., Raik J. Cycle-Based Simulation Algorithms for Digital Systems Using High-Level Decision Diagrams // DATE. 2000. P. 743.
43. Raik J., Ubar R. Fast Test Pattern Generation for Sequential Circuits Using Decision Diagram Representations. Journal of Electronic Testing: Theory and Applications. Kluwer Academic Publishers. 2000. V. 16, No. 3. P. 213-226.
44. Ubar R., Devadze S., Jenihhin M., Raik J., Jervan G., Ellervee P. Hierarchical Calculation of Malicious Faults for Evaluating the Fault-Tolerance // 4th IEEE DELTA. 2008. P. 222-227.
45. Jenihhin M., Raik J., Fujiwara H., Ubar R., Viilukas T. An Approach for Verification Assertions Reuse in RTL Test Pattern Generation // Proc. of the IEEE 11th Workshop on RTL and High Level Testing -WRTL'10. 2010. P. 1-6.
46. Raik J., Repinski U., Ubar R., Jenihhin M., Chepurov A. High-Level Design Error Diagnosis Using Backtrace on Decision Diagrams // The 28th IEEE NORCHIP Conference, Tampere. Nov. 15-16. 2010.
47. Raik J., Repinski U., Ubar R., Jenihhin M., Chepurov A. High-Level Design Error Diagnosis Using Backtrace on Decision Diagrams // The 28th IEEE NORCHIP Conference, Tampere. Nov. 15-16. 2010.
48. Guarnieri V., DiGuglielmo G., Bombieri N., Pravadelli G., Fummi F., Hantson H., Raik J., Jenihhin M., Ubar R. On the Reuse of TLM Mutation Analysis at RTL // J. Electron Test, DOI 10.1007/s10836-012-5303-6. 2012.
49. Karputkin A., Ubar R., Tombak M., Raik J. Automated Correction of Design Errors by Edge Redirection on High-Level Decision Diagrams // IEEE International Symposium on Quality Electronic Design - ISQED, Santa Clara, CA USA, 2012.

Убар Раймонд, д-р техн. наук, профессор. E-mail: raiub@pld.ttu.ee
Таллиннский технический университет (Эстония)

Поступила в редакцию 3 февраля 2014 г.

Убар Раймонд. (Таллиннский технический университет, Эстония)

Влияние неисправностей цифровых систем на топологию решающих диаграмм низкого и высокого уровней.

Ключевые слова: решающие диаграммы; генерация тестов; диагностика неисправностей.

Для тестирования сложных цифровых систем необходимо исследовать их иерархические многоуровневые представления. Рассматриваются Structurally Synthesized Binary Decision Diagram (SSBDD) решающие диаграммы, обеспечивающие соответствие между полюсами таких диаграмм и вентилями цифровых систем. Методы синтеза тестов и диагностики неисправностей, разработанные на основе использования SSBDD решающих диаграмм, развиваются для описаний цифровых систем на более высоком уровне абстракции. Речь идет об использовании High Level Decision Diagram (HLDD) решающих диаграмм для автоматической генерации тестов и диагностики неисправностей сложных цифровых систем.

REFERENCES

1. Lee C.Y. Representation of switching circuits by binary decision programs. *The Bell System Technical Journal*, 1959, pp. 985-999. DOI: 10.1002/j.1538-7305.1959.tb01585.x.

2. Ubar R. Test generation for digital circuits with alternative graphs. *Proceedings of Tallinn Technical University*, 1976, no. 409, pp.75-81. DOI: 10.1007/3-540-58426-9_129.
3. Akers S.B. Functional testing with binary decision diagrams. *J. of Design Automation and Fault-Tolerant Computing*, 1978, vol. 2, pp. 311-331. DOI: 10.1109/TC.1986.1676774.
4. Bryant R.E. Graph-based algorithms for Boolean function manipulation. *IEEE Trans on Comp*, 1986, vol. C-35, no. 8, pp. 667-690. DOI: 10.1109/TC.1986.1676819.
5. Sasao T., Fujita M. (eds.). *Representations of discrete functions*. Kluwer Academic Publ., 1996. 331 p.
6. Drechsler R., Becker B. *Binary decision diagrams*. Kluwer Academic Publ., 1998. 200 p.
7. Minato S., Ishiura N. Shared binary decision diagrams with attributed edges for efficient Boolean function manipulation. *Proc. 27th IEEE/ACM ICCAD*, 1990, pp. 52-57. DOI: 10.1145/123186.123225.
8. Srinivasan A., Kam T., Malik S., Bryant R.E. Algorithms for discrete function manipulation. *Proc. of Informations Conference on CAD -ICCAD*, 1990, pp. 92-95. DOI: 10.1109/ICCAD.1990.129849.
9. Kebschull U., Schubert E., Rosenstiel W. Multilevel logic synthesis based on functional decision diagrams. *IEEE EDAC*, 1992.
10. Minato S. Zero-suppressed BDDs for set manipulation in combinational problems. *Proc. 30th DAC*, 1995, pp. 272-277. DOI: 10.1145/157485.164890.
11. Bahar R., Frohm E., Gaona C., Hachtel G., Macii E., Pardo A., Somenzi F. Algebraic decision diagrams and their applications. *Int. Conf. on CAD*, 1993, pp. 188-191. DOI: 10.1109/ICCAD.1993.580054.
12. Drechsler R., Sarabi A., Theobald M., Becker B., Perkowski M.A. Efficient representation and manipulation of switching functions based on Ordered Kronecker Functional Decision Diagrams. *Proc. DAC*, 1994. DOI: 10.1145/196244.196444.
13. Bryant R.E., Chen Y.-A. Verification of arithmetic functions with binary moment diagrams. *32nd ACM/IEEE DAC*, 1995. DOI: 10.1145/217474.217583.
14. Bern J., Meinel C., Slobodova A. Efficient OBDD-based manipulation in CAD beyond current limits. *32-nd DAC*, 1995, pp. 408-413. DOI: 10.1145/217474.217563.
15. Clarke E., Fujita N., Zhao X. *Multi-terminal binary decision diagrams and hybrid decision diagrams*. In: Sasao T., Fujita M. (eds.) *Representations of discrete functions*. Kluwer Academic Publishers, 1996, pp. 93-108.
16. Stankovic R., Astola J., Stankovic M., Egiazarjan K. Circuit synthesis from Fibonacci decision diagrams. *VLSI Design, Special Issue on Spectral Techniques and Decision Diagrams*, 2002, vol.14, pp. 23-34. DOI: 10.1080/10655140290009783.
17. Karpovsky M.G., Stankovic R.S., Astola J.T. *Spectral logic and its applications for the design of digital devices*. Wiley-Interscience, 2008. 500 p.
18. Ubar R. Test synthesis with alternative graphs. *IEEE Design&Test of Computers*, Spring, 1996, pp. 48-57. DOI: 10.1109/54.485782.
19. Ubar R., Raik J., Jutman A., Jenihhin M. *Diagnostic Modeling of Digital Systems with Multi-Level DDs*. In: Ubar R., Raik J., Vierhaus H.Th. (eds.) *Design and test technology for dependable SoC*. 2011, pp. 92-118.
20. Brace K.S., Rudell R.L., Bryant R.E. Efficient implementation of a BDD Package. *Proc. of the 27th DAC*, June 1990, pp. 40-45. DOI: 10.1145/123186.123222.
21. Giinther W., Drechsler R. Minimization of Free BDDs. *Proc. of Asia and South Pacific Design Automation Conf, Hong Kong*, Jan 1999, pp. 323-326. DOI: 10.1016/S0167-9260(02)00041-X.
22. Ubar R., Vassiljeva T., Raik J., Jutman A., Tombak M., Peder A. Optimization of structurally synthesized BDDs. *IASTED Conf. on Modelling, Simulation and Optimization*, Kauai, Hawaii, USA, August 17-19, 2004, pp. 234-240.
23. Peder A., Nestra H., Raik J., Tombak M., Ubar R. Linear algorithms for testing and parsing superpositional graphs. *Facta Universitatis (Nis) Ser.: Elec. Energ.*, 2011, vol. 24, no. 3, pp. 325-339. DOI: 10.2298/FUEE1103325P.
24. To K. Fault folding for irredundant and redundant combinational circuits. *IEEE Transactions on Computers*, 1973, no.11, pp. 1008-1015. DOI: 10.1109/T-C.1973.223637.
25. Ubar R. Overview about low-level and high-level decision diagrams for diagnostic modeling of digital systems. *Facta Universitatis (Nis) Ser.: Elec. Energ.*, 2011, vol. 24, no. 3, pp. 303-324. DOI: 10.2298/FUEE1103303U.
26. Ubar R., Devadze S., Raik J., Jutman A. Fast fault simulation in digital circuits with scan path. *IEEE Proc. of 13th Asia and South Pacific Design Automation Conference - ASP-DAC*, 2008, pp. 667-672.
27. Ubar R., Devadze S., Raik J., Jutman A. Parallel X-fault simulation with critical path tracing technique. *IEEE Conf. Design, Automation & Test in Europe - DATE*, 2010, pp. 1-6. DOI: 10.1109/DATE.2010.5456929.
28. Jutman A., Ubar R., Peng Z. Algorithms for speeding-up timing simulation of digital circuits. *DATE*, 2001, pp. 460-465.
29. Ubar R. Multi-valued simulation of digital circuits with structurally synthesized binary decision diagrams. *OPA, N.V. Gordon and Breach Publishers, Multiple Valued Logic*, 1998, vol. 4, pp. 141-157.
30. Cox H., Rajski J. A method of fault analysis for test generation and fault diagnosis. *IEEE Transactions on CAD*, 1988, vol. 7, no. 7, pp. 813-833. DOI: 10.1109/43.3952.
31. Kajihara S., Nishigaya R., Sumioka T., Kinoshita K. Efficient techniques for multiple fault test generation. *3rd ATS*, 1994, pp. 52-56. DOI: 10.1109/ATS.1994.367254.
32. Agrawal A., Saldanha A., Lavagno L. Compact and complete test set generation for multiple stuck-at faults. *ICCAD'93*, 1996, pp. 212-219.
33. Ubar R., Kostin S., Raik J. Multiple stuck-at-fault detection theorem. *The 15th IEEE Symposium on Design and Diagnostics of Electronic Circuits and Systems - DDECS*, Tallinn, Estonia, April 18-20, 2012. DOI: 10.1109/DDECS.2012.6219064

34. Ubar R., Mironov D., Raik J., Jutman A. Structural fault collapsing by superposition of BDDs for test generation. *ISQED*, 2010, pp. 250-257. DOI: 10.1109/ISQED.2010.5450451
35. Ubar R., Raik J., Karputkin A., Tombak A. Synthesis of high-level decision diagrams for functional test pattern generation. *16th Int. Conference MIXDES*, 2009, pp. 519-524.
36. Raik J., Ubar R. Fast test pattern generation for sequential circuits using decision diagram representations. *Journal of Electronic Testing: Theory and Applications*, 2000, vol. 16, no. 3, pp. 213-226. DOI: 10.1023/A:1008335130158.
37. Thatte S.M., Abraham J.A. Test generation for microprocessors. *IEEE Transactions On Comp.*, 1980, vol. C-29, no. 6, pp. 429-441. DOI: 10.1109/TC.1980.1675602.
38. Gupta A.K., Armstrong J.R. Functional fault modelling and simulation for VLSI Devices. *22nd Design Automation Conference*, 1985, pp.720-726. DOI: 10.1109/DAC.1985.1586022
39. Ubar R., Raik J., Jutman A., Instenberg M., Wuttke H.-D. Modeling microprocessor faults on high-level decision diagrams. *International Conference on Dependable Systems & Networks: Anchorage*, USA, Alaska, June 24-27, 2008, pp. 17-22.
40. Jervan G., Ubar R., Peng Z., Eles P. *Test generation: A hierarchical approach*. In: Sonza Reorda M., Peng Z., Violante M. (eds.) *System-level test and validation of HW/SW systems*. Springer Series in Advanced Microelectronics, 2005, vol. 17, pp. 63-77. DOI: 10.1007/1-84628-145-8_5.
41. Xeveugle R., Ubar R. Modeling VHDL clock-driven multi-processes by decision diagrams. *Journal of Electron Technology*, 1999, vol. 32, no. 3, pp. 282-287.
42. Ubar R., Morawiec A., Raik J. Cycle-based simulation algorithms for digital systems using high-level decision diagrams. *DATE*, 2000, pp. 743.
43. Raik J., Ubar R. Fast test pattern generation for sequential circuits using decision diagram representations. *Journal of Electronic Testing: Theory and Applications*, 2000, vol. 16, no. 3, pp. 213-226. DOI: 10.1023/A:1008335130158.
44. Ubar R., Devadze S., Jenihhin M., Raik J., Jervan G., Ellervee P. Hierarchical calculation of malicious faults for evaluating the fault-tolerance. *4th IEEE DELTA*, 2008, pp. 222-227. DOI: 10.1109/DELTA.2008.60.
45. Jenihhin M., Raik J., Fujiwara H., Ubar R., Viilukas T. An approach for verification assertions Reuse in RTL test pattern generation. *Proc. of the IEEE 11th Workshop on RTL and High Level Testing -WRTL'10*, 2010, pp. 1-6.
46. Raik J., Repinski U., Ubar R., Jenihhin M., Chepurov A. High-level design error diagnosis using backtrace on decision diagrams. *The 28th IEEE NORCHIP Conference*, Tampere. November 15-16, 2010. DOI: 10.1109/NORCHIP.2010.5669486.
47. Raik J., Repinski U., Ubar R., Jenihhin M., Chepurov A. High-level design error diagnosis using backtrace on decision diagrams. *The 28th IEEE NORCHIP Conference*, Tampere. November 15-16. 2010. DOI: 10.1109/NORCHIP.2010.5669486.
48. Guarnieri V., DiGuglielmo G., Bombieri N., Pravadelli G., Fummi F., Hantson H., Raik J., Jenihhin M., Ubar R. On the Reuse of TLM mutation analysis at RTL. *Journal of Electronic Testing*, 2012. DOI 10.1007/s10836-012-5303-6.
49. Karputkin A., Ubar R., Tombak M., Raik J. Automated correction of design errors by edge redirection on high-level decision diagrams. *IEEE International Symposium on Quality Electronic Design - ISQED*, Santa Clara, CA USA, 2012. DOI: 10.1109/ISQED.2012.6187566.

СВЕДЕНИЯ ОБ АВТОРАХ

БАБАНОВ Алексей Михайлович – доцент, кандидат технических наук, доцент кафедры программной инженерии факультета информатики Томского государственного университета. E-mail: babanov2000@mail2000.ru

БЕККЕРМАН Екатерина Николаевна – старший преподаватель кафедры информационных технологий в исследовании дискретных структур радиофизического факультета Томского государственного университета. E-mail: koranet@mail.ru

БУКРЕЕВ Виктор Григорьевич – профессор, доктор технических наук, профессор кафедры ЭПЭО, ЭНИН Томского политехнического университета. E-mail: bukreev@tpu.ru

БУРДОНОВ Игорь Борисович – доктор физико-математических наук, ведущий научный сотрудник Института системного программирования РАН (г. Москва). E-mail: igor@ispras.ru

ГОРЦЕВ Александр Михайлович – профессор, доктор технических наук, декан факультета прикладной математики и кибернетики Томского государственного университета. E-mail: dekanat@fpmk.tsu.ru

ЕФАНОВ Дмитрий Викторович – кандидат технических наук, доцент кафедры «Автоматика и телемеханика на железных дорогах» Петербургского государственного университета путей сообщения. E-mail: TrES-4b@yandex.ru

КЕЛЛЕР Юрий Александрович – кандидат физико-математических наук, доцент кафедры исследования операций факультета прикладной математики Томского государственного университета. E-mail: kua1102@rambler.ru

КОСАЧЕВ Александр Сергеевич – кандидат физико-математических наук, ведущий научный сотрудник Института системного программирования РАН (г. Москва). E-mail: kos@ispras.ru

КИТУРКО Ольга Михайловна – аспирантка факультета математики и информатики Гродненского государственного университета им. Я. Купалы (Республика Беларусь). E-mail: sytaya_om@mail.ru

МАТАЛЫЦКИЙ Михаил Алексеевич – профессор, доктор физико-математических наук, заведующий кафедрой стохастического анализа и эконометрического моделирования Гродненского государственного университета им. Я. Купалы (Республика Беларусь). E-mail: m.matalytski@gmail.com

МОИСЕЕВ Александр Николаевич – доцент, кандидат технических наук, доцент кафедры программной инженерии факультета информатики Томского государственного университета. E-mail: moiseev.tsu@gmail.com

САПОЖНИКОВ Валерий Владимирович – профессор, доктор технических наук, профессор кафедры «Автоматика и телемеханика на железных дорогах» Петербургского государственного университета путей сообщения. E-mail: kat@pgups.edu

САПОЖНИКОВ Владимир Владимирович – профессор, доктор технических наук, профессор кафедры «Автоматика и телемеханика на железных дорогах» Петербургского государственного университета путей сообщения. E-mail: kat@pgups.edu

ПАРАЕВ Юрий Иванович – профессор, доктор технических наук, заведующий кафедрой прикладной математики факультета прикладной математики и кибернетики Томского государственного университета. E-mail: paraev@mail.ru

ПОЛИТОВ Арсентий Михайлович – магистрант факультета информатики Томского государственного университета. E-mail: a.m.politov@gmail.com

РАХИМОВ Валентин Валентинович – магистрант факультета информатики Томского государственного университета. E-mail: rakhimovvv@gmail.com

УБАР Раймунд – профессор, доктор технических наук, профессор кафедры компьютерной инженерии и диагностики Таллиннского технического университета (Эстония). E-mail: raiub@pld.ttu.ee

ХОМИЧ Максим Олегович – магистрант факультета информатики Томского государственного университета. E-mail: mohomich@gmail.com

ЧЕРЕМИСИНОВА Людмила Дмитриевна – доктор технических наук, главный научный сотрудник Объединенного института проблем информатики национальной академии наук Беларуси (г. Минск. Республика Беларусь). E-mail: cld@newman.bas-net.by

ШИЛИН Александр Анатольевич – кандидат технических наук, доцент кафедры ЭПЭО, ЭНИН Томского политехнического университета. E-mail: shilin@tpu.ru

Научный журнал

**ВЕСТНИК
ТОМСКОГО
ГОСУДАРСТВЕННОГО
УНИВЕРСИТЕТА**

**УПРАВЛЕНИЕ,
ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА
И ИНФОРМАТИКА**

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

2014. № 3 (28)

Редактор Н.А. Афанасьева
Оригинал-макет А.И. Лелоюр
Редакторы-переводчики: Г.М. Кошкин; В.Н. Горенинцева
Дизайн обложки Л.Д. Кривцова

Подписано к печати 17.09.2014 г. Формат 60x84¹/₈.

Гарнитура Times.

Печ. л. 14,5; усл. печ. л. 13,4.

Тираж 250 экз. Заказы № 528, 529.

Журнал отпечатан на полиграфическом оборудовании
Издательского Дома Томского государственного университета
634050, г. Томск, Ленина, 36
Телефон 8+(382-2)–53-15-28