

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Национальный исследовательский
Томский политехнический университет»



ISSN 1684-8519

ИЗВЕСТИЯ

ТОМСКОГО ПОЛИТЕХНИЧЕСКОГО УНИВЕРСИТЕТА

Том 322, № 5, 2013

Управление, вычислительная техника
и информатика

ИЗДАТЕЛЬСТВО  ТПУ

г. Томск

ИЗВЕСТИЯ ТОМСКОГО ПОЛИТЕХНИЧЕСКОГО УНИВЕРСИТЕТА

Редакционный совет:

Чубик П.С. (председатель), д.т.н., ректор ТПУ (г. Томск)
Пестряков А.Н. (заместитель председателя),
д.х.н., проректор ТПУ по НИИ (г. Томск)
Алексеев С.В., д.ф.-м.н.,
член-корреспондент РАН (г. Новосибирск)
Болдырев В.В., д.х.н., академик РАН (г. Новосибирск)
Боровиков Ю.С., к.т.н. (г. Томск)
Гуляев Ю.В., д.ф.-м.н., академик РАН (г. Москва)
Дамам Ж., PhD (Франция)
Дмитриев А.Ю., к.т.н. (г. Томск)
Долматов О.Ю., к.т.н. (г. Томск)
Ершов Ю.Л., д.ф.-м.н., академик РАН (г. Новосибирск)
Замятин А.В., к.т.н. (г. Томск)
Клименов В.А., д.т.н. (г. Томск)
Конторович А.Э., д.г.-м.н., академик РАН (г. Новосибирск)
Крёнинг М., PhD (Германия)
Летников Ф.А., д.г.-м.н., академик РАН (г. Иркутск)
Месяц Г.А., д.ф.-м.н., академик РАН (г. Москва)
Михайленко Б.Г., д.ф.-м.н.,
академик РАН (г. Новосибирск)
Нагоряков В.Е., д.т.н., академик РАН (г. Новосибирск)
Панин В.Е., д.ф.-м.н., академик РАН (г. Томск)
Сигов А.С., д.ф.-м.н., академик РАН (г. Москва)
Сигфуссон Т.И., PhD (Исландия)
Турнаев В.И., д.и.н. (г. Томск)
Филлипов Г.А., д.т.н., академик РАН (г. Москва)
Чайковский Д.В., к.ф.н. (г. Томск)
Шень Джаоли, PhD (Китай)
Яковлев А.Н., к.ф.-м.н. (г. Томск)

Редакционная коллегия:

Пестряков А.Н. (председатель, главный редактор), д.х.н.
Коробейников А.Ф. (зам. председателя), д.г.-м.н.
Глазырин А.С. (главный редактор Издательства ТПУ), к.т.н.
Могильницкий С.Б. (учёный секретарь), к.ф.-м.н.
Барышева Г.А., д.э.н.
Григорьев В.П., д.ф.-м.н.
Заворин А.С., д.т.н.
Иванчина Э.Д., д.т.н.
Ильин А.П., д.ф.-м.н.
Корниенко А.А., д.ф.н.
Лавринович В.А., д.т.н.
Погребной В.К., д.т.н.
Савичев О.Г., д.г.н.
Тузовский А.Ф., д.т.н.
Шаманин И.В., д.ф.-м.н.

Журнал зарегистрирован Министерством
Российской Федерации по делам печати,
телерадиовещания и средств массовых коммуникаций.
Свидетельство ПИ № 77-16615 от 24 октября 2003 г.
Учредитель: Томский политехнический университет

Издается с 1903 г.

© ФГБОУ ВПО НИ ТПУ, 2013

BULLETIN OF THE TOMSK POLYTECHNIC UNIVERSITY

Editorial Board:

Chubik P.S. (Chairman), D.E., rector of TPU (Tomsk)
Pestryakov A.N. (Deputy Chairman), D. Chem.,
pro-rector of TPU for Research and Innovation (Tomsk)
Aleksenko S.V., D. Phys. and Math. Sc.,
corresponding member of RAS (Novosibirsk)
Boldyrev V.V., D. Chem., member of RAS (Novosibirsk)
Borovikov Yu.S., Candidate of Science (Tomsk)
Gulyaev Yu.V., D. Phys. and Math. Sc., member of RAS (Moscow)
Damamm G., PhD (France)
Dmitriev A.Yu., Candidate of Science (Tomsk)
Dolmatov O.Yu., Candidate of Science (Tomsk)
Ershov Yu.L., D. Phys. and Math. Sc., member of RAS (Novosibirsk)
Zamyatin A.V., Candidate of Science (Tomsk)
Klimenov V.A., D.E. (Tomsk)
Kontorovich A.E., D. Geol. and Mineral. Sc., member of RAS (Novosibirsk)
Kröning M., PhD (Germany)
Letnikov F.A., D. Geol. and Mineral. Sc., member of RAS (Irkutsk)
Mesyats G.A., D. Phys. and Math. Sc., member of RAS (Moscow)
Mikhailenko B.G., D. Phys. and Math. Sc.,
member of RAS (Novosibirsk)
Nakoryakov V.E., D. E., member of RAS (Novosibirsk)
Panin V.E., D. Phys. and Math. Sc., member of RAS (Tomsk)
Sigov A.S., D. Phys. and Math. Sc., member of RAS (Moscow)
Sigfusson T.I., PhD (Iceland)
Turnaev V.I., PhD (Tomsk)
Fillipov G.A., D.E., member of RAS (Moscow)
Chaikovsky D.V., Candidate of Science (Tomsk)
Shen Zhaoli, PhD (China)
Yakovlev A.N., Candidate of Science (Tomsk)

Editorial:

Pestryakov A.N. (Chairman, Editor in Chief), D. Chem.
Korobeinikov A.F. (Deputy Editor in Chief), D. Geol. and Mineral. Sc.
Glazyrin A.S. (Editor in Chief), Candidate of Science
Mogilnitsky S.B. (Science Secretary), Candidate of Science
Barysheva G.A., Ec. D.
Grigoriev V.P., D. Phys. and Math. Sc.
Zavorin A.S., D.E.
Ivanchina E.D., D.E.
Ilyin A.P., D. Phys. and Math. Sc.
Kornienko A.A., Ph. D.
Lavrinovich V.A., D.E.
Pogrebnoy V.K., D.E.
Savichev O.G., D. Geog. Sc.
Tuzovsky A.F., D.E.
Shamanin I.V., D. Phys. and Math. Sc.

Подписной индекс по каталогу
Агентства «Роспечать» – 18054

Журнал рассылается в адреса 50-и библиотек РФ,
США, ФРГ, Великобритании, Франции
и 9-и стран ближнего зарубежья

Полнотекстовый доступ к электронной версии
журнала возможен на сайтах ТПУ: portal.tpu.ru/izvestiya/;
ООО «Научная электронная библиотека»:
www.elibrary.ru, www.e-library.ru,
а также поисковой системы scholar.google.com

Импакт-фактор РИНЦ 2011 г. 0,210

© Tomsk Polytechnic University, 2013

СОДЕРЖАНИЕ

ПРИКЛАДНАЯ МАТЕМАТИКА	
Исследование эффективности оптимальной передачи сигнала Рожкова С.В.	5
Распознавание состояний стохастических систем с фиксированной памятью Рожкова С.В.	8
Производные и интегралы дробных комплексных порядков функций дискретной переменной Чуриков В.А.	11
Выделение тренда дисперсии случайного процесса для пуассоновского потока моментов измерений Устинова И.Г.	14
Оценка функции автокорреляции в виде сплайна первого порядка Устинова В.Н., Устинова И.Г.	19
УПРАВЛЕНИЕ ТЕХНИЧЕСКИМИ СИСТЕМАМИ	
Управление опционными ресурсами Китаева А.В., Степанова Н.В.	23
Диалоговая процедура решения задач векторной оптимизации с неопределенными параметрами Реизлин В.И., Орлов В.А.	28
Анализ путей повышения эффективности расчетов частотно-временной корреляционной функции Лунева Е.Е., Аврамчук В.С.	33
Модель угроз системы физической защиты линейной части магистрального нефтепровода Кукало И.А., Кшнянкин А.П., Гривцов С.Н.	37
Идентификация эволюционных процессов жизненного цикла систем с учетом априорной информации Наймущин А.Г., Сергеев В.Л.	42
Синтез встраиваемой одноконтурной системы автоматического управления с самонастраивающимся регулятором и оценка ее робастности Курганкин В.В., Замятин С.В., Замятин В.М., Пушкарев М.И.	46
Автоматизация процесса нагнетания при гидродинамических исследованиях метаноугольных скважин Однокопылов И.Г., Гнеушев В.В., Сизиков Д.А., Шишляев В.В.	50
Исследование метода непараметрических оценок длины периода и периодической составляющей сигнала Марченко В.В., Берестнева О.Г., Девятых Д.В., Суханова Е.Ф.	55
Параллельные вычисления в задаче пространственного моделирования изменений ландшафтного покрова Замятин А.В., Афанасьев А.А.	60
Применение ячеек однородной структуры в задаче обнаружения шумовых гидроакустических сигналов на основе интегрального вейвлет-спектра Хамухин А.А.	68
Исследование температурного поля коаксиального шунта в программном комплексе ANSYS Бедарева Е.В., Костин Е.В.	72
Приборный комплекс для сбора и регистрации результатов измерений магнитных параметров электрофизической установки Меркулов С.В., Голобоков Ю.Н., Байструков К.И., Павлов В.М., Качкин А.Г., Мезентцев А.А., Ли А.М., Хохлаков В.С.	76
Разработка системы управления технологическим процессом переработки натриевого теплоносителя Кривцов П.Ю., Павлов В.М.	82
Программный модуль предупредительной сигнализации, аварийной защиты и помощи оператору вакуумной системы Токамак КТМ Коровиков А.Г., Павлов В.М., Ольховик Д.А.	88
Программный комплекс для имитационного моделирования момента сопротивления производственных механизмов Филипас А.А., Миколаенко С.Н., Кладиев С.Н.	93

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

О концепции интегрированной модели представления знаний Загоруйко Ю.А.	98
---	----

CONTENTS

APPLIED MATHEMATICS

The research of the efficiency of signal optimal transmission Rozhkova S.V.	5
Recognition of states of stochastic systems with fixed memory Rozhkova S.V.	8
The derivatives and integrals of fractional complex orders of discrete variable function Churikov V.A.	11
Selection of dispersion trend of random process for Poisson current of measuring moments Ustinova I.G.	14
Estimation of autocorrelation function in the form of first-order spline Ustinova V.N., Ustinova I.G.	19

ENGINEERING SYSTEM CONTROL

Option resource control Kitaeva A.V., Stepanova N.V.	23
Interactive procedure of solving vector optimization problems with uncertain parameters Reyzlin V.I., Orlov V.A.	28
The analysis of ways for increasing the efficiency when calculating time-and-frequency correlation function Luneva E.E., Avramchuk V.S.	33
The model of physical security threats to linear part of oil pipeline Kukalo I.A., Kshnyankin A.P., Grivtsov S.N.	37
Identification of evolution processes of a system life cycle considering a-priori information Naymushin A.G., Sergeev V.L.	42
Synthesis of embedded single-circuit automated control system with self-tuning controller and its robustness estimation Kurgankin V.V., Zamyatin S.V., Zamyatin V.M., Pushkarev M.I.	46
Injection automation at hydrodynamic researches of coal seams filtration characteristics Odnokopylov I.G., Gneushev V.V., Sizikov D.A., Shishlyayev V.V.	50
The research of nonparametric estimate method for the period length and signal periodic component Marchenko V.V., Berestneva O.G., Devyatykh D.V., Sukhanova E.F.	55
Parallel computing in the problem of land cover changes 3D modeling Zamyatin A.V., Afanasyev A.A.	60
Application of homogeneous structure cell when detecting noise hydroacoustic signals based on integrated wavelet spectrum Khamukhin A.A.	68
The research of coaxial shunt temperature field in ANSYS Bedareva E.V., Kostin E.V.	72
The system for acquisition and logging the results of measuring magnetic parameters of electrophysical unit Merkulov S.V., Golobokov Yu.N., Baystrukov K.I., Pavlov V.M., Kachkin A.G., Mezentssev A.A., Li A.M., Khokhryakov V.S.	76
The development of control system for sodium coolant processing Krivtsov P.Yu., Pavlov V.M.	82
Software module of safety alarm, emergency protection and aid to the operator of vacuum system of tokamak TKM Korovikov A.G., Pavlov V.M., Olkhovik D.A.	88
Software system for simulation of industrial mechanism load torque Filipas A.A., Mikolaenko S.N., Kladiev S.N.	93

INTELLIGENCE SYSTEMS

The concept of knowledge representation integrated model Zagorulko Yu.A.	98
---	----

Построение модели доверия к объектам автоматизированной системы для предотвращения деструктивного воздействия на систему Бешта А.А., Кирпо М.А.	104	Construction of object trust model in the automated information system for preventing destructive influence on the system Beshta A.A., Kirpo M.A.
Анализ информационных свойств взаимных фазовых спектров отраженных сейсмических волн Иванченков В.П., Кочегуров А.И., Черкасова М.А.	108	The analysis of information features of mutual phase spectra of reflected seismic waves Ivanchenkov V.P., Kochegurov A.I., Cherkasova M.A.
Система анализа катастрофоустойчивости Аткина В.С.	116	The analysis system of disaster tolerance Atkina V.S.
Распознавание структурированных символов на изображениях с использованием гистограмм средней интенсивности и сверточной нейронной сети Друки А.А.	120	Recognition of structured symbols on images using histogram of average intensity and convolution neural network Druki A.A.
Применение пространственных методов улучшения изображений на видеопоследовательностях Белоусов А.А., Спицын В.Г.	126	Application of 3D methods of video image enhancement Belousov A.A., Spitsyn V.G.
Разработка реляционных средств декомпозиции предикатов Рудометкина М.Н., Спицын В.Г., Эттель В.А.	131	Development of relational means of predicate decomposition Rudometkina M.N., Spitsyn V.G., Ettel V.A.
Метод контроля прямого доступа к семантическим базам данных Хоанг Ван Куэт, Тузовский А.Ф.	138	Control method of direct access to semantic data base Hoang Van Kuet, Tuzovsky A.F.
Разработка лингвистического обеспечения информационных систем на основе онтологических моделей знаний Сидорова Е.А.	143	The development of linguistic support of information systems on the basis of knowledge ontological models Sidorova E.A.
Методы определения уровней безопасности элементов онтологии Хоанг Ван Куэт, Тузовский А.Ф.	148	Methods for determining security levels of ontology elements Hoang Van Kuet, Tuzovsky A.F.
Метод трёхстороннего слияния онтологий на языке OWL 2 Заикин И.А.	153	The method of OWL 2 ontology three-way merge Zaikin I.A.
Семантическое аннотирование документов в электронных библиотеках Ле Хоай, Тузовский А.Ф.	157	Document semantic annotation in digital libraries Le Hoai, Tuzovsky A.F.
Формальная модель структуры взаимосвязей разнотипных объектов проектирования Вичугова А.А., Вичугов В.Н., Цапко Г.П.	164	Formal model of a structure of different design objects relationships Vichugova A.A., Vichugov V.N., Tsapko G.P.
АЛГОРИТМИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ		SOFTWARE AND ALGORITHMIC SUPPORT
Исследование программных реализаций алгоритмов вычисления CRC совместимых с PKZIP, WINRAR, ETHERNET Мыцко Е.А., Мальчуков А.Н.	170	The research of software implementation of CRC computing algorithms compatible with PKZIP, WINRAR, ETHERNET Mytsko E.A., Malchukov A.N.
Эффективность режимов внутреннего предсказания блоков в современных стандартах сжатия видео Шарабайко М.П., Марков Н.Г.	175	The efficiency of block intra prediction modes in modern standards of video compression Sharabayko M.P., Markov N.G.
Исследование аппаратных реализаций табличного и матричного алгоритмов вычисления CRC32 Мыцко Е.А., Мальчуков А.Н.	182	The research of hardware implementation of table- and matrix-driven algorithms of computing CRC32 Mytsko E.A., Malchukov A.N.
Оценивание числа клеток на изображениях цитологических растительных препаратов Тарков М.С., Осипов М.И.	187	Estimation of cell number on images of cytological herbal preparations Tarkov M.S., Osipov M.I.
Задача определения оценок сходства структур двух графов на основе выделения общих частей Погребной В.К.	194	The task of estimating the similarity of two graphs structures on the basis of general parts definition Pogrebnoy V.K.
Инвариант графа на основе компактных подграфов и алгоритм его вычисления Погребной Ан. В., Погребной В.К.	200	Invariant of a graph based on dense subgraphs and the algorithm of its computation Pogrebnoy An.V., Pogrebnoy V.K.
Программно-информационное обеспечение оценки параметров мощных СВЧ импульсов Косицын В.С., Гальченко В.Г., Гладкова Т.А.	205	Software and dataware for analyzing microwave pulse parameters Kositsyn V.S., Galchenko V.G., Gladkova T.A.
Онтологические системы перехода и их применение к семантике компьютерных языков Ануреев И.С.	209	Ontological transition systems and their application to computer language semantics Anureev I.S.
НАШИ ЮБИЛЯРЫ		ANNIVERSARIES
Профессору В.Г. Спицыну – 65 лет	214	Professor V.G. Spitsyn is 65
Профессору Б.Л. Аграновичу – 75 лет	217	Professor B.L. Agranovich is 75
РЕФЕРАТЫ СТАТЕЙ	219	SUMMARIES
СВЕДЕНИЯ ОБ АВТОРАХ	223	INFORMATION ABOUT AUTHORS

Прикладная математика

УДК 519.2

ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ ОПТИМАЛЬНОЙ ПЕРЕДАЧИ СИГНАЛА

С.В. Рожкова

Томский политехнический университет

E-mail: rozhkova@tpu.ru

Рассматривается задача исследования эффективности оптимальной передачи стохастических процессов по непрерывно-дискретным каналам с памятью и запаздыванием.

Ключевые слова:

Сигнал, стохастические системы, канал передачи, память, запаздывание.

Key words:

Signal, stochastic systems, transmission channel, memory, lag.

1. Постановка задачи

Данная работа является развитием результатов [1–4]. Система обозначений та же, что и в [1–5].

Сигнал x_t , сообщение на выходе канала передачи z_t и сообщение на выходе дискретного канала передачи $\eta(t_m)$ задаются на реализациях процессов, определяемых стохастическими дифференциальными уравнениями

$$dx_t = F(t)x_t dt + \Phi_1(t)dw_t, \quad p_0(x) = N\{x; \mu_0, \gamma_0\},$$

$$dz_t = h(t, x_t, x_\tau, z)dt + \Phi_2(t)dv_t,$$

$$\eta(t_m) = g(t_m, x_\tau, z) + \Phi_3(t_m)\xi(t_m),$$

Задача: в классе кодирующих функционалов $K = \{H; G\} = \{h^0(\cdot), g^0(\cdot)\}$, удовлетворяющих энергетическим ограничениям

$$M\{h^2(t, x_t, x_\tau, z)\} \leq \tilde{h}(t) \leq \tilde{h},$$

$$M\{g^2(t_m, x_{t_m}, x_\tau, z)\} \leq \tilde{g}(t_m) \leq \tilde{g},$$

найти функционалы $h^0(\cdot)$ и $g^0(\cdot)$, обеспечивающие относительно задачи фильтрации минимальную ошибку декодирования $\Delta^0(t) = \inf \Delta(t)$, где $\Delta(t) = M\{[x_t - \hat{x}(t, z, \eta)]^2\}$ является ошибкой оценки фильтрации $\hat{x}(t, z, \eta)$ процесса x_t , которая соответствует принятому сообщению $\{z_0^i; \eta_0^m\}$ при заданных $h^0(\cdot)$, $g^0(\cdot)$. Так как при заданных $h^0(\cdot)$ и $g^0(\cdot)$ оптимальной в среднеквадратическом смысле оценкой фильтрации является апостериорное среднее $\mu(t) = M\{x_t | z_0^i, \eta_0^m\}$, то $\Delta(t) \geq M\{\gamma(t)\}$, где $\gamma(t) \geq M\{[x_t - \mu(t)]^2 | z_0^i, \eta_0^m\}$. Таким образом $\Delta^0(t) = \inf M\{\gamma(t)\}$.

Данная задача была решена в [1] на классе кодирующих функционалов $K_l = \{H_l; G_l\}$, где

$$H_l = \begin{cases} h(\cdot) : h(t, x_t, x_\tau, z) = \\ = h(t, z) + H_0(t, z)x_t + H_1(t, z)x_\tau \end{cases},$$

$$G_l = \begin{cases} g(\cdot) : g(t_m, x_\tau, z) = \\ = g(t_m, z) + G_0(t_m, z)x_{t_m} + G_1(t_m, z)x_\tau \end{cases},$$

в [2] на классе кодирующих функционалов $K_l^1 = \{H_l^1; G_l^1\}$, где

$$H_l^1 = \{h(\cdot) : h(t, x_t, x_\tau, z) = h(t, z) + H_1(t, z)x_\tau\},$$

$$G_l^1 = \{g(\cdot) : g(t_m, x_\tau, z) = g(t_m, z) + G_1(t_m, z)x_\tau\},$$

в [3] на классе кодирующих функционалов $K_l^{0,1} = \{H_l; G_l^1\}$, где

$$H_l = \begin{cases} h(\cdot) : h(t, x_t, x_\tau, z) = \\ = h(t, z) + H_0(t, z)x_t + H_1(t, z)x_\tau \end{cases},$$

$$G_l^1 = \{g(\cdot) : g(t_m, x_\tau, z) = g(t_m, z) + G_1(t_m, z)x_\tau\}.$$

Рассмотрим эффективность оптимальной непрерывной передачи с памятью относительно оптимальной непрерывной передачи с запаздыванием в случае, когда оптимальная передача в дискретном канале является передачей с памятью.

Замечание. Так как при оптимальном способе передачи при энергетических ограничениях $M\{h^2(t, x_t, x_\tau, z)\} \leq \tilde{h}(t) \leq \tilde{h}$, $M\{g^2(t_m, x_{t_m}, x_\tau, z)\} \leq \tilde{g}(t_m) \leq \tilde{g}$ в каналах с памятью передаются только текущие зна-

чения процесса x_t , тогда согласно Замечанию 2 из [1] каналы с памятью и без памяти эквивалентны.

Согласно данному Замечанию поставленная задача эквивалента задаче исследования эффективности непрерывной передачи без памяти относительно непрерывной передачи с запаздыванием, когда дискретный канал без памяти.

В качестве мер эффективности непрерывной передачи без памяти относительно непрерывной передачи с запаздыванием при фильтрационном и интерполяционном приемах берем величины $\varepsilon_f(t) = \Delta^0(t)/\tilde{\Delta}^0(t)$ и $\varepsilon_i(t) = \Delta_{11}^0(t)/\tilde{\Delta}_{11}^0(t)$, где $\Delta^0(t)$ и $\Delta_{11}^0(t)$, $\Delta^0(t)$ и $\tilde{\Delta}_{11}^0(t)$ являются среднеквадратическими ошибками воспроизведения сигнала x_t .

Введем обозначения

$$\begin{aligned} t^* &= t - \tau, \quad \delta_c(t) = \tilde{h}(t)/R(t), \quad \delta_d(t_m) = \tilde{g}(t_m)/V(t_m), \\ \rho(\tau, t) &= (\Delta_{01}^0(\tau, t))^2 / \Delta^0(t) \Delta_{11}^0(\tau, t), \\ \tilde{\rho}(\tau, t) &= (\tilde{\Delta}_{01}^0(\tau, t))^2 / \tilde{\Delta}^0(t) \tilde{\Delta}_{11}^0(\tau, t). \end{aligned} \quad (1)$$

Очевидно, что величина t^* есть время запаздывания, $\delta_c(t)$ и $\delta_d(t_m)$ – отношения сигнал/шум по интенсивности в непрерывном и дискретном каналах при оптимальных способах передачи, $\rho(\tau, t)$ и $\tilde{\rho}(\tau, t)$ – квадраты коэффициентов корреляции между $\tilde{\mu}(t) = x_t - \mu^0(t)$ и $\tilde{\mu}(\mu, t) = x_t - \mu^0(\tau, t)$ при непрерывной передаче без памяти и с запаздыванием, соответственно. Данное исследование проведем для частного случая, когда

$$\begin{aligned} F(t) &= -a, \quad a > 0, \quad \tilde{h}(t) = \tilde{h}, \quad \tilde{g}(t_m) = \tilde{g}, \\ R(t) &= R, \quad V(t_m) = V, \end{aligned} \quad (2)$$

т. е. процесс x_t является процессом Орнштейна–Уленбека [5] и каналы передачи являются стационарными с постоянными параметрами.

Предполагается: 1) t_m – первый момент передачи в дискретном канале; 2) время τ является достаточно большим, когда на интервале $t \in [0, \tau]$ дифференциальное уравнение для $\Delta^0(t)$ при условиях (2) достигает стационарного значения $\Delta^0(\tau) = \Delta^0 = \text{const}$.

2. Основные результаты

Теорема 1.

1) Для ε_f и ε_i справедливы формулы

$$\varepsilon_f(t^*) = \Delta^0(t^*)/\tilde{\Delta}^0(t^*) \quad \text{и} \quad \varepsilon_i(t^*) = \Delta_{11}^0(t^*)/\tilde{\Delta}_{11}^0(t^*), \quad (3)$$

где

$$\begin{aligned} \Delta^0(t^*) &= [Q/(2a + \delta_c)] [1 - \exp\{-(2a + \delta_c)t^*\}] + \\ &+ \Delta^0(\tau) \exp\{-(2a + \delta_c)t^*\}, \end{aligned} \quad (4)$$

$$\begin{aligned} \tilde{\Delta}^0(t^*) &= [Q/2a] [1 - \exp\{-2at^*\}] + \\ &+ \Delta^0(\tau) \exp\{-(2a + \delta_c)t^*\}, \end{aligned} \quad (5)$$

$$\begin{aligned} \Delta_{11}^0(t^*) &= \Delta^0(\tau) [1 - \delta_c/(2a + \delta_c)] \times \\ &\times [1 - \exp\{-(2a + \delta_c)t^*\}], \end{aligned} \quad (6)$$

$$\tilde{\Delta}_{11}^0(t^*) = \Delta^0(\tau) \exp\{-\delta_c t^*\}, \quad (7)$$

$$\begin{aligned} \Delta^0(\tau) &= [Q/(2a + \delta_c)] [1 - \exp\{-(2a + \delta_c)\}] + \\ &+ \gamma^0 \exp\{-(2a + \delta_c)\tau\}. \end{aligned}$$

2) Пусть $\varepsilon_f(0) = \lim_{t \downarrow 0} \varepsilon_f(t)$ и $\varepsilon_i(0) = \lim_{t \downarrow 0} \varepsilon_i(t)$ при $t \downarrow 0$; $\varepsilon_f(\infty) = \lim_{t \uparrow \infty} \varepsilon_f(t)$ и $\varepsilon_i(\infty) = \lim_{t \uparrow \infty} \varepsilon_i(t)$ при $t \uparrow \infty$. Тогда $\varepsilon_f(t) \leq 1$ является функцией t , монотонно убывающей от значения $\varepsilon_f(0) = 1$ до значения $\varepsilon_f(\infty) = [1 + (\delta_c/2a)]^{-1}$ и $\varepsilon_i(t) \geq 1$ является функцией t , монотонно возрастающей от значения $\varepsilon_i(0) = 1$ до значения $\varepsilon_i(\infty) = \infty$.

Доказательство:

Решения уравнений (15), (14), (10) из [2] для $\tilde{\Delta}_{01}^0(\tau, t)$, $\tilde{\Delta}_{11}^0(\tau, t)$, $\tilde{\Delta}^0(t)$ с начальными условиями $\tilde{\Delta}_{01}^0(\tau, \tau) = \Delta_{01}^0(\tau, \tau) = \Delta^0(\tau)$, $\tilde{\Delta}_{11}^0(\tau, \tau) = \Delta_{11}^0(\tau, \tau) = \Delta^0(\tau)$ при выполнении (2) имеют вид

$$\begin{aligned} \tilde{\Delta}_{01}^0(\tau, t) &= \Delta^0(\tau) \exp\{-(a + \delta_c)(t - \tau)\}, \\ \tilde{\Delta}_{11}^0(\tau, t) &= \Delta^0(\tau) \exp\{-\delta_c(t - \tau)\}, \\ \tilde{\Delta}^0(t) &= [Q/2a] [1 - \exp\{-2a(t - \tau)\}] + \\ &+ \Delta^0(\tau) \exp\{-(2a + \delta_c)(t - \tau)\}. \end{aligned} \quad (8)$$

Тогда (5), (7) следуют из (8). Решение уравнения (10) из [1] с начальным условием $\Delta^0(\tau)$, при $t \geq \tau$ и выполнении (2) имеет вид

$$\begin{aligned} \Delta^0(t) &= [Q/(2a + \delta_c)] [1 - \exp\{-(2a + \delta_c)(t - \tau)\}] + \\ &+ \Delta^0(\tau) \exp\{-(2a + \delta_c)(t - \tau)\}. \end{aligned} \quad (9)$$

Тогда (4) следует из (9). Из [3] следует, что на кодировании (5) в [1] уравнения для $\Delta_{01}^0(\tau, t)$ и $\Delta_{11}^0(\tau, t)$ при выполнении (2) имеют вид

$$\begin{aligned} d\Delta_{01}^0(\tau, t)/dt &= -(a + \delta_c)\Delta_{01}^0(\tau, t), \\ d\Delta_{11}^0(\tau, t)/dt &= -\delta_c(\Delta_{01}^0(\tau, t))^2 / \Delta^0(t). \end{aligned}$$

При начальных условиях $\Delta_{01}^0(\tau, \tau) = \Delta^0(\tau)$ и $\Delta_{11}^0(\tau, \tau) = \Delta^0(\tau)$ решение уравнения для $\Delta_{01}^0(\tau, t)$ имеет свойство $\Delta_{01}^0(\tau, t) = \tilde{\Delta}_{01}^0(\tau, t)$, а решение уравнения для $\Delta_{11}^0(\tau, t)$ имеет вид

$$\Delta_{11}^0(\tau, t) = \Delta^0(\tau) \left(\frac{1 - [\delta_c/2(a + \delta_c)] \times [1 - \exp\{-2(a + \delta_c)(t - \tau)\}]}{1 - \exp\{-2(a + \delta_c)(t - \tau)\}} \right). \quad (10)$$

Тогда (6) следует из (10). Учитывая в (8), что на интервале непрерывный канал может быть только каналом без памяти, получаем $\tilde{\Delta}^0(\tau) = \Delta^0(\tau)$, и формула для $\Delta^0(\tau)$ следует из (9), где $\tau = 0$, $t = \tau$, $\Delta^0(0) = \gamma^0$. Свойства 2) следуют из (3)–(7).

Далее рассмотрим влияние непрерывных каналов на дискретную передачу без памяти при фильтрационном и интерполяционном приемах, вводя меры эффективности

$$\begin{aligned} \varepsilon_f(t_m) &= \Delta^0(t_m)/\Delta^0(t_m - 0), \\ \tilde{\varepsilon}_f(t_m) &= \tilde{\Delta}^0(t_m)/\tilde{\Delta}^0(t_m - 0), \\ \varepsilon_i(t_m) &= \Delta_{11}^0(\tau, t_m)/\Delta_{11}^0(\tau, t_m - 0), \\ \tilde{\varepsilon}_i(t_m) &= \tilde{\Delta}_{11}^0(\tau, t_m)/\tilde{\Delta}_{11}^0(\tau, t_m - 0) \end{aligned} \quad (11)$$

и предполагая, что t_m – первый момент передачи в дискретном канале.

Теорема 2.

1) Для $\varepsilon_f(t_m)$, $\tilde{\varepsilon}_f(t_m)$, $\varepsilon_i(t_m)$, $\tilde{\varepsilon}_i(t_m)$ справедливы формулы

$$\varepsilon_f(t_m) = \tilde{\varepsilon}_f(t_m) = [1 + \delta_d]^{-1}, \quad (12)$$

$$\begin{aligned} \varepsilon_i(t_m) &= \varepsilon_i(t_m; t^*) = [1 + \delta_d]^{-1} [1 + \delta_d (1 - \rho(t^*))], \\ \tilde{\varepsilon}_i(t_m) &= \tilde{\varepsilon}_i(t_m; t^*) = [1 + \delta_d]^{-1} [1 + \delta_d (1 - \tilde{\rho}(t^*))], \end{aligned} \quad (13)$$

$$\begin{aligned} \rho(t^*) &= \left[\frac{\delta_c}{2(a + \delta_c)} + \left(1 - \frac{\delta_c}{2(a + \delta_c)} \right) \times \right. \\ &\quad \left. \times \exp \{ 2(a + \delta_c) t^* \} \right]^{-1}, \\ \tilde{\rho}(t^*) &= \exp \{ -2at^* \}. \end{aligned} \quad (14)$$

2) Пусть: $\varepsilon_i(t_m; 0) = \lim_{t \downarrow 0} \varepsilon_i(t_m; t)$ и $\tilde{\varepsilon}_i(t_m; 0) = \lim_{t \downarrow 0} \tilde{\varepsilon}_i(t_m; t)$ при $t \downarrow 0$; $\varepsilon_i(t_m; \infty) = \lim_{t \uparrow \infty} \varepsilon_i(t_m; t)$ и $\tilde{\varepsilon}_i(t_m; \infty) = \lim_{t \uparrow \infty} \tilde{\varepsilon}_i(t_m; t)$ при $t \uparrow \infty$. Тогда $\varepsilon_i(t_m; t) \leq 1$ и $\tilde{\varepsilon}_i(t_m; t) \leq 1$ являются функциями t , монотонно возрастающими от значения $\varepsilon_i(t_m; 0) = \tilde{\varepsilon}_i(t_m; 0) = \varepsilon_i(t_m) = (1 + \delta_d)^{-1}$ до значения $\varepsilon_i(t_m; \infty) = \tilde{\varepsilon}_i(t_m; \infty) = 1$, и $\varepsilon_i(t_m; t) > \tilde{\varepsilon}_i(t_m; t)$ при конечных значениях $t > 0$.

Доказательство:

Из (12) в [1], (16) в [3] с учетом (1), получаем

$$\Delta^0(t_m) = [1 + \delta_d]^{-1} \Delta^0(t_m - 0),$$

$$\tilde{\Delta}^0(t_m) = [1 + \delta_d]^{-1} \tilde{\Delta}^0(t_m - 0),$$

$$\begin{aligned} \Delta_{11}^0(\tau, t_m) &= [1 + \delta_d]^{-1} [1 + \delta_d (1 - \rho(t^*))] \Delta_{11}^0(\tau, t_m - 0), \\ \tilde{\Delta}_{11}^0(\tau, t_m) &= [1 + \delta_d]^{-1} [1 + \delta_d (1 - \tilde{\rho}(t^*))] \tilde{\Delta}_{11}^0(\tau, t_m - 0). \end{aligned} \quad (15)$$

Тогда формулы (12), (13) следуют из (11), (15), а (14) следуют из (1), (6), (7), (10) с учетом $\Delta_{01}^0(t) = \tilde{\Delta}_{01}^0(t) = \Delta^0(\tau) \exp \{ -(a + \delta_c)t \}$. Свойства 2) следуют из (13), (14) с учетом того, что $\rho(t) < \tilde{\rho}(t)$ при конечных $t > 0$.

В качестве меры информационной эффективности для случаев фильтрационного и интерполяционного приемов берем величины

$$\Delta I_t = I_t[\cdot] - \tilde{I}_t[\cdot],$$

$$\Delta I_{t_m} = I_{t_m}[\cdot] - I_{t_m-0}[\cdot], \quad \Delta \tilde{I}_{t_m} = \tilde{I}_{t_m}[\cdot] - \tilde{I}_{t_m-0}[\cdot],$$

$$\Delta I_\tau^t = I_\tau^t[\cdot] - \tilde{I}_\tau^t[\cdot],$$

$$\Delta I_\tau^{t_m} = I_\tau^{t_m}[\cdot] - I_\tau^{t_m-0}[\cdot], \quad \Delta \tilde{I}_\tau^{t_m} = \tilde{I}_\tau^{t_m}[\cdot] - \tilde{I}_\tau^{t_m-0}[\cdot].$$

Теорема 3.

1) Для ΔI_t и ΔI_τ^t справедливы формулы

$$\Delta I_t(t^*) = (1/2) \ln \{ \tilde{\Delta}^0(t^*) / \Delta^0(t^*) \}, \quad (16)$$

$$\Delta I_\tau^t(t^*) = (1/2) \ln \{ \tilde{\Delta}_{11}^0(t^*) / \Delta_{11}^0(t^*) \}, \quad (17)$$

где $\Delta^0(t)$, $\tilde{\Delta}^0(t)$, $\Delta_{11}^0(t)$, $\tilde{\Delta}_{11}^0(t)$ определены в (4)–(7).

2) Для ΔI_{t_m} , $\Delta \tilde{I}_{t_m}$, $\Delta I_\tau^{t_m}$, $\Delta \tilde{I}_\tau^{t_m}$ справедливы формулы

$$\Delta I_{t_m} = \Delta \tilde{I}_{t_m} = (1/2) \ln \{ [1 + \delta_d] \}, \quad (18)$$

$$\Delta I_\tau^{t_m} = (1/2) \ln \{ [1 + \delta_d] [1 + \delta_d (1 - \rho(t^*))]^{-1} \}, \quad (19)$$

$$\Delta \tilde{I}_\tau^{t_m} = (1/2) \ln \{ [1 + \delta_d] [1 + \delta_d (1 - \tilde{\rho}(t^*))]^{-1} \}, \quad (20)$$

где $\rho(t)$, $\tilde{\rho}(t)$ определены в (14).

3) Пусть: $\Delta I_t(0) = \lim_{t \downarrow 0} \Delta I_t(t)$ и $\Delta I_\tau^t(0) = \lim_{t \downarrow 0} \Delta I_\tau^t(t)$ при $t \downarrow 0$; $\Delta I_t(\infty) = \lim_{t \uparrow \infty} \Delta I_t(t)$ и $\Delta I_\tau^t(\infty) = \lim_{t \uparrow \infty} \Delta I_\tau^t(t)$ при $t \uparrow \infty$; $\Delta I_\tau^{t_m}(0) = \lim_{t \downarrow 0} \Delta I_\tau^{t_m}(t)$ и $\Delta \tilde{I}_\tau^{t_m}(0) = \lim_{t \downarrow 0} \Delta \tilde{I}_\tau^{t_m}(t)$ при $t \downarrow 0$; $\Delta I_\tau^{t_m}(\infty) = \lim_{t \uparrow \infty} \Delta I_\tau^{t_m}(t)$ и $\Delta \tilde{I}_\tau^{t_m}(\infty) = \lim_{t \uparrow \infty} \Delta \tilde{I}_\tau^{t_m}(t)$ при $t \uparrow \infty$. Тогда $\Delta I_t(t) \geq 0$ является функцией t , монотонно возрастающей от значения $\Delta I_t(0) = 0$ до значения $\Delta I_t(\infty) = (1/2) \ln \{ 1 + (\delta_c/2a) \}$, а $\Delta I_\tau^t(t) \leq 0$ является функцией t , монотонно убывающей от значения $\Delta I_\tau^t(0) = 0$ до значения $\Delta I_\tau^t(\infty) = -\infty$; $\Delta I_\tau^{t_m}(t) \geq 0$ и $\Delta \tilde{I}_\tau^{t_m}(t) \geq 0$ является функцией t , монотонно убывающей от значения $\Delta I_\tau^{t_m}(0) = \Delta \tilde{I}_\tau^{t_m}(0) = (1/2) \ln \{ 1 + \delta_d \}$ до значения $\Delta I_\tau^{t_m}(\infty) = \Delta \tilde{I}_\tau^{t_m}(\infty) = 0$, и $\Delta I_\tau^{t_m}(t) < \Delta \tilde{I}_\tau^{t_m}(t)$ при конечных значениях $t > 0$.

Доказательство:

Из (46) в [2] имеем

$$I_t[\cdot] = (1/2) \ln [D(t) / \Delta^0(t)],$$

$$\tilde{I}_t[\cdot] = (1/2) \ln [D(t) / \tilde{\Delta}^0(t)].$$

Аналогично

$$I_\tau^t[\cdot] = (1/2) \ln [D(\tau) / \Delta_{11}^0(\tau, t)],$$

$$\tilde{I}_\tau^t[\cdot] = (1/2) \ln [D(\tau) / \tilde{\Delta}_{11}^0(\tau, t)].$$

Тогда формулы (16), (17) очевидны, а (18)–(20) следуют из (14), (33) из [4] и (15). Свойства 3) следуют из (16)–(20), (4)–(7), (14) с учетом того, что $\rho(t) < \tilde{\rho}(t)$ при конечных $t > 0$.

Заключение

1. Свойства $\varepsilon_i(t) < 1$, $\Delta I_i(t) > 0$ и $\varepsilon_i(t) > 1$, $\Delta I_i(t) < 0$ означают, что непрерывный канал передачи без памяти более эффективен, чем непрерывный канал передачи с запаздыванием при фильтрационном приеме, и имеет место обратные свойства при интерполяционном приеме. Эти свойства объясняются тем, что при фильтрационном приеме восстанавливается текущее значение x_t процесса x_σ , содержащееся в текущем наблюдении z_t в случае каналов без памяти, а при интерполяционном приеме восстанавливается прошлое значение x_t процесса x_σ , содержащееся в текущем наблюдении z_t в случае каналов с запаздыванием.
2. Свойства (12) и (18) означают, что при фильтрационном приеме эффективности дискретных каналов без памяти при непрерывных каналах без памяти и с запаздыванием совпадают. Эти свойства объясняются тем, что при фильтрационном приеме в момент времени t_m в дискретном канале восстанавливается значение x_{t_m} процесса x_σ , которое содержится в текущем наблюдении $\eta(t_m)$, и следовательно имеет место пересчет среднеквадратических ошибок $\Delta^0(t_m - 0)$ и $\tilde{\Delta}^0(t_m - 0)$ в $\tilde{\Delta}^0(t_m)$ и $\tilde{\Delta}^0(t_m)$ с единым коэффициентом, определяемым свойствами только дискретного канала (см. (15) для $\Delta^0(t_m)$ и $\tilde{\Delta}^0(t_m)$).
3. Свойства $\varepsilon_i(t_m; t) > \tilde{\varepsilon}_i(t_m; t)$ и $\Delta I_\tau^{t_m}(t) > \Delta \tilde{I}_\tau^{t_m}(t)$ для $t > 0$ означают, что при интерполяционном приеме эффективность дискретных каналов без памяти выше при непрерывных каналах с

запаздыванием, чем при непрерывных каналах без памяти. Это Свойство объясняется тем, что при интерполяционном приеме в момент времени t_m воспроизводятся прошлые значения x_t процесса x_σ , которые не входят в текущие дискретные $\eta(t_m)$ и текущие непрерывные наблюдения z_m при непрерывном канале без памяти, и входят в текущие непрерывные наблюдения z_m при непрерывных наблюдениях с запаздыванием. Таким образом, имеет место пересчет величин $\Delta_{11}^0(\tau, t_m - 0)$ и $\tilde{\Delta}_{11}^0(\tau, t_m - 0)$ в $\Delta_{11}^0(\tau, t_m)$ и $\tilde{\Delta}_{11}^0(\tau, t_m)$ (см. (15) для $\Delta_{11}^0(\tau, t_m)$ и $\tilde{\Delta}_{11}^0(\tau, t_m)$) с раз-

личными коэффициентами, а более высокую эффективность непрерывных каналов с запаздыванием обеспечивают указанные свойства. Отсутствие непрерывной передачи соответствует случаю $\delta_c = 0$. Так как $\lim_{\delta_c \rightarrow 0} \rho(f) = \tilde{\rho}(f)$ при $\delta_c \downarrow 0$, тогда из (13), (19), (20) для случая отсутствия непрерывных каналов $\delta_c = 0$ имеют место свойства $\varepsilon_i(t_m; f) = \tilde{\varepsilon}_i(t_m; f)$ и $D\Delta I_t^{1n}(f) = \tilde{D}\Delta I_t^{1n}(f)$, физический смысл этих свойств очевиден.

Работа выполнена при поддержке ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009–2013, проект № 14.B37.21.0861.

СПИСОК ЛИТЕРАТУРЫ

1. Рожкова С.В. Оптимальная непрерывно-дискретная передача сигнала по каналам с памятью при наличии бесшумной обратной связи // Известия Томского политехнического университета. — 2012. — Т. 321. — № 5. — С. 6–9.
2. Рожкова С.В. Оптимальная непрерывно-дискретная передача сигнала по каналам с запаздыванием // Известия Томского политехнического университета. — 2012. — Т. 321. — № 5. — С. 10–13.
3. Рожкова С.В. Оптимальная передача сигнала по совокупности непрерывного и дискретного каналов с памятью и запаздыванием // Известия Томского политехнического университета. — 2013. — Т. 322. — № 2. — С. 8–10.
4. Демин Н.С., Рожкова С.В. О структуре количества информации в совместной задаче фильтрации и интерполяции по наблюдениям с памятью. Условно-гауссовский случай // Известия Томского политехнического университета. — 2004. — Т. 307. — № 4. — С. 6–10.
5. Демин Н.С., Рожкова С.В., Рожкова О.В. Обобщенная скользящая экстраполяция стохастических процессов по совокупности непрерывных и дискретных наблюдений с памятью // Известия РАН. Теория и системы управления. — 2000. — № 4. — С. 39–51.

Поступила 01.04.2013 г.

УДК 519.2

РАСПОЗНАВАНИЕ СОСТОЯНИЙ СТОХАСТИЧЕСКИХ СИСТЕМ С ФИКСИРОВАННОЙ ПАМЯТЬЮ

С.В. Рожкова

Томский политехнический университет
E-mail: rozhkova@tpu.ru

Рассматривается задача распознавания произвольного числа гипотез, когда ненаблюдаемый процесс является процессом с непрерывным временем, а наблюдаемый процесс представляет собой совокупность процессов с непрерывным и дискретным временем с фиксированной памятью произвольной кратностью.

Ключевые слова:

Стохастические системы, память, распознавание, отношение правдоподобия.

Key words:

Stochastic systems, memory, recognition, likelihood ratio.

1. Постановка задачи

На вероятностном пространстве $(\Omega, \mathcal{F}, \mathcal{F} = (\mathcal{F}_t)_{t \in [0, T]}, \mathbf{P})$ ненаблюдаемый n -мерный процесс x_t (полезный сигнал) и наблюдаемый l -мерный процесс z_t непрерывным временем определяются стохастическими дифференциальными уравнениями

$$dx_t = f(t, x_t, z, \theta)dt + \Phi_1(t, x_t, z, \theta)dw_t, \quad (1)$$

$$dz_t = h(t, x_t, x_{t_1}, \dots, x_{t_N}, z, \theta)dt + \Phi_2(t, z)dv_t, \quad (2)$$

а наблюдаемый q -мерный процесс $\eta(t_m)$ с дискретным временем имеет вид

$$\eta(t_m) = g(t_m, x_{t_m}, x_{t_1}, \dots, x_{t_N}, z, \theta) + \Phi_3(t_m, z, \theta)\xi(t_m), \quad m = 0, 1, \dots, \quad (3)$$

где $0 < \tau_N < \dots < \tau_1 < t_m \leq T$. Параметр θ является идентификатором типа гипотезы и может принимать значения из множества $\Omega_\theta = \{\theta_0, \theta_1, \dots, \theta_r\}$ с априорными вероятностями $p_0(\theta_j) = \mathbf{P}\{\theta = \theta_j\}$, $j = 0; r$. Предполагается: 1) процессы w_t и v_t являются стандартными винеровскими процессами размеров r_1 и r_2 и при всех $\theta \in \Omega_\theta$ коэффициенты уравнений (1), (2) удовлетворяют условиям [1], а $\theta \in \Omega_\theta$ непрерывна по всем аргументам; 2) $\xi(t_m)$ — стандартная белая гауссовская последовательность размера r_3 с $\mathbf{M}\{\xi(t_m)|\theta = \theta_j\} = 0$ и $\mathbf{M}\{\xi(t_m)\xi^T(t_m)|\theta = \theta_j\} = I_{r_3}$; 3) $x_0, w_t, v_t, \xi(t_m), \theta$ статистически независимы; 4) $f(\cdot), \Phi(\cdot), h(\cdot), \Phi_2(\cdot), g(\cdot), \Phi_3(\cdot)$ являются неупреждающими функционалами от реализаций соответственно $z = z_0^t$ и $z = z_0^{t_m}$; 5) $Q(\cdot) = \Phi_1^T(\cdot) > 0$, $R(\cdot) = \Phi_2(\cdot)\Phi_2^T(\cdot) > 0$, $V(\cdot) = \Phi_3(\cdot)\Phi_3^T(\cdot) > 0$ при всех

$\theta \in \Omega_\theta$; 6) заданы начальные плотности $p_0(x|\theta_j) = \partial \mathbf{P}\{x_0 \leq x | \theta = \theta_j\} / \partial x, j=0; r$.

Задача: по совокупности реализаций $z_0^t = \{z(\sigma): 0 \leq \sigma \leq t\}$ и $\eta_0^m = \{\eta(t_0), \eta(t_1), \dots, \eta(t_m)\}$ найти отношения правдоподобия $\Lambda_i(\theta; \theta_\alpha)$ в задаче распознавания гипотез $H_j\{\theta = \theta_j\}$ и $H_j\{\theta = \theta_j\}, j=0; r, \alpha=0; r$.

2. Основные результаты

Метод нахождения $\Lambda_i(\theta; \theta_\alpha)$ основан на формуле [2]

$$\Lambda_i(\theta_j : \theta_\alpha) = [p_0(\theta_\alpha) / p_0(\theta_j)] P_i(\theta_j : \theta_\alpha), \quad (4)$$

которая связывает отношение правдоподобия и отношение апостериорных вероятностей гипотез

$$P_i(\theta_j : \theta_\alpha) = p_i(\theta_j) / p_i(\theta_\alpha), \quad (5)$$

где

$$p_i(\theta_j) = \mathbf{P}\{\theta = \theta_j | z_0^t, \eta_0^m\}, \quad j = 0; r. \quad (6)$$

Теорема 1.

Апостериорные вероятности (6) гипотез $H_j\{\theta = \theta_j\}$ на интервалах $t_m \leq t < t_{m+1}$ определяются уравнениями

$$d_t p_i(\theta_j) = p_i(\theta_j) [h(t, z | \theta_j) - \overline{h(t, z)}]^T \times \\ \times R^{-1}(t, z) [dz_t - \overline{h(t, z)} dt] \quad (7)$$

с начальными условиями

$$p_{t_m}(\theta_j) = [C(\eta(t_m), z | \theta_j) / C(\eta(t_m), z)] p_{t_m-0}(\theta_j), \quad (8)$$

где

$$\overline{h(t, z | \theta_j)} = \mathbf{M}\{h(t, x_t, \tilde{x}_t^N, z, \theta) | \theta = \theta_j, z_0^t, \eta_0^m\}, \quad (9)$$

$$\overline{h(t, z)} = \mathbf{M}\{h(t, x_t, \tilde{x}_t^N, z, \theta) | z_0^t, \eta_0^m\} = \\ = \sum_{j=0}^r \overline{h(t, z | \theta_j)} p_i(\theta_j),$$

$$C(\eta(t_m), z | \theta_j) = \\ = \mathbf{M}\{C(\eta(t_m), z, x_{t_m}, \tilde{x}_t^N, \theta) | \theta = \theta_j, z_0^t, \eta_0^{m-1}\}, \quad (10)$$

$$C(\eta(t_m), z) = \mathbf{M}\{C(\eta(t_m), z, x_{t_m}, \tilde{x}_t^N, \theta) | z_0^t, \eta_0^{m-1}\} = \\ = \sum_{j=0}^r C(\eta(t_m), z | \theta_j) p_{t_m-0}(\theta_j), \quad (11)$$

$$C(\eta(t_m), z, x, \tilde{x}_N, \theta_j) = |V(t_m, z, \theta_j)|^{-1/2} \times \\ \times \exp \left\{ -\frac{1}{2} [\eta(t_m) - g(t_m, x, \tilde{x}_N, z, \theta_j)]^T \times \right. \\ \left. \times V(t_m, z, \theta_j) [\eta(t_m) - g(t_m, x, \tilde{x}_N, z, \theta_j)] \right\}, \quad (12)$$

а $p_{t_m-0}(\theta_j) = \lim_{t \uparrow t_m} p_i(\theta_j)$ при $t \uparrow t_m$.

Доказательство:

$$p_i(x; \tilde{x}_N; \theta_j) = \\ = \partial^{N+1} \mathbf{P}\{x_t \leq x, \tilde{x}_t^N \leq \tilde{x}_N, \theta = \theta_j | z_0^t, \eta_0^m\} / \partial x \partial \tilde{x}_N$$

на интервалах $t_m \leq t < t_{m+1}$ справедливо уравнение

$$d_t p_i(x; \tilde{x}_N; \theta_j) = L_{i,x}[p_i(x; \tilde{x}_N; \theta_j)] dt + \\ + p_i(x; \tilde{x}_N; \theta_j) [h(t, x, \tilde{x}_N, z, \theta_j) - \overline{h(t, z)}]^T \times \\ \times R^{-1}(t, z) [dz_t - \overline{h(t, z)} dt], \quad (13)$$

где $L_{i,x}[\cdot]$ – прямой оператор Колмогорова, соответствующий процессу x_t при $\theta = \theta_j$ [3]. Так как

$$p_i(x; \tilde{x}_N; \theta_j) = p_i(x; \tilde{x}_N | \theta_j) p_i(\theta_j), \quad (14)$$

где

$$p_i(x; \tilde{x}_N | \theta_j) = \\ = \partial^{N+1} \mathbf{P}\{x_t \leq x, \tilde{x}_t^N \leq \tilde{x}_N | \theta = \theta_j, z_0^t, \eta_0^m\} / \partial x \partial \tilde{x}_N, \quad (15)$$

то интегрирование (13) по $\{x, \tilde{x}_N\}$ с учетом (14) дает (7). По формуле Байеса

$$p_{t_m}(x; \tilde{x}_N; \theta_j) = \\ = \frac{p(\eta(t_m) | x_{t_m} = x, \tilde{x}_t^N = \tilde{x}_N, \theta = \theta_j, z_0^t, \eta_0^{m-1}) \times \\ \times p_{t_m-0}(x; \tilde{x}_N; \theta_j)}{\sum_{k=0}^N \left[\int p(\eta(t_m) | x_{t_m} = x, \tilde{x}_t^N = \tilde{x}_N, \theta = \theta_k, z_0^t, \eta_0^{m-1}) \times \right. \\ \left. \times p_{t_m-0}(x; \tilde{x}_N; \theta_k) dx d\tilde{x}_N \right]}, \quad (16)$$

где $p_{t_m-0}(x; \tilde{x}_N; \theta_j) = \lim_{t \uparrow t_m} p_i(x; \tilde{x}_N; \theta_j)$ при $t \uparrow t_m$. Из (3) с учетом Предположений 2), 3) следует

$$p(\eta(t_m) | x, \tilde{x}_N, \theta_j, z_0^t, \eta_0^{m-1}) = \\ = \mathbf{N}\{\eta(t_m); g(t_m, x, \tilde{x}_N, z, \theta_j), V(t_m, z, \theta_j)\}. \quad (17)$$

Использование (17) в (16) с учетом (11), (12) дает

$$p_{t_m}(x; \tilde{x}_N; \theta_j) = \\ = \frac{C(\eta(t_m), z, x, \tilde{x}_N, \theta_j)}{C(\eta(t_m), z)} p_{t_m-0}(x; \tilde{x}_N; \theta_j). \quad (18)$$

Интегрирование (18) по $\{x, \tilde{x}_N\}$ с учетом (10), (14) дает (8).

Следствие 1.

Для $p_i(\theta_j)$ справедлива формула

$$p_i(\theta_j) = p_{t_1-0}(\theta_j) \times \\ \times \exp \left\{ \sum_{t_1 \leq t \leq t} \ln \left[\frac{C(\eta(t), z | \theta_j)}{C(\eta(t), z)} \right] + \right. \\ \left. + \int_{t_1}^t [h(s, z | \theta_j) - \overline{h(s, z)}]^T \times \right. \\ \left. \times R^{-1}(s, z) \left[dz_s - \frac{1}{2} \overline{h(s, z | \theta_j)} ds - \frac{1}{2} \overline{h(s, z)} ds \right] \right\}. \quad (19)$$

Доказательство:

Пусть $p \sim_i(\theta_j) = \ln\{p_i(\theta_j)\}$. Дифференцирование по формуле Ито с учетом (7) дает для $t_m \leq t < t_{m+1}$

$$d_t \tilde{p}_i(\theta_j) = [h(t, z | \theta_j) - \overline{h(t, z)}]^T \times \\ \times R^{-1}(t, z) \left[dz_t - \frac{1}{2} \overline{h(t, z | \theta_j)} dt - \frac{1}{2} \overline{h(t, z)} dt \right]. \quad (20)$$

Из (8) следует

$$\tilde{p}_{t_m}(\theta_j) = \tilde{p}_{t_m-0}(\theta_j) + \ln \frac{C(\eta(t_m), z | \theta_j)}{C(\eta(t_m), z)}. \quad (21)$$

Так как $p_i(\theta_j) = \exp\{\tilde{p}_i(\theta_j)\}$, то (19) для $t \geq \tau_i$ следует из (20), (21).

Теорема 2.

Отношение правдоподобия $\Lambda_i(\theta_j; \theta_\alpha)$ в задаче распознавания гипотез $H_j = \{\theta = \theta_j\}$ и $H_\alpha = \{\theta = \theta_\alpha\}$, $j=0;r$, $\alpha=0;r$ на интервалах $t_m \leq t < t_{m+1}$ определяется уравнением

$$\begin{aligned} d_t \Lambda_i(\theta_j; \theta_\alpha) &= \\ &= \Lambda_i(\theta_j; \theta_\alpha) [\overline{h(t, z | \theta_j)} - \overline{h(t, z | \theta_\alpha)}]^T \times \\ &\times R^{-1}(t, z) [dz_t - \overline{h(t, z | \theta_\alpha)} dt], \end{aligned} \quad (22)$$

с начальным условием

$$\begin{aligned} \Lambda_{t_m}(\theta_j; \theta_\alpha) &= \\ &= [C(\eta(t_m), z | \theta_j) / C(\eta(t_m), z | \theta_\alpha)] \Lambda_{t_m-0}(\theta_j; \theta_\alpha), \end{aligned} \quad (23)$$

где $\Lambda_{t_m-0}(\theta_j; \theta_\alpha) = \lim_{t \uparrow t_m} \Lambda_i(\theta_j; \theta_\alpha)$ при $t \uparrow t_m$.

Доказательство:

Дифференцирование (5) по формуле Ито с учетом (7) дает уравнение

$$\begin{aligned} d_t P_i(\theta_j; \theta_\alpha) &= P_i(\theta_j; \theta_\alpha) [\overline{h(t, z | \theta_j)} - \overline{h(t, z | \theta_\alpha)}]^T \times \\ &\times R^{-1}(t, z) [dz_t - \overline{h(t, z | \theta_\alpha)} dt]. \end{aligned} \quad (24)$$

Использование (4) в (24) приводит к (22). Использование (8) в (5) с последующим использованием (4) дает (23).

Следствие 2.

Для $\Lambda_i(\theta_j; \theta_\alpha)$ справедлива формула

$$\begin{aligned} \Lambda_i(\theta_j; \theta_\alpha) &= \Lambda_{\tau_i-0}(\theta_j; \theta_\alpha) \times \\ &\times \exp \left\{ \sum_{\tau_i \leq t_j \leq t} \ln \left[\frac{C(\eta(t_j), z | \theta_j)}{C(\eta(t_j), z | \theta_\alpha)} \right] + \right. \\ &\left. \int_{\tau_i}^t [\overline{h(s, z | \theta_j)} - \overline{h(s, z | \theta_\alpha)}]^T \times \right. \\ &\times R^{-1}(s, z) \left[dz_s - \frac{1}{2} \overline{h(s, z | \theta_j)} ds - \frac{1}{2} \overline{h(s, z | \theta_\alpha)} ds \right] \Bigg\}. \end{aligned} \quad (25)$$

Доказательство:

Представим уравнение (22) в виде

$$\begin{aligned} d_t \Lambda_i(\theta_j; \theta_\alpha) &= \Lambda_i(\theta_j; \theta_\alpha) [\overline{h(t, z | \theta_j)} - \overline{h(t, z | \theta_\alpha)}]^T \times \\ &\times R^{-1}(t, z) [\overline{h(t, z)} - \overline{h(t, z | \theta_\alpha)}] dt + \\ &+ \Lambda_i(\theta_j; \theta_\alpha) [\overline{h(t, z | \theta_j)} - \overline{h(t, z | \theta_\alpha)}]^T \times \\ &\times R^{-1}(t, z) [dz_t - \overline{h(t, z)} dt]. \end{aligned} \quad (26)$$

Пусть

$$\tilde{\Lambda}_i(\theta_j; \theta_\alpha) = \ln \{\Lambda_i(\theta_j; \theta_\alpha)\}. \quad (27)$$

Дифференцирование (27) по формуле Ито с учетом (26) дает для $t_m \leq t < t_{m+1}$

$$\begin{aligned} d_t \tilde{\Lambda}_i(\theta_j; \theta_\alpha) &= [\overline{h(t, z | \theta_j)} - \overline{h(t, z | \theta_\alpha)}]^T \times \\ &\times R^{-1}(t, z) \left[dz_t - \frac{1}{2} \overline{h(t, z | \theta_j)} dt - \frac{1}{2} \overline{h(t, z | \theta_\alpha)} dt \right]. \end{aligned} \quad (28)$$

Из (23) следует

$$\tilde{\Lambda}_i(\theta_j; \theta_\alpha) = \tilde{\Lambda}_{t_m-0}(\theta_j; \theta_\alpha) + \ln \frac{C(\eta(t_m), z | \theta_j)}{C(\eta(t_m), z | \theta_\alpha)}. \quad (29)$$

Так как $\Lambda_i(\theta_j; \theta_\alpha) = \exp\{\tilde{\Lambda}_i(\theta_j; \theta_\alpha)\}$, то (25) для $t \geq \tau_i$ следует из (28), (29).

Заключение

Из Теорем 1, 2 следует, что эффективное вычисление $p_i(\theta_j)$ и $\Lambda_i(\theta_j; \theta_\alpha)$ реализуется при возможности эффективного вычисления $\overline{h(t, z | \theta_j)}$ и $C(\eta(t_m), z | \theta_j)$, которые согласно (9), (10), (15) определяются формулами

$$\begin{aligned} \overline{h(t, z | \theta_j)} &= \int \dots \int h(t, x, \tilde{x}_N, z, \theta_j) p_i(x; \tilde{x}_N | \theta_j) dx d\tilde{x}_N, \\ C(\eta(t_m), z | \theta_j) &= \\ &= \int \dots \int C(\eta(t_m), z, x, \tilde{x}_N, \theta_j) p_{t_m-0}(x; \tilde{x}_N | \theta_j) dx d\tilde{x}_N, \end{aligned}$$

где условные апостериорные плотности $p_i(x; \tilde{x}_N | \theta_j)$ на интервалах $t_m \leq t < t_{m+1}$ определяются уравнениями

$$\begin{aligned} d_t p_i(x; \tilde{x}_N | \theta_j) &= L_{t,x} [p_i(x; \tilde{x}_N | \theta_j)] dt + \\ &+ p_i(x; \tilde{x}_N | \theta_j) [\overline{h(t, x, \tilde{x}_N, z, \theta_j)} - \overline{h(t, z | \theta_j)}]^T \times \\ &\times R^{-1}(t, z) [dz_t - \overline{h(t, z | \theta_j)} dt] \end{aligned} \quad (30)$$

с начальными условиями

$$\begin{aligned} p_{t_m}(x; \tilde{x}_N | \theta_j) &= \\ &= \frac{C(\eta(t_m), z, x, \tilde{x}_N, \theta_j)}{C(\eta(t_m), z | \theta_j)} p_{t_m-0}(x; \tilde{x}_N | \theta_j). \end{aligned} \quad (31)$$

Уравнение (30) следует в результате дифференцирования по формуле Ито соотношения

$$p_i(x; \tilde{x}_N | \theta_j) = p_i(x; \tilde{x}_N; \theta_j) / p_i(\theta_j). \quad (32)$$

С учетом (15) и (13), а (31) следует из (8), (20), (32).

Работа выполнена при поддержке ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009–2013 гг., проект № 14.B37.21.0861.

СПИСОК ЛИТЕРАТУРЫ

1. Липшер Р.Ш., Ширяев А.Н. Статистика случайных процессов. – М.: Наука, 1974. – 696 с.
2. Middleton D. Introduction to statistical communication theory. – New York: Mc Graw-Hill, 1960. – 1140 p.

3. Демин Н.С., Рожкова С.В., Рожкова О.В. Обобщенная скользящая экстраполяция стохастических процессов по совокупности непрерывных и дискретных наблюдений с памятью // Известия РАН. Теория и системы управления. – 2000. – № 4. – С. 39–51.

Поступила 04.04.2013 г.

ПРОИЗВОДНЫЕ И ИНТЕГРАЛЫ ДРОБНЫХ КОМПЛЕКСНЫХ ПОРЯДКОВ ФУНКЦИЙ ДИСКРЕТНОЙ ПЕРЕМЕННОЙ

В.А. Чуриков

Томский политехнический университет

E-mail: vachurikov@list.ru

Вводится дискретный d -оператор дробного интегродифференцирования комплексных порядков. Рассматривается алгоритм дискретного дифференцирования и дискретного интегрирования функций дискретной переменной.

Ключевые слова:

d -оператор дискретной переменной, дискретная производная, дискретный интеграл.

Key words:

d -operator of discrete variable, discrete derivative, discrete integral.

Как известно, такие аналитические операции, как дифференцирование и интегрирование, применяются к функциям, которые обладают рядом необходимых свойств. Например, дифференцируемая функции в точке должна быть непрерывной в данной точке, а интегрируемая функции (по Риману) на отрезке должна быть кусочно непрерывной на данном отрезке [1].

Формально операции дифференцирования и интегрирования (интегродифференцирования) можно определить и для последовательностей — функций $f(n)$ дискретной переменной n , пробегающей ряд натуральных чисел. Особенностью этих операций является то, что они должны определяться на нигде не плотном множестве. Эти операции назовём *дискретной производной* и *дискретным интегралом* порядка s , и будем обозначать

$$d^{-s}n : f(n) \equiv \frac{d^s}{dn^s} f(n) \equiv \left(\frac{d}{dn} \right)^s f(n);$$

$$d^s n : f(n) \equiv \int f(n) d^s n,$$

где символами $d^{-s}n$ и $d^s n$ обозначены, соответственно, операторы дифференцирования и интегрирования порядка s , по дискретной переменной n .

Прежде чем определить данные операции, сформулируем условия, которым они должны удовлетворять.

Потребуем, чтобы функции дискретной переменной $f(n)$ переходили в функции непрерывной переменной $f(x)$, а операторы дискретного дифференцирования $d^{-s}n$ и дискретного интегрирования $d^s n$ переходили в соответствующие непрерывные операторы дифференцирования $d^{-s}x$ и интегрирования $d^s x$, при замене дискретной переменной n непрерывной переменной x . Такой переход назовём *прямым переходом*, а переход при замене непрерывной переменной на дискретную в функциях и операторах — *обратным переходом*. Прямой и обратный переходы должны всегда выполняться и приводить к однозначным результатам для всех значений n из области определения функции $f(n)$. Порядок интегродифференцирования s является комплексным и не меняется при прямом и обратном переходах

$$\begin{aligned} f(n) &\xleftrightarrow{n \leftrightarrow x} f(x) \Leftrightarrow f(n) = f(x)|_{n=x}; \\ d^{\pm s} n &\xleftrightarrow{n \leftrightarrow x} d^{\pm s} x \Leftrightarrow d^{\pm s} n = d^{\pm s} x|_{n=x}. \end{aligned} \quad (1)$$

Из этого следует выполнение равенств, для производных и интегралов при тех же преобразованиях

$$\begin{aligned} d^{\pm s} n : f(n) &\xleftrightarrow{n \leftrightarrow x} d^{\pm s} x : f(x) \Leftrightarrow d^{\pm s} n : f(n) = \\ &= d^{\pm s} x : f(x)|_{n=x}. \end{aligned}$$

Дополнительно потребуем, чтобы при переходе функций дискретной переменной к функциям непрерывной переменной, от которых берётся производная и/или интеграл порядка 1, результат совпадал с аналогичным результатом, если эту функцию продифференцировать и/или проинтегрировать в стандартном анализе. Выполнение этого условия является следствием *принципа соответствия*.

В данной работе введём дискретную производную и дискретный неопределённый интеграл на основе локального d -оператора дробного интегродифференцирования. Этот оператор носит алгебраический характер, что и позволяет применить его к функциям дискретной переменной. В данном случае переопределим d -оператор как оператор дискретной переменной n , действующий в *пространстве степенных функций дискретной переменной* n^q . Это сужает возможности ранее введённого d -оператора [2, 3].

В рассматриваемом d -операторе порядки интегродифференцирования и показатели степенных функций, на которые действует d -оператор, являются комплексными числами, что в данном случае расширяет область применения d -оператора.

Определение. *Дискретным d -оператором*, или оператором *дробного дифференцирования и дробного интегрирования дискретной переменной* $n \in \mathbb{Z}$ комплексного порядка $s = \chi + i\gamma$, $\chi, \gamma \in \mathbb{R}$; $\chi, \gamma = \text{const}$; $\chi, \gamma \geq 0$, действующим в пространстве степенных функций дискретной переменной n с комплексными показателями $q = \mu + i\nu$; $\mu, \nu \in \mathbb{R}$; $\mu, \nu = \text{const}$, будем называть равенства

$$\left\{ \begin{array}{l} d^{-s}n : n^q = \frac{\Gamma(q+1)}{\Gamma(q-s+1)} n^{q-s}; q \neq -1, -2, -3, \dots; \\ d^s n : n^q = \frac{\Gamma(q+1)}{\Gamma(q+s+1)} n^{q+s} + C_s(n); \\ \left\{ \begin{array}{l} q \neq -1, -2, -3, \dots; s \notin \mathbb{N}; \\ q = -1, -2, -3, \dots; s \in \mathbb{N}; s < |q|; \end{array} \right. \\ d^{-s}n : n^{-m} = \frac{(-1)^{m-1}}{(m-1)!\Gamma(-s-m+1)} n^{-m-s}; \\ m \in \mathbb{N}; s \neq 0, 1, 2, 3, \dots; \\ d^s n : n^{-m} = \frac{(-1)^{m-1}}{(m-1)!\Gamma(s-m+1)} n^{-m+s} + C_s(n); \\ m \in \mathbb{N}; s \neq 0, 1, 2, 3, \dots; \\ d^1 n : n^{-1} = \ln |n| + C_1; C_1 = \text{const}. \end{array} \right. \quad (2)$$

Рассмотрим частные случаи возможных порядков интегриродифференцирования.

Когда порядок $s=\chi=\gamma=0$, это единичный оператор, переводящий функции самих в себя.

Когда порядок интегриродифференцирования вещественный, $s=\text{Re}(s)=\chi>0$, а в равенствах (2) перед показателем порядка оператора s стоит знак минус, это будет соответствовать оператору *дробного дифференцирования вещественного порядка* χ , а если значение порядка оператора со знаком плюс, то это будет оператор *дробного интегрирования вещественного порядка* χ .

Когда порядок интегриродифференцирования мнимый, $s=i\gamma$, $\gamma>0$, и если в равенствах (2) перед показателем порядка оператора s , стоит знак минус, то это будет оператор *дробного дифференцирования мнимого порядка* χ , а если значение мнимого порядка оператора со знаком плюс, то это оператор *дробного интегрирования мнимого порядка* χ .

Если порядок интегриродифференцирования комплексный, $s=\chi+i\gamma$, $\chi, \gamma>0$, а знак у порядка отрицательный, то это будет *дробное дифференцирование комплексного порядка*, а если знак положительный, то это соответствует *дробному интегрированию комплексного порядка*.

Если знаки у вещественной и мнимой части порядка интегриродифференцирования различаются, т. е. $s=-\chi+i\gamma$ или $s=\chi-i\gamma$, то такие порядки будем называть *смешанными комплексными порядками*. В этом случае нельзя говорить только о дифференцировании или только об интегрировании. Если в операторе у смешанного порядка перед вещественной частью стоит знак минус, то формально будем говорить, что это оператор *смешанного дифференцирования комплексного порядка* s , а если знак плюс, то это оператор *смешанного интегрирования комплексного порядка* s .

Первое равенство в операторе (2) определяет дробное дифференцирование порядка s . Дополнительные условия исключают случаи дифференцирования в полюсах гамма-функции $\Gamma(\dots)$ в числителе коэффициента оператора. Полюса у гамма-

функции имеются для целочисленных вещественных порядков $s=\text{Re}(s)=\chi=0, -1, -2, -3, \dots$

Второе равенство определяет дробное интегрирование порядка $s \geq 0$, когда значение гамма-функции в числителе не попадает в полюс. Первые дополнительные условия в данном равенстве исключают случаи интегрирования в полюсах. Вторые дополнительные условия исключают интегрирование в логарифмических случаях.

Третье и четвертое равенства определяют дифференцирование и интегрирование в полюсах, когда показатели степени степенных функций вещественные и имеют отрицательные целочисленные значения.

Пятое равенство определяет интегрирование в логарифмическом случае, которое соответствует такому случаю в стандартном анализе.

Все указанные дополнительные условия в операторе, налагаемые на d -оператор, лежат в вещественной области.

Далее $C_s(n)$ и C_1 — полиномы интегрирования порядков s и 1 соответственно, которые являются обобщениями констант интегрирования стандартного анализа. Производная порядка s от полинома интегрирования порядка s равна нулю.

При дискретном дифференцировании порядка s полиномов интегрирования порядка s получим ноль

$$d^{-s}n : C_s(n) = 0.$$

Полиномы интегрирования дискретного аргумента определяются как в случае одномерного дробного анализа [2, 3]

$$C_s(n) = \begin{cases} C_0(n) = 0; s = 0; \\ C_\alpha(n) = \sum_{k=1}^{\infty} a_k n^{-k+\alpha}; s = \alpha; \\ \alpha = \chi + i\gamma \wedge s = \pm \chi \mp i\gamma; \\ a_k \in \mathbb{C}; a_k = \text{const}; |\alpha| > 0; \alpha \neq 1, 2, 3, \dots; \\ C_m(n) = \sum_{k=0}^{m-1} a_k n^k; a_k = \text{const}; s = m; m \in \mathbb{N}. \end{cases} \quad (3)$$

Здесь неопределённые коэффициенты a_k являются константами интегрирования, которых будет k в случае целочисленных порядков и бесконечное счётное множество для нецелочисленных порядков.

При дискретном интегрировании порядка s , по дискретной переменной n , функции дискретной переменной $f(n)$ получим

$$d^s n : f(n) = F^{(s)}(n) + C_s(n) \equiv f^{(-s)}(n) + C_s(n).$$

Здесь $F^{(s)}(n) \equiv F^{(-s)}(n)$ — базовая первообразная порядка s функции $f(n)$, т. е. такая первообразная, у которой полином интегрирования равен нулю [4] и $C_s(n)$ — полином интегрирования порядка s .

При прямом переходе, $n \rightarrow x$, дискретный оператор (2) и полином интегрирования (3) переходят в непрерывный d -оператор и полином интегриро-

вания, зависящие от непрерывной переменной. Порядки дифференцирования и интегрирования в этом операторе в общем случае являются комплексными. Действует такой оператор на степенные функции x^q , показатели степени q у которых в общем случае тоже комплексные.

Дискретный d -оператор является линейным

$$d^{\pm s} n : (\eta f(n) + \lambda g(n)) = \eta d^{\pm s} n : f(n) + \lambda d^{\pm s} n : g(n).$$

Здесь $f(n), g(n)$ – функции дискретной переменной; $\eta, \lambda \in \mathbb{C}$; $\eta, \lambda = \text{const}$.

Для непрерывного d -оператора справедлив принцип соответствия [5], из которого следует, что он выполняется и для дискретного d -оператора ввиду того, что дискретный d -оператор является частным случаем непрерывного d -оператора.

Нахождение дискретной производной и дискретного интеграла от функций дискретной переменной, которые можно представить в виде суперпозиции степенных функций от дискретной переменной, производится, используя оператор (2).

В силу преобразований (1) возможен и более сложный, но более универсальный, алгоритм нахождения дискретной производной и дискретного интеграла от функции дискретной переменной, который заключается в последовательности действий:

- 1) замена в функции дискретной переменной самой дискретной переменной на непрерывную и получение функции непрерывной переменной (прямой переход);
- 2) представление функции непрерывной переменной в виде конечной линейной комбинации степенных функций или в виде степенного ряда;
- 3) интегриродифференцирование полученной функции непрерывным d -оператором;
- 4) обратный переход, т. е. замена в производной (первообразной) функции непрерывной переменной на функции дискретной переменной.

Второй способ интегриродифференцирования необходимо использовать, когда функция дискретного аргумента не представлена в виде суперпозиции степенных функций дискретного аргумента.

Производную дискретной переменной, можно интерпретировать как скорость изменения после-

довательности при изменении дискретного аргумента.

Рассмотрим пример дифференцирования комплексного порядка $\beta = a + ib$; $a, b \in \mathbb{R}$; $a, b = \text{const}$ ря-

да арифметической прогрессии $\sum_{n=1}^{\infty} n$. Непосредственное дифференцирование даёт

$$d^{-\beta} n : \sum_{n=1}^{\infty} n = \frac{\Gamma(1+1)}{\Gamma(1-\beta+1)} \sum_{n=1}^{\infty} n^{1-\beta} = \frac{1}{\Gamma(1+s)} \sum_{n=1}^{\infty} n^{-s}.$$

Здесь $1-\beta = 1-a-ib = -s$.

Если $\text{Re}(s) = a-1 > 1$, то полученную производную можно выразить через дзета-функцию Римана

$$\zeta(s) = \sum_{n=1}^{\infty} n^{-s} \quad [6]$$

$$d^{-a} n : \sum_{n=1}^{\infty} n = \frac{1}{\Gamma(1+s)} \sum_{n=1}^{\infty} n^{-s} = \frac{1}{\Gamma(1+s)} \zeta(s).$$

Пример интегрирования комплексного порядка $\rho = a + ib$ гармонического ряда $\sum_{n=1}^{\infty} n^{-1}$

$$\begin{aligned} d^{\rho} n : \sum_{n=1}^{\infty} n^{-1} + C_{\rho}(n) &= \\ &= \frac{(-1)^0}{0!} \frac{1}{\Gamma(-1+\rho+1)} \sum_{n=1}^{\infty} n^{\rho-1} + C_{\rho}(n) = \\ &= \frac{1}{\Gamma(1-s)} \sum_{n=1}^{\infty} n^{-s} + C_{\rho}(n). \end{aligned}$$

Здесь $\rho-1 = -s = a-1+ib$.

Если $\text{Re}(s) = a-1 > 1$, то полученный неопределённый интеграл можно выразить через дзета-функцию Римана

$$d^{\rho} n : \sum_{n=1}^{\infty} n^{-1} + C_{\rho}(n) = \frac{1}{\Gamma(1-s)} \zeta(s) + C_{\rho}(n).$$

Последовательности играют важную роль для строгого определения пределов в анализе. Дискретный d -оператор позволяет использовать дифференцирование и интегрирование для более глубокого изучения самих последовательностей.

СПИСОК ЛИТЕРАТУРЫ

1. Ильин В.А., Садовничий В.А., Сендов Бл.Х. Математический анализ. Начальный курс / под ред. А.Н. Тихонова. 2-е изд., перераб. – М.: МГУ, 1985. – 662 с.
2. Чуриков В.А. Локальный d -оператор дифференцирования и интегрирования конечных вещественных порядков для дробного анализа // Известия Томского политехнического университета. – 2011. – Т. 318. – № 2. – С. 5–10.
3. Чуриков В.А. Краткое введение в дробный анализ целочисленных порядков. – Томск: Изд-во ТПУ, 2011. – 72 с.
4. Чуриков В.А. Дополнительные главы анализа. Дробное интегрирование и дробное дифференцирование на основе d -оператора. – Томск: Изд-во Томского политехнического университета, 2010. – 118 с.
5. Чуриков В.А. Доказательство принципа соответствия в дробном анализе на основе d -оператора // Математическое моделирование фрактальных процессов, родственные проблемы анализа и информатики: Матер. Междунар. конф. молодых ученых. – Нальчик, 5–8 декабря, 2011. – Нальчик: Изд-во КБНЦ РАН, 2011. – С. 237–239.
6. Воронин С.М., Карацуба А.А. Дзета-функция Римана. – М.: Физматлит, 1994. – 376 с.

Поступила 16.07.2012 г.

ВЫДЕЛЕНИЕ ТРЕНДА ДИСПЕРСИИ СЛУЧАЙНОГО ПРОЦЕССА ДЛЯ ПУАССОНОВСКОГО ПОТОКА МОМЕНТОВ ИЗМЕРЕНИЙ

И.Г. Устинова

Томский политехнический университет
E-mail: igu@sibmail.com

Получены оценки тренда дисперсии в явном виде при пуассоновском потоке моментов измерений. Рассмотрены частные случаи тренда: линейный, квадратичный и в виде сплайна первого порядка. Исследованы статистические характеристики полученных оценок. Показана их несмещенность, найдена ковариационная матрица оценок параметров тренда.

Ключевые слова:

Тренд дисперсии, сплайн первого порядка, пуассоновский поток, оценки параметров, статистические свойства оценок.

Key words:

Dispersion trend, first-order spline, Poisson current, parameter estimations, estimations statistical properties.

Целью работы является нахождение оценок параметров тренда дисперсии случайного процесса, когда моменты измерений образуют пуассоновский поток постоянной интенсивности λ , и исследование их статистических характеристик.

Постановка задачи

Рассмотрим случайный процесс $x(t)$. Обозначим $x(t_i) = x_i$, $i = \overline{1, N}$ независимые случайные величины, распределенные нормально с $M[x_i] = 0$ и $D[x_i] = D[t_i]$. Моменты измерений t_i , $i = \overline{1, N}$ образуют пуассоновский поток постоянной интенсивности λ . В качестве модели неизвестной дисперсии предлагается

использовать $D[t_i] = \sum_{s=0}^k \theta_s \varphi_s(t_i)$, где $\varphi_s(t_i)$ – некоторые известные функции времени, а θ_s – неизвестные параметры. Оценки неизвестных параметров

$\theta_s, s = \overline{0, k}$ будем искать в виде $\hat{\theta}_s = \sum_{i=1}^N x_i^2(t_i) \psi_s(t_i)$, где

функции $\varphi_s(t_i)$, $s = \overline{0, k}$ выбираются таким образом, чтобы выполнялось условие: $M[\hat{\theta}_s] = \theta_s$.

Оценки неизвестных параметров тренда

Найдем оценки $\hat{\theta}_s$, $s = \overline{0, k}$, исходя из условия:

$$Q = \sum_{i=1}^N \left\{ x_i^2 - \sum_{s=0}^k \hat{\theta}_s \varphi_s(t_i) \right\}^2 \Rightarrow \min_{\hat{\theta}_s}$$

методом наименьших квадратов [1]. Из условия

$\frac{\partial Q}{\partial \hat{\theta}_s} = 0$ получаем систему $k+1$ линейных уравнений с $k+1$ неизвестными:

$$\begin{cases} \frac{\partial Q}{\partial \hat{\theta}_0} = \sum_{i=1}^N \left\{ x_i^2 - \sum_{s=0}^k \hat{\theta}_s \varphi_s(t_i) \right\} \varphi_0(t_i) = 0; \\ \frac{\partial Q}{\partial \hat{\theta}_1} = \sum_{i=1}^N \left\{ x_i^2 - \sum_{s=0}^k \hat{\theta}_s \varphi_s(t_i) \right\} \varphi_1(t_i) = 0; \\ \dots \\ \frac{\partial Q}{\partial \hat{\theta}_k} = \sum_{i=1}^N \left\{ x_i^2 - \sum_{s=0}^k \hat{\theta}_s \varphi_s(t_i) \right\} \varphi_k(t_i) = 0. \end{cases} \quad (1)$$

Перейдем к матричным обозначениям:

$$\Phi = \begin{bmatrix} \varphi_0(t_1) & \varphi_1(t_1) & \dots & \varphi_k(t_1) \\ \varphi_0(t_2) & \varphi_1(t_2) & \dots & \varphi_k(t_2) \\ \dots & \dots & \dots & \dots \\ \varphi_0(t_N) & \varphi_1(t_N) & \dots & \varphi_k(t_N) \end{bmatrix},$$

$$Y = \begin{bmatrix} \sum_{i=1}^N x_i^2(t_i) \varphi_0(t_i) \\ \sum_{i=1}^N x_i^2(t_i) \varphi_1(t_i) \\ \dots \\ \sum_{i=1}^N x_i^2(t_i) \varphi_k(t_i) \end{bmatrix},$$

$$\hat{\theta} = [\hat{\theta}_0 \quad \hat{\theta}_1 \quad \dots \quad \hat{\theta}_k]^T.$$

В матричных обозначениях система (1) будет иметь вид: $\Phi^T \Phi \hat{\theta} = Y$. Обозначим $\Phi^T \Phi$ через X , тогда

$$X \hat{\theta} = Y. \quad (2)$$

Заметим, что матрица

$$X = \begin{bmatrix} \sum_{i=1}^N \varphi_0^2(t_i) & \sum_{i=1}^N \varphi_0(t_i) \varphi_1(t_i) & \dots & \sum_{i=1}^N \varphi_0(t_i) \varphi_k(t_i) \\ \sum_{i=1}^N \varphi_0(t_i) \varphi_1(t_i) & \sum_{i=1}^N \varphi_1^2(t_i) & \dots & \sum_{i=1}^N \varphi_1(t_i) \varphi_k(t_i) \\ \dots & \dots & \dots & \dots \\ \sum_{i=1}^N \varphi_0(t_i) \varphi_k(t_i) & \sum_{i=1}^N \varphi_1(t_i) \varphi_k(t_i) & \dots & \sum_{i=1}^N \varphi_k^2(t_i) \end{bmatrix}$$

есть матрица со случайными элементами. Нахождение обратной матрицы – сложная задача, а нахождение обратной матрицы к случайной – практически неразрешимая. Поэтому для нахождения оценок неизвестных параметров θ_s , $s = \overline{0, k}$ перейдем от матрицы X к матрице $\bar{X}$, усреднив матрицу X по моментам измерений $\{t_i, i = \overline{1, N}\}$, применяя технику усреднения [2]

$$\bar{X} = \lambda \begin{bmatrix} \int_0^T \varphi_0^2(u) du & \int_0^T \varphi_0(u) \varphi_1(u) du & \dots & \int_0^T \varphi_0(u) \varphi_k(u) du \\ \int_0^T \varphi_0(u) \varphi_1(u) du & \int_0^T \varphi_1^2(u) du & \dots & \int_0^T \varphi_1(u) \varphi_k(u) du \\ \dots & \dots & \dots & \dots \\ \int_0^T \varphi_0(u) \varphi_k(u) du & \int_0^T \varphi_1(u) \varphi_k(u) du & \dots & \int_0^T \varphi_k^2(u) du \end{bmatrix},$$

Обозначим

$$\bar{\mathbf{X}} = \lambda \mathbf{X}_0, \quad (3)$$

где

$$\mathbf{X}_0 = \begin{bmatrix} \int_0^T \varphi_0^2(u) du & \int_0^T \varphi_0(u) \varphi_1(u) du & \dots & \int_0^T \varphi_0(u) \varphi_k(u) du \\ \int_0^T \varphi_0(u) \varphi_1(u) du & \int_0^T \varphi_1^2(u) du & \dots & \int_0^T \varphi_1(u) \varphi_k(u) du \\ \dots & \dots & \dots & \dots \\ \int_0^T \varphi_0(u) \varphi_k(u) du & \int_0^T \varphi_1(u) \varphi_k(u) du & \dots & \int_0^T \varphi_k^2(u) du \end{bmatrix},$$

тогда (2) перепишем в виде $\lambda \mathbf{X}_0 \hat{\boldsymbol{\theta}} = \mathbf{Y}$ и, следовательно,

$$\hat{\boldsymbol{\theta}} = \frac{1}{\lambda} \mathbf{X}_0^{-1} \mathbf{Y}, \quad (4)$$

что и дает явное выражение для оценок неизвестных параметров.

Найдем аналитическое выражение для неизвестных функций $\varphi_s(t_i)$, $s=0, k$.

$$\hat{\theta}_s = \sum_{i=1}^N x_i^2 \psi_s(t_i), \quad (5)$$

Введем обозначения:

$$\boldsymbol{\Psi} = \begin{bmatrix} \psi_0(t_1) & \psi_0(t_2) & \dots & \psi_0(t_N) \\ \psi_1(t_1) & \psi_1(t_2) & \dots & \psi_1(t_N) \\ \dots & \dots & \dots & \dots \\ \psi_k(t_1) & \psi_k(t_2) & \dots & \psi_k(t_N) \end{bmatrix}, \quad \mathbf{X}_2 = \begin{bmatrix} x^2(t_1) \\ x^2(t_2) \\ \dots \\ x^2(t_N) \end{bmatrix},$$

тогда (5) будет иметь вид: $\hat{\boldsymbol{\theta}} = \boldsymbol{\Psi} \mathbf{X}_2$.

Для нахождения выражений для функций $\varphi_s(t_i)$, $s=0, k$ потребуем, чтобы выполнялось условие $M[\hat{\boldsymbol{\theta}}] = \boldsymbol{\theta}$,

$$\text{тогда } M\left[\frac{1}{\lambda} \mathbf{X}_0^{-1} \mathbf{Y}\right] = M[\boldsymbol{\Psi} \mathbf{X}_2] \text{ или } \frac{1}{\lambda} \mathbf{X}_0^{-1} \mathbf{Y} = \boldsymbol{\Psi} \mathbf{X}_2,$$

откуда получаем

$$\boldsymbol{\Psi} = \frac{1}{\lambda} \mathbf{X}_0^{-1} \boldsymbol{\Phi}^T, \quad (6)$$

где $\mathbf{X}_0$ – это матрица из (3).

Свойства оценок параметров

Найдем статистические характеристики полученных оценок.

$$M[\hat{\boldsymbol{\theta}}] = M\left[\frac{1}{\lambda} \mathbf{X}_0^{-1} \mathbf{Y}\right] = \frac{1}{\lambda} \mathbf{X}_0^{-1} M[\boldsymbol{\Phi}^T \boldsymbol{\Phi} \boldsymbol{\theta}] = \boldsymbol{\theta}, \quad (7)$$

и следовательно полученные оценки являются несмещенными.

Найдем теперь ковариационную матрицу оценок $\hat{\boldsymbol{\theta}}$. Имеем

$$\begin{aligned} \mathbf{V}[\hat{\boldsymbol{\theta}}] &= M[(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})^T] = \\ &= M\left[\left(\frac{1}{\lambda} \mathbf{X}_0^{-1} \mathbf{Y} - M\left[\frac{1}{\lambda} \mathbf{X}_0^{-1} \mathbf{Y}\right]\right) \times \right. \\ &\quad \left. \times \left(\frac{1}{\lambda} \mathbf{X}_0^{-1} \mathbf{Y} - M\left[\frac{1}{\lambda} \mathbf{X}_0^{-1} \mathbf{Y}\right]\right)^T\right] = \\ &= \frac{1}{\lambda^2} \mathbf{X}_0^{-1} \mathbf{V}[\mathbf{Y}] (\mathbf{X}_0^{-1})^T. \end{aligned} \quad (8)$$

Здесь

$$\begin{aligned} \mathbf{V}[\mathbf{Y}] &= M[\mathbf{Y} \mathbf{Y}^T] - M[\mathbf{Y}] M[\mathbf{Y}^T] = \\ &= 2N \begin{bmatrix} \sum_{i=1}^N dc^2 & \sum_{i=1}^N dca & \dots & \sum_{i=1}^N dcb \\ \sum_{i=1}^N dca & \sum_{i=1}^N da^2 & \dots & \sum_{i=1}^N dab \\ \dots & \dots & \dots & \dots \\ \sum_{i=1}^N dcb & \sum_{i=1}^N dab & \dots & \sum_{i=1}^N db^2 \end{bmatrix}, \end{aligned}$$

$$a = \varphi_1(t_i), \quad b = \varphi_k(t_i), \quad c = \varphi_0(t_i), \quad d = D^2(t_i). \quad (9)$$

При вычислении $\mathbf{V}[\mathbf{Y}]$ воспользуемся тем фактом, что $M[x_i^4] = 3D^2(t_i)$.

Рассмотрим теперь следующие частные случаи.

Линейный тренд дисперсии

Предположим теперь, что тренд дисперсии хорошо аппроксимируется прямой, т. е.

$$D[t_i] = a + \frac{b}{T} \left(t_i - \frac{T}{2}\right).$$

В данном случае $\hat{\boldsymbol{\theta}} = \begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix}$, причем сами оценки,

как и прежде, доставляются выражением (4). В данном случае функции $\varphi_s(t_i)$, $s=0, 1$ имеют вид:

$$\varphi_0(t_i) = 1, \quad \varphi_1(t_i) = \frac{t_i}{T} - \frac{1}{2}.$$

$$\text{Тогда матрица } \bar{\mathbf{X}} = \lambda T \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{12} \end{bmatrix}.$$

Убедившись, что определитель, соответствующий матрице $\bar{\mathbf{X}}$, отличен от нуля, нетрудно найти

$$\text{обратную матрицу } \bar{\mathbf{X}}^{-1} = \frac{1}{\lambda T} \begin{bmatrix} 1 & 0 \\ 0 & 12 \end{bmatrix}.$$

В соответствии с равенством (6)

$$\boldsymbol{\Psi} = \frac{1}{\lambda T} \begin{bmatrix} 1 & 0 \\ 0 & 12 \end{bmatrix} \boldsymbol{\Phi}^T,$$

где матрица $\boldsymbol{\Phi}$ – матрица размерности $N \times 2$ вида

$$\boldsymbol{\Phi} = \begin{bmatrix} 1 & \frac{t_1}{T} - \frac{1}{2} \\ 1 & \frac{t_1}{T} - \frac{1}{2} \\ \dots & \dots \\ 1 & \frac{t_1}{T} - \frac{1}{2} \end{bmatrix}.$$

Зная $\boldsymbol{\Psi}$, записываем явное выражение для оценок неизвестных параметров:

$$\begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix} = \frac{1}{\lambda T} \begin{bmatrix} \sum_{i=1}^N x^2(t_i) \\ 12 \sum_{i=1}^N x^2(t_i) \left(\frac{t_i}{T} - \frac{1}{2} \right) \end{bmatrix}.$$

Исследуем статистические характеристики полученных оценок:

$$M[\hat{a}] = M \left[\frac{1}{\lambda T} \sum_{i=1}^N x^2(t_i) \right] = \frac{1}{\lambda T} \sum_{i=1}^N M[x^2(t_i)].$$

Усреднение по величинам x_i приводит к выражению

$$\frac{1}{\lambda T} \sum_{i=1}^N D[t_i] = \frac{1}{\lambda T} \sum_{i=1}^N \left[a + b \left(\frac{t_i}{T} - \frac{1}{2} \right) \right].$$

Полученное выражение усредним по величинам t_i , применяя методику [2]

$$\frac{1}{\lambda T} \lambda \int_0^T \left(a + b \left(\frac{u}{T} - \frac{1}{2} \right) \right) du = a,$$

таким образом, имеем $M[\hat{a}] = a$.

Аналогично показываем несмещенность оценки параметра b .

Ковариационная матрица оценок неизвестных параметров a, b задается выражением (8), в котором матрица $V[Y]$ задается выражением (9). В данном случае матрица $V[Y]$ будет иметь вид:

$$V[Y] = 2N \begin{bmatrix} \sum_{i=1}^N D^2(t_i) & \sum_{i=1}^N D^2(t_i) \left(\frac{t_i}{T} - \frac{1}{2} \right) \\ \sum_{i=1}^N D^2(t_i) \left(\frac{t_i}{T} - \frac{1}{2} \right) & \sum_{i=1}^N D^2(t_i) \left(\frac{t_i}{T} - \frac{1}{2} \right)^2 \end{bmatrix}.$$

Тогда ковариационная матрица оценок параметров тренда дисперсии представляет собой

$$V[\hat{\theta}] = \frac{2N}{\lambda^2 T^2} \times \begin{bmatrix} \sum_{i=1}^N D^2(t_i) & 12 \sum_{i=1}^N D^2(t_i) \left(\frac{t_i}{T} - \frac{1}{2} \right) \\ 12 \sum_{i=1}^N D^2(t_i) \left(\frac{t_i}{T} - \frac{1}{2} \right) & 144 \sum_{i=1}^N D^2(t_i) \left(\frac{t_i}{T} - \frac{1}{2} \right)^2 \end{bmatrix}.$$

Найдем $M[V[\hat{\theta}]]$. Для этого усредним элементы матрицы $V[\hat{\theta}]$ по величинам t_i , тогда

$$M[V[\hat{\theta}]] = \frac{2N}{\lambda T} \begin{bmatrix} a^2 + \frac{b^2}{12} & 2ab \\ 2ab & 12a^2 + \frac{9}{5}b^2 \end{bmatrix}.$$

Так как параметры a, b нам неизвестны, а известны лишь их оценки, то в последнем выражении заменим неизвестные параметры на их оценки.

Квадратичный тренд дисперсии

Пусть тренд дисперсии хорошо аппроксимируется параболой $D[t_i] = a + bt_i + ct_i^2$.

Задача выделения тренда в этом случае заключается в нахождении оценок неизвестных параметров a, b, c .

В данном случае $\hat{\theta} = \begin{bmatrix} \hat{a} \\ \hat{b} \\ \hat{c} \end{bmatrix}$, и сами оценки, как и

прежде, доставляются выражением (5). Функции $\varphi_s(t_i)$, $s=0,2$ будут иметь вид: $\varphi_0(t_i)=1$, $\varphi_1(t_i)=t_i$, $\varphi_2(t_i)=t_i^2$. Тогда матрица

$$\bar{X} = \lambda T \begin{bmatrix} 1 & \frac{T}{2} & \frac{T^2}{3} \\ \frac{T}{2} & \frac{T^2}{3} & \frac{T^3}{4} \\ \frac{T^2}{3} & \frac{T^3}{4} & \frac{T^4}{5} \end{bmatrix} = \lambda T X_0,$$

$$\text{где } X_0 = \begin{bmatrix} 1 & \frac{T}{2} & \frac{T^2}{3} \\ \frac{T}{2} & \frac{T^2}{3} & \frac{T^3}{4} \\ \frac{T^2}{3} & \frac{T^3}{4} & \frac{T^4}{5} \end{bmatrix}.$$

Убедившись, что определитель, соответствующий матрице X_0 , отличен от нуля, находим обратную матрицу

$$X_0^{-1} = \frac{720}{T^4} \begin{bmatrix} \frac{T^4}{80} & -\frac{T^3}{20} & \frac{T^2}{24} \\ -\frac{T^3}{20} & \frac{4T^2}{15} & -\frac{T}{4} \\ \frac{T^2}{24} & -\frac{T}{4} & \frac{1}{4} \end{bmatrix}.$$

В соответствии с равенством (6)

$$\Psi = \frac{1}{\lambda T} \frac{720}{T^4} \begin{bmatrix} \frac{T^4}{80} & -\frac{T^3}{20} & \frac{T^2}{24} \\ -\frac{T^3}{20} & \frac{4T^2}{15} & -\frac{T}{4} \\ \frac{T^2}{24} & -\frac{T}{4} & \frac{1}{4} \end{bmatrix} \cdot \Phi^T,$$

$$\text{где } \Phi^T = \begin{bmatrix} 1 & 1 & \dots & 1 \\ t_1 & t_2 & \dots & t_N \\ t_1^2 & t_2^2 & \dots & t_N^2 \end{bmatrix}.$$

Зная матрицу Ψ , записываем явное выражение для оценок неизвестных параметров:

$$\begin{bmatrix} \hat{a} \\ \hat{b} \\ \hat{c} \end{bmatrix} = \frac{720}{\lambda T^5} \begin{bmatrix} \sum_{i=1}^N \left(\frac{T^4}{80} - \frac{T^3}{20} t_i + \frac{T^2}{24} t_i^2 \right) x^2(t_i) \\ \sum_{i=1}^N \left(-\frac{T^3}{20} + \frac{4T^2}{15} t_i - \frac{T}{4} t_i^2 \right) x^2(t_i) \\ \sum_{i=1}^N \left(\frac{T^2}{24} - \frac{T}{4} t_i + \frac{1}{4} t_i^2 \right) x^2(t_i) \end{bmatrix}.$$

Исследуем статистические характеристики полученных оценок.

$$M[\hat{a}] = \frac{720}{\lambda T^5} \sum_{i=1}^N M \left[\left(\frac{T^4}{80} - \frac{T^3}{20} t_i + \frac{T^2}{24} t_i^2 \right) x^2(t_i) \right].$$

Усреднение по величинам x_i приводит к выражению

$$\begin{aligned} & \frac{720}{\lambda T^5} \sum_{i=1}^N \left(\frac{T^4}{80} - \frac{T^3}{20} t_i + \frac{T^2}{24} t_i^2 \right) D(t_i) = \\ & = \frac{720}{\lambda T^5} \sum_{i=1}^N \left(\frac{T^4}{80} - \frac{T^3}{20} t_i + \frac{T^2}{24} t_i^2 \right) (a + b t_i + c t_i^2). \end{aligned}$$

Полученное выражение усредним по величинам t_i , применяя методику [2]

$$\frac{720}{\lambda T^5} \lambda \int_0^T \left(\frac{T^4}{80} - \frac{T^3}{20} u + \frac{T^2}{24} u^2 \right) (a + b u + c u^2) du = a.$$

Аналогично показываем несмещенность оценок параметров b и c .

Ковариационная матрица оценок неизвестных параметров a, b, c задается выражением (8), в котором матрица $V[Y]$ задана выражением (9). В данном случае матрица $V[Y]$ будет иметь вид

$$V[Y] = 2N \begin{bmatrix} \sum_{i=1}^N D^2[t_i] & \sum_{i=1}^N D^2[t_i] t_i & \sum_{i=1}^N D^2[t_i] t_i^2 \\ \sum_{i=1}^N D^2[t_i] t_i & \sum_{i=1}^N D^2[t_i] t_i^2 & \sum_{i=1}^N D^2[t_i] t_i^3 \\ \sum_{i=1}^N D^2[t_i] t_i^2 & \sum_{i=1}^N D^2[t_i] t_i^3 & \sum_{i=1}^N D^2[t_i] t_i^4 \end{bmatrix}.$$

Выделение тренда дисперсии в виде сплайна первого порядка

Уточним поставленную задачу для случая, когда функции $\varphi_s(t)$, $s=\overline{0, k}$ – заданные функции времени. Разобьем весь отрезок наблюдений $[0; T]$ на части: $[0; T_0], [T_0; 2T_0], \dots, [(k-1)T_0; kT_0]$. На каждом таком отрезке тренд оценивается в виде полинома первой степени. На границах отрезков эти полиномы сшиваются так, чтобы получилась непрерывная кривая, которая и называется сплайном. Пусть, как и прежде, тренд дисперсии в момент времени t_i мож-

жет быть представлен $D[t_i] = \sum_{s=0}^k \theta_s \varphi_s(t_i)$, где $\varphi_s(t_i)$,

$s=\overline{0, k}$ – это функции из [3], проиллюстрированные на рисунке.

Оценки неизвестных параметров θ_s , $s=\overline{0, k}$ в данном случае задаются выражением (4). Для нахождения X_0^{-1} выражения (4) рассмотрим

$$\begin{aligned} X &= \Phi^T \Phi = \\ &= \begin{bmatrix} \sum_{i=1}^{T_0} (1-a)^2 & \sum_{i=1}^{T_0} a(1-a) & \dots & 0 \\ \sum_{i=1}^{T_0} a(1-a) & \sum_{i=1}^{T_0} a^2 + \sum_{i=T_0+1}^{2T_0} (2-a)^2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \sum_{i=(k-1)T_0+1}^{kT_0} (a-k+1)^2 \end{bmatrix}, \\ & a = \frac{t}{T_0}. \end{aligned}$$

Матрица $\bar{X} = M[X]$ будет иметь вид

$$\begin{aligned} \bar{X} &= \lambda \times \\ & \times \begin{bmatrix} \int_0^{T_0} (1-a)^2 dt & \int_0^{T_0} a(1-a) dt & \dots & 0 \\ \int_0^{T_0} a(1-a) dt & \int_0^{T_0} a^2 dt + \int_{T_0}^{2T_0} (2-a)^2 dt & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \int_{(k-1)T_0}^{kT_0} (a-k+1)^2 dt \end{bmatrix}, \\ & a = \frac{t}{T_0}. \end{aligned}$$

Вычислив интегралы, входящие в матрицу $\bar{X}$, получим

$$\bar{X} = \frac{\lambda T_0}{6} \begin{bmatrix} 2 & 1 & \dots & 0 \\ 1 & 4 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 2 \end{bmatrix}.$$

Пусть

$$X_0 = \begin{bmatrix} 2 & 1 & \dots & 0 \\ 1 & 4 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 2 \end{bmatrix}.$$

$$\begin{aligned} \varphi_0(t_i) &= \begin{cases} 1 - \frac{t_i}{T_0}, & \text{если } t_i \in [0; T_0], \\ 0, & \text{если } t_i \notin [0; T_0], \end{cases} \\ \varphi_s(t_i) &= \begin{cases} \frac{t_i}{T_0} + 1 - s, & \text{если } t_i \in [(s-1)T_0; sT_0], \\ s + 1 - \frac{t_i}{T_0}, & \text{если } t_i \in [sT_0; (s+1)T_0], \\ 0, & \text{если } t_i \notin [(s-1)T_0; (s+1)T_0], \end{cases} \quad s=\overline{1, k-1}, \\ \varphi_k(t_i) &= \begin{cases} \frac{t_i}{T_0} - k + 1, & \text{если } t_i \in [(k-1)T_0; kT_0], \\ 0, & \text{если } t_i \notin [(k-1)T_0; kT_0]. \end{cases} \end{aligned}$$

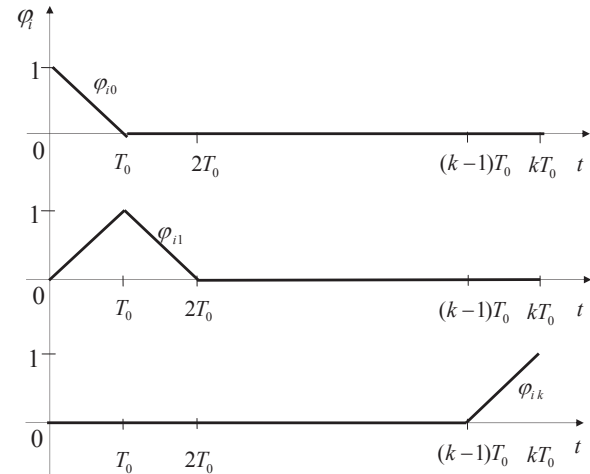


Рисунок. Графики функций $\varphi_s(t)$, $s=\overline{0, k}$

Для нахождения $\mathbf{X}_0^{-1}$ воспользуемся алгоритмом [4].

В соответствии с этим алгоритмом элемент матрицы $\mathbf{X}_0^{-1}$ (обозначим его z_{ij}) задается выражением

$$z_{ij} = \frac{1}{d_k} \begin{cases} (-1)^{i-j} \beta_{i-1} \beta_{k-j+1}, & \text{если } i \leq j, \\ (-1)^{j-i} \beta_{j-1} \beta_{k-i+1}, & \text{если } i > j, \end{cases}$$

где

$$\beta_s = \frac{(2 + \sqrt{3})^s + (2 - \sqrt{3})^s}{2}, \quad d_k = 2\beta_k - \beta_{k-1}.$$

Тогда

$$\bar{\mathbf{X}}^{-1} = \frac{6}{T_0 d_k} \times \begin{bmatrix} \beta_0 \beta_k & -\beta_0 \beta_{k-1} & \dots & (-1)^{-k} \beta_0^2 \\ -\beta_0 \beta_{k-1} & \beta_1 \beta_{k-1} & \dots & (-1)^{1-k} \beta_0 \beta_1 \\ \dots & \dots & \dots & \dots \\ (-1)^{-k} \beta_0^2 & (-1)^{1-k} \beta_0 \beta_1 & \dots & \beta_0 \beta_k \end{bmatrix}.$$

Таким образом, $\hat{\theta} = \bar{\mathbf{X}}^{-1} \mathbf{Y}$ дает выражение для оценок неизвестных параметров. Здесь

$$\mathbf{Y} = \begin{bmatrix} \sum_{t=1}^{T_0} x^2(t) \left(1 - \frac{t}{T_0}\right) \\ \sum_{t=1}^{T_0} x^2(t) \frac{t}{T_0} + \sum_{t=T_0+1}^{2T_0} x^2(t) \left(2 - \frac{t}{T_0}\right) \\ \dots \\ \sum_{t=(k-1)T_0+1}^{kT_0} x^2(t) \left(\frac{t}{T_0} - k + 1\right) \end{bmatrix}.$$

СПИСОК ЛИТЕРАТУРЫ

1. Вентцель Е.С. Теория вероятностей. – М.: Наука, 1964. – 576 с.
2. Идрисов Ф.Ф. Выделение трендов временных рядов при наличии ошибок в измерениях моментов времени // Изв. вузов. Физика. – 1999. – Т. 42. – № 4. – С. 11–16.
3. Константинова И.Г. Выделение трендов временных рядов при случайном числе измерений сплайнами первого порядка //

Исследуем статистические характеристики полученных оценок. Несмещенность полученных оценок вытекает из выражения (7). Ковариационная матрица оценок параметров тренда дисперсии, доставляемая выражением (8), зависит от неизвестных параметров $\theta_s, s=\overline{0, k}$. Поэтому заменим неизвестные параметры $\theta_s, s=\overline{0, k}$ на их оценки. Знание матрицы вариаций позволяет обычным способом строить доверительные интервалы для неизвестных параметров.

Выводы

Таким образом, в результате проведенных исследований:

- 1) на основе метода наименьших квадратов получены оценки параметров тренда дисперсии при пуассоновском потоке моментов измерений;
- 2) рассмотрены следующие частные случаи: линейный тренд, квадратичный и в виде сплайна первого порядка;
- 3) исследованы статистические характеристики полученных оценок.

Математическое моделирование. Кибернетика. Информатика / под ред. А.М. Горцева. – Томск: Изд-во ТГУ, 1999. – С. 81–88.

4. Ильин В.П., Кузнецов Ю.И. Тредиагональные матрицы и их приложения. – М.: Наука, 1985. – 208 с.

Поступила 06.11.2012 г.

ОЦЕНКА ФУНКЦИИ АВТОКОРРЕЛЯЦИИ В ВИДЕ СПЛАЙНА ПЕРВОГО ПОРЯДКА

В.Н. Устинова, И.Г. Устинова

Томский политехнический университет

E-mail: igu@sibmail.com

Предложен метод выделения тренда автокорреляционной функции в виде сплайна первого порядка. На примере модельной сейсмической трассы и функции Бесселя показано, что предложенный метод дает лучшее приближение автокорреляционной функции, чем приближение, полученное с использованием стандартной формулы для оценивания функции автокорреляции.

Ключевые слова:

Автокорреляционная функция, тренд автокорреляционной функции, сплайн первого порядка, сейсмические данные, обработка сейсмической информации.

Key words:

Autocorrelation function, trend of the autocorrelation function, first-order spline, seismic data, processing of seismic data.

В рамках эргодичной и стационарной модели геофизического поля важнейшие сведения о свойствах сейсмического сигнала можно получить по автокорреляционной функции $R[\tau]$, энергетическому спектру $S[\omega]$, математическому ожиданию $M[x]$ [1]. Наиболее информативными для оценки свойств составляющих сейсмического сигнала и полей сейсмических параметров являются функция автокорреляции и энергетический спектр [2, 3]. Оценка функции автокорреляции определяется, согласно [4], выражением

$$\hat{R}[\tau] = \frac{1}{n-|\tau|} \sum_{i=1}^{n-|\tau|} (y_i - \bar{y})(y_{i+\tau} - \bar{y}), \quad (1)$$

где $y_1, y_2, \dots, y_n$ – текущие значения геофизического по-

ля F в i -й точке сейсмической трассы; $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$,

τ – разность аргументов: $\tau = t_j - t_i$, $i, j = \overline{1, n}$. Анализ формы автокорреляционной функции (АКФ) нашел применение при решении следующих задач обработки сейсмических данных:

1. Оценка корреляционных свойств сигналов и помех.
2. Расчеты весовых функций и частотных характеристик оптимальных фильтров базируются на знании АКФ сигналов и помех [3].
3. Оценка разрешающей способности сейсмической записи [4].

Часто для оценки функции автокорреляции приходится использовать короткие отрезки сейсмических записей, поэтому вопрос о точности оценивания автокорреляционной функции имеет большое практическое значение. В данной статье выделен тренд функции автокорреляции в виде сплайна первого порядка.

Постановка задачи

Функция автокорреляции $R[\tau]$ стационарного случайного процесса является одной из его важнейших характеристик второго порядка [5]. Приведем описание математической модели процесса, которая будет исследована далее.

Пусть $y(t)$ – гауссовский стационарный случайный процесс с математическим ожиданием $M[y(t)] = 0$ и функцией автокорреляции $R[\tau] = R[t_j - t_i] = M[y(t_i)y(t_i + \tau)]$. Пусть изучаемый процесс наблюдается на отрезке времени $[0; T]$, и моменты времени, в которые производятся измерения t_i , известны точно. Результатами измерений являются величины $y_i = y(t_i)$, $i = \overline{1, N}$. По результатам измерений $\{y_i, i = \overline{1, N}\}$ требуется построить оценку $\hat{R}[\tau]$ функции корреляции $R[\tau]$ на интервале значений $[0; T]$ аргумента τ .

Решение задачи

Пусть на отрезке значений аргумента $[(k-1)\tau_0; k\tau_0]$ функция автокорреляции задана выражением

$$R[\tau] = R_{k-1} \frac{k\tau_0 - \tau}{\tau_0} + R_k \frac{\tau - (k-1)\tau_0}{\tau_0}, \quad k = 1, 2, \dots, \quad (2)$$

т. е. функция автокорреляции имеет вид сплайна первого порядка. В этом случае решение задачи об аппроксимации функции автокорреляции состоит в нахождении оценок параметров $R_0, R_1, \dots, R_n$,

параметр R_0 можно определить как $\hat{R}_0 = \frac{1}{N} \sum_{i=1}^N y_i^2$.

Заметим, что последнее выражение является несмещенной оценкой R_0 . Для нахождения оценки R_k рассмотрим выражение

$$S_k = \sum_{i,j \in M_k} y_i y_j \left[A \frac{k\tau_0 - (t_j - t_i)}{\tau_0} + B \frac{(t_j - t_i) - (k-1)\tau_0}{\tau_0} \right],$$

где A, B – неизвестные параметры, которые подлежат определению, а i, j – номера моментов измерений, M_k – множество индексов (i, j) , удовлетворяющих условию $M_k = \{(i, j): (k-1)\tau_0 \leq t_j - t_i \leq k\tau_0\}$. Усредняя S_k по реализациям процесса $y(t)$, получим

$$M[S_k] = \sum_{i,j \in M_k} R[t_j - t_i] \left[A \frac{k\tau_0 - (t_j - t_i)}{\tau_0} + B \frac{(t_j - t_i) - (k-1)\tau_0}{\tau_0} \right]$$

или

$$M[S_k] = \int_{(k-1)\tau_0}^{k\tau_0} R[\tau] \left[A \frac{k\tau_0 - \tau}{\tau_0} + B \frac{\tau - (k-1)\tau_0}{\tau_0} \right] d\tau.$$

Подставляя в последнее выражение (2) и вычисляя полученный интеграл, будем иметь

$$M[S_k] = \frac{\tau_0}{6} [(2A+B)R_{k-1} + (A+2B)R_k].$$

Оценку $\hat{R}_k$ параметра R_k найдем из уравнения

$$S_k = \frac{\tau_0}{6} [(2A+B)R_{k-1} + (A+2B)R_k],$$

из которого получим

$$\sum_{i,j \in M_k} y_i y_j \left[A \frac{k\tau_0 - (t_j - t_i)}{\tau_0} + B \frac{(t_j - t_i) - (k-1)\tau_0}{\tau_0} \right] = \frac{\tau_0}{6} [(2A+B)R_{k-1} + (A+2B)R_k],$$

откуда

$$\hat{R}_k = -\frac{2A+B}{A+2B} \hat{R}_{k-1} + \frac{6}{\tau_0} \frac{1}{A+2B} \times \sum_{i,j \in M_k} y_i y_j \left[A \frac{k\tau_0 - (t_j - t_i)}{\tau_0} + B \frac{(t_j - t_i) - (k-1)\tau_0}{\tau_0} \right].$$

Заметим, что полученное уравнение является разностным уравнением первого порядка с начальным условием $\hat{R}_0 = \frac{1}{N} \sum_{i=1}^N y_i^2$ [6]. Для того чтобы решение уравнения в конечных разностях было устойчивым, необходимо, согласно [7], чтобы

$$\left| \frac{2A+B}{A+2B} \right| < 1, \quad (3)$$

отсюда, используя графический метод, получаем следующее решение неравенства, представленное на рис. 1.

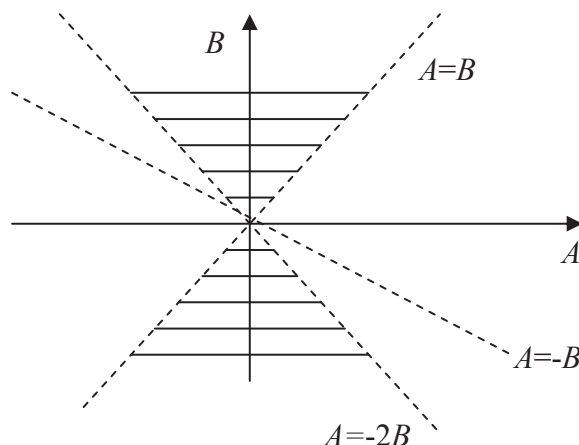


Рис. 1. Решение неравенства (3) графическим методом

Рассмотрим A и B такие, чтобы $\frac{2A+B}{A+2B} = 0$, например, $A = -\frac{1}{2}$, $B=1$, тогда

$$\hat{R}_k = \frac{2}{\tau_0} \sum_{i,j \in M_k} y_i y_j \frac{3(t_j - t_i) - (3k-2)\tau_0}{\tau_0}, \quad (4)$$

причем $M[\hat{R}_k] = R_k$, то есть полученная оценка является несмещенной.

Имитационное моделирование полученных оценок

Все расчеты и построение графиков выполнены в системе Mathcad 2000 [8].

В качестве модели сейсмического сигнала возьмем функцию Бесселя первого рода нулевого порядка, реализация которой изображена на рис. 2, а, для которой построим функцию автокорреляции, используя выражение (1) (рис. 2, б) и выражение (4) в виде сплайна первого порядка (рис. 2, в).

Согласно [9] в приложениях при анализе временных последовательностей часто применяются стационарные случайные процессы с показательной функцией автокорреляции

$$R[\tau] = e^{-\alpha|\tau|}, \quad -\infty < \tau < +\infty, \quad \alpha > 0. \quad (5)$$

Другим часто встречающимся типом АКФ стационарной случайной функции является показательно-косинусная автокорреляционная функция

$$R[\tau] = e^{-\alpha|\tau|} \cos \omega_0 \tau, \quad -\infty < \tau < +\infty, \quad \alpha > 0, \quad (6)$$

где ω_0 – видимый период колебаний, а α – коэффициент затухания. На практике, как правило, выражение (6) используется для описания формы автокорреляционной функции сейсмических колебаний, представленных моделью случайного процесса. Третьей часто встречающейся автокорреляционной функцией стационарного случайного процесса является АКФ, задаваемая выражением

$$R[\tau] = e^{-\alpha|\tau|} (\cos \omega_0 \tau + \gamma \sin \omega_0 |\tau|), \quad -\infty < \tau < +\infty, \quad \alpha > 0, \quad \gamma < \frac{\alpha}{\omega_0}. \quad (7)$$

Процесс с такой автокорреляционной функцией не имеет угловых точек, поэтому данная АКФ дифференцируема в среднем квадратическом смысле. Автокорреляционные функции (5)–(7) изображены на рис. 2, е. На рис. 2, з изображены все пять рассмотренные автокорреляционные функции (1, 4, 5–7). Из рис. 2 видно, что форму исходной функции (сигнала) отражает лучше автокорреляционная функция, построенная в виде сплайна первого порядка. Аналогичный вывод делаем на основе доверительного интервала для среднего значения функции автокорреляции. Если в качестве доверительной вероятности γ взять 0,95, то доверительный интервал для среднего значения АКФ, построенной по формуле (1), есть $(-0,016; 0,072)$, для функции, полученной с использованием выражения (4), – $(-0,056; 0,014)$.

Аналогичные расчеты были проведены и для модельного сейсмического сигнала, реализация которого изображена на рис. 2, г, для которого построена функция автокорреляции в виде сплайна первого порядка, рис. 2, д. На рис. 2, ж изображены три ранее рассмотренные модельные функции автокорреляции (5–7) и одна, построенная по формулам (4).

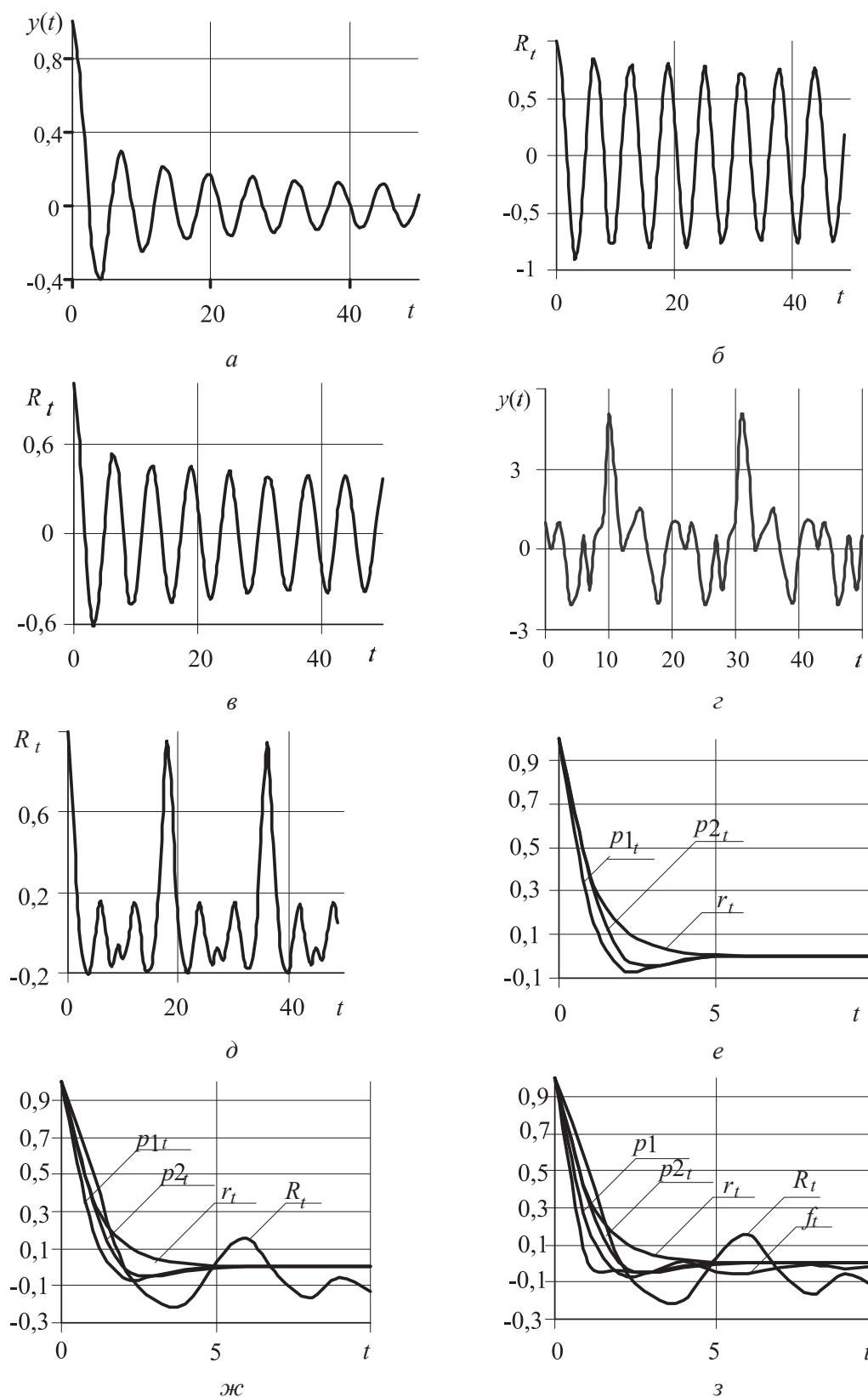


Рис. 2. Графики моделей сейсмических сигналов, автокорреляционных функций: а) функция Бесселя первого рода нулевого порядка, б) реализация модельного сейсмического сигнала, в) д) функция автокорреляции, вычисленная с использованием выражения (1), (4), АКФ модельного сигнала в виде сплайна первого порядка, е) три модельные АКФ (формулы 5-7); $p1_t$ обозначена автокорреляционная функция (5), $p2_t$ – АКФ (6), и r_t – АКФ (7), ж) автокорреляционные функции (4-7) $p1_t$, $p2_t$, r_t и R_t – автокорреляционная функция, построенная по формуле (4), з) изображены все пять АКФ (1, 4, 5-7) $p1_t$, $p2_t$, r_t , R_t , f_t – обозначение автокорреляционной функции (1)

В таблице приведена сумма квадратов отклонений оцененных значений функции автокорреляции от истинных. В верхней строке представлены номера расчетных формул, во второй — искомые значения сумм.

Таблица. Значение суммы квадратов отклонений истинных значений функции автокорреляции от оценок, найденных по формулам (4)–(7)

Номер формулы	4	5	6	7
Значение суммы	0,054	0,227	0,132	0,217

Из приведенной таблицы можно сделать вывод, что наименьшее значение суммы получается при оценивании функции автокорреляции сплайном первого порядка.

Выводы

Выполнен расчет коэффициентов сплайна первого порядка в приложении к оценке функции автокорреляции. Возможности предложенного метода апробированы на модельном сейсмическом сигнале и на модели, применяемой для аппроксимации сейсмических сигналов. В результате проведенного исследования установлено, что автокорреляционная функция, построенная в виде сплайна первого порядка, лучше отражает форму исходного процесса, нежели модельные автокорреляционные функции. Построенные доверительные интервалы для оценки математического ожидания автокорреляционной функции также свидетельствуют о более точном приближении в виде сплайна, чем в опубликованных работах.

СПИСОК ЛИТЕРАТУРЫ

1. Гордин В.Н., Бабаева Т.Н., Михайлов В.О. О статистической параметризации аномальных потенциальных полей // Геофизический журнал. — 1984. — Т. 6. — № 2. — С. 55–63.
2. Устинова В.Н., Устинова И.Г. Статистическая параметризация симметричных геофизических объектов // Известия ТПУ. — 2003. — № 5. — С. 36–41.
3. Устинова В.Н., Устинова И.Г. Дискретные иерархические системы в геофизике // Известия ТПУ. — 2012. — Т. 320. — № 1. — С. 91–97.
4. Никитин А.А. Теоретические основы обработки геофизической информации. — М.: Недра, 1986. — 342 с.
5. Идрисов Ф.Ф., Устинова И.Г. Оценка функции корреляции стационарного случайного процесса при случайном числе измерений // Экономика, технология, предпринимательство. — Вып. 1. — Томск: Изд-во ТГПУ, 2000. — С. 75–79.
6. Корн Г., Корн Т. Справочник по математике для научных работников. — М.: Наука, 1978. — 831 с.
7. Гельфонд А.О. Исчисление конечных разностей. — М.: Наука, 1967. — 375 с.
8. Дьяконов В. Mathcad 2000. — СПб.: Изд-во «Питер», 2001. — 592 с.
9. Пугачев В.С., Сеницын И.Н. Стохастические дифференциальные системы. Анализ и фильтрация. — М.: Наука, 1990. — 642 с.

Поступила 07.11.2012 г.

Управление техническими системами

УДК 519.874

УПРАВЛЕНИЕ ОПЦИОННЫМИ РЕСУРСАМИ

А.В. Китаева, Н.В. Степанова

Томский политехнический университет
E-mail: kit1157@yandex.ru

Рассмотрена модель с одним видом ресурса с ограниченным сроком использования, поставляемого в начале производственного цикла. Спрос на ресурс в процессе производства носит случайный характер: поток запросов образует стационарный процесс, объемы запросов есть независимые одинаково распределенные случайные величины с заданными средним и дисперсией. Найдены асимптотические распределения общего объема спроса на сырье в течение производственного цикла и длительности использования ресурса. Получены уравнения для статистического оценивания необходимых параметров и приближенное значение оптимального объема опционного ресурса в смысле максимума средней прибыли предприятия.

Ключевые слова:

Управление запасами, стохастическая модель, нормальная аппроксимация.

Key words:

Inventory control, stochastic model, normal approximation.

Введение и постановка задачи

Системное исследование моделей управления запасами, отражающих неопределенность и динамику, началось в начале 50-х гг. XX в. с работ K.J. Arrow, Th.E. Harris и J. Marschak [1], а также A. Dvoretzky, J. Kiefer и J. Wolfowitz [2]. Большое количество классических моделей управления запасами проанализировано в [3]. В настоящее время существует ряд стохастических моделей для решения проблемы управления запасами при различных условиях, встречающихся на практике, например работы Sh.M. Ross [4], S. Chopra и P. Meindl [5], D. Beyer et al. [6]. Отличие рассматриваемой задачи от традиционных задач управления запасами состоит в том, что мы не рассматриваем стоимость хранения сырья, и дополнительные поставки опционного сырья невозможны, т. е. его можно расценивать как некоторый не возобновляемый во время производственного цикла ресурс.

Рассмотрим следующую модель производственного процесса, в котором может быть использовано так называемое опционное сырье, т. е. его применение не является обязательным в производстве. В случае его отсутствия предприятие работает в обычном режиме и выпускает стандартную продукцию. Потребность в этом ресурсе возникает по запросу заказчиков, носит случай-

ный характер и выгодна производителю, поскольку в конечном счете он получает дополнительную прибыль в размере s денежных единиц на каждую единицу использованного сырья. Приобретенное сырье имеет ограниченный срок годности, по истечении которого подлежит утилизации. Примером может служить крупное предприятие, выпускающее некоторую однородную продукцию, имеющую стабильный рынок сбыта, при этом на предприятие время от времени (не регулярно) поступают специальные заказы. Любое коммерческое предприятие стремится максимизировать свою прибыль, поэтому здесь возникает оптимизационная задача, рассмотрению которой и посвящена данная работа.

Введем необходимые для математического моделирования задачи обозначения. Пусть ξ — это случайная величина, равная величине одного запроса на сырье, запросы приходят независимо друг от друга, $E\{\xi\}=a_1$, и $E\{\xi^2\}=a_2$, где символ $E\{\xi\}$ обозначает математическое ожидание величины ξ .

При осуществлении производственного цикла длительности T предприятие покупает некоторую партию опционного сырья объема Q по цене d за единицу. По истечении времени T сырье использовать не может и утилизируется по цене $d_{\text{ит}}$ за единицу.

Общий объём спроса на сырьё в течение производственного цикла

Рассмотрим случай, когда число запросов на сырьё $n \gg 1$ и образует стационарный случайный процесс. Пусть также длительность производственного цикла $T \gg 1$. Тогда n можно считать приближенно нормальной случайной величиной с $E\{n\} = m_T$ и дисперсией $D\{n\} = \sigma_T^2$, где величины m_T и σ_T^2 пропорциональны T [7, 8].

Обозначим через X общий объём спроса на сырьё в течение производственного цикла, через $p(\cdot)$ – функцию плотности вероятностей случайной величины X .

Теорема 1. При стационарности потока заказов и больших значениях n и T объём спроса на сырьё в течение производственного цикла можно считать нормальной случайной величиной со средним $m_x = a_1 m_T$ и дисперсией $\sigma_x^2 = m_T(a_2 - a_1^2) + \sigma_T^2 a_1^2$.

Доказательство. Пусть число запросов на сырьё n фиксировано, тогда объём X_n спроса на сырьё в течение производственного цикла можно представить в виде $X_n = \xi_1 + \xi_2 + \dots + \xi_n$, где ξ_i – величина, равная объёму i -го запроса. Плотность вероятностей $p(\cdot)$ можно найти по формуле полной вероятности

$$p(x) = \sum_{n=0}^{\infty} p_n(x) p_n, \text{ где } \{p_n, n=0, 1, \dots\} \text{ – распределе-}$$

ние вероятностей числа возможных запросов за время T , $p_n(\cdot)$ – плотность распределения X_n .

Обозначим через $g(\cdot)$ характеристическую функцию случайной величины ξ , то есть $g(\omega) = E\{e^{i\omega\xi}\}$, через $D\{\xi\} = a_2 - a_1^2$ – дисперсию величины ξ . Характеристическая функция величины X_n есть $g^n(\omega)$, и характеристическая функция $G_x(\cdot)$ ве-

личины X равна $G_x(\omega) = \sum_{n=0}^{\infty} g^n(\omega) p(n)$. Для $\ln g(\cdot)$

справедливо разложение

$$\ln g(\omega) = i\omega a_1 - \frac{\omega^2 D\{\xi\}}{2} + O(\omega^3). \quad (1)$$

Учитывая асимптотическую нормальность величины n , можно приближенно записать

$$\begin{aligned} G_x(\omega) &= \frac{1}{\sigma_T \sqrt{2\pi}} \int_{-\infty}^{\infty} g^z(\omega) e^{-\frac{(z-m_T)^2}{2\sigma_T^2}} dz = \\ &= \frac{1}{\sigma_T \sqrt{2\pi}} \int_{-\infty}^{\infty} e^{z \ln g(\omega)} e^{-\frac{(z-m_T)^2}{2\sigma_T^2}} dz. \end{aligned} \quad (2)$$

Принимая во внимание, что

$$\int_{-\infty}^{\infty} e^{-ax^2 + bx} dx = \sqrt{\frac{\pi}{a}} \exp\left(\frac{b^2}{4a}\right),$$

и вычисляя интеграл в (2), получаем

$$G_x(\omega) \approx \exp\left(m_T \ln g(\omega) + \frac{\sigma_T^2}{2} \ln^2 g(\omega)\right). \quad (3)$$

Подставляя разложение (1) в (3), имеем

$$G_x(\omega) =$$

$$= \exp\left(i\omega a_1 m_T - \frac{\omega^2}{2} (m_T D\{\xi\} + a_1^2 \sigma_T^2) + O(\omega^3) T\right).$$

Перейдём от величины X к величине

$$\zeta = \frac{x - a_1 m_T}{\sqrt{m_T D\{\xi\} + a_1^2 \sigma_T^2}}.$$

Используя свойства характеристических функций, получим, что

$$G_{\zeta}(\omega) = \exp\left(-\frac{\omega^2}{2} + O\left(\frac{\omega^3}{(m_T D\{\xi\} + a_1^2 \sigma_T^2)^{3/2}}\right) T\right).$$

При $T \rightarrow \infty$ имеем $\lim_{T \rightarrow \infty} G_{\zeta}(\omega) = \exp\left(-\frac{\omega^2}{2}\right)$, то

есть асимптотически ζ является стандартной нормальной случайной величиной.

Поэтому при больших n и T

$$\begin{aligned} p(x) &\approx \frac{1}{\sqrt{2\pi(m_T(a_2 - a_1^2) + \sigma_T^2 a_1^2)}} \times \\ &\times \exp\left(-\frac{(x - a_1 m_T)^2}{2(m_T(a_2 - a_1^2) + \sigma_T^2 a_1^2)}\right). \end{aligned} \quad (4)$$

Доказательство закончено.

Оптимальный объём партии сырья для производственного цикла

Найдем среднюю прибыль в течение производственного цикла. На покупку сырья объёма Q будет потрачено Qd денег. Если $X > Q$, то будет использована вся партия сырья. Это произойдёт с вероятностью

$\int_Q^{\infty} p(x) dx$, и предприятие дополнительно по-

лучит Qc денег. Если $X \leq Q$, то партия сырья объёма $Q - X$ останется неиспользованной, её придётся направить на утилизацию, будет получено cX , а потрачено $d_{\text{ут}}(Q - X)$ денег.

Средняя прибыль предприятия составит величину

$$\begin{aligned} S &= cQ \int_Q^{\infty} p(x) dx + c \int_0^Q xp(x) dx - \\ &- d_{\text{ут}} \int_0^Q (Q - x) p(x) dx - Qd. \end{aligned}$$

Оптимальный объём партии сырья для производственного цикла найдём из условия $S \Rightarrow \max_Q$, т. е.

$$\frac{dS}{dQ} = c \int_Q^{\infty} p(x) dx - d_{\text{ут}} \int_0^Q p(x) dx - d = 0.$$

Учитывая тождество $\int_0^Q p(x) dx = 1 - \int_Q^{\infty} p(x) dx$, по-

лучим уравнение для определения оптимального

объёма партии сырья Q_{opt} для производственного цикла

$$\int_{Q_{\text{opt}}}^{\infty} p(x)dx = \frac{d + d_{\text{ut}}}{c + d_{\text{ut}}}.$$

Таким образом, Q_{opt} является квантилем распределения вероятностей общего объёма спроса на сырьё в течение производственного цикла. Эту величину можно найти различными статистическими методами, наблюдая за процессом поступления спецзаказов или пользуясь приближенной формулой (4).

В последнем случае оптимальный объём Q_{opt} сырья для производственного цикла определится уравнением

$$\int_{Q_{\text{opt}}}^{\infty} p(x)dx = 1 - \Phi\left(\frac{Q_{\text{opt}} - a_1 m_T}{\sqrt{m_T(a_2 - a_1^2) + \sigma_T^2 a_1^2}}\right) = \frac{d + d_{\text{ut}}}{c + d_{\text{ut}}},$$

где $\Phi(\cdot)$ есть функция Лапласа. Обозначая через $\Psi(\cdot)$ функцию, обратную к функции $\Phi(\cdot)$, получаем окончательно

$$Q_{\text{opt}} = a_1 m_T + \sqrt{m_T(a_2 - a_1^2) + \sigma_T^2 a_1^2} \Psi\left(1 - \frac{d + d_{\text{ut}}}{c + d_{\text{ut}}}\right). \quad (5)$$

Рассмотрим два частных случая формулы (5).

1. Поток запросов является стационарным пуассоновским потоком интенсивности λ . Тогда $m_T = \lambda T$, $\sigma_T^2 = \lambda T$ и формула (5) приобретает вид

$$Q_{\text{opt}} = a_1 \lambda T + \sqrt{a_2 \lambda T} \Psi\left(1 - \frac{d + d_{\text{ut}}}{c + d_{\text{ut}}}\right).$$

2. Поток запросов является дважды стохастическим пуассоновским потоком, интенсивность $\lambda(t)$ этого потока является стационарным случайным процессом с $E\{\lambda(t)\} = \bar{\lambda}$ и функцией ковариации $\text{cov}(\lambda(t_1), \lambda(t_2)) = R(t_2 - t_1)$. Тогда, при фиксированной реализации $\lambda(t)$,

$$E\{n | \lambda(t)\} = \int_0^T \lambda(t) dt,$$

$$E\{n^2 | \lambda(t)\} = \int_0^T \lambda(t) dt + \int_0^T \int_0^T \lambda(t_1) \lambda(t_2) dt_1 dt_2.$$

Усредняя по реализациям $\lambda(t)$, получим

$$E\{n\} = m_T = \bar{\lambda} T,$$

$$\begin{aligned} E\{n^2\} &= \bar{\lambda} T + \int_0^T \int_0^T (\bar{\lambda}^2 + R(t_2 - t_1)) dt_1 dt_2 = \\ &= \bar{\lambda} T + m_T^2 + \int_0^T \int_0^T R(t_2 - t_1) dt_1 dt_2, \end{aligned}$$

так, что

$$\sigma_T^2 = \bar{\lambda} T + \int_0^T \int_0^T R(t_2 - t_1) dt_1 dt_2.$$

Последний интеграл можно преобразовать к виду

$$\int_0^T \int_0^T R(t_2 - t_1) dt_1 dt_2 = 2T \int_0^T \left(1 - \frac{w}{T}\right) R(w) dw.$$

При $T \rightarrow \infty$

$$\int_0^T \int_0^T R(t_2 - t_1) dt_1 dt_2 \sim 2T \int_0^{\infty} R(w) dw$$

и поэтому

$$\sigma_T^2 \sim \bar{\lambda} T + 2T \int_0^{\infty} R(w) dw = T \left(\bar{\lambda} + 2 \int_0^{\infty} R(w) dw \right).$$

Таким образом, в рассмотренных случаях однозначно определяется оптимальный объём Q_{opt} партии опционного сырья для производственного цикла.

Длительность использования опционного ресурса

Процесс длительности использования опционного ресурса в течение производственного цикла также представляет интерес, как для потенциальных заказчиков, так и для производителя.

Найдём плотность вероятностей $q(\cdot)$ длительности использования опционного сырья объёма Q в диффузионном приближении.

Будем считать, что запросы образуют стационарный случайный процесс. Обозначим

$$\lim_{T \rightarrow \infty} \frac{a_1 m_T}{T} = m_0, \quad \lim_{T \rightarrow \infty} \frac{m_T(a_2 - a_1^2) + \sigma_T^2 a_1^2}{T} = \sigma_0^2.$$

Тогда процесс спроса на сырьё $X(\cdot)$ удовлетворяет уравнению

$$dX(t) = m_0 dt + \sigma_0 dw_t,$$

где w_t — стандартный винеровский случайный процесс.

Обозначим через $\tau(x)$ случайное время достижения процессом $X(\cdot)$ порогового значения Q , если в начальный момент времени t значение процесса $X(\cdot)$ было равно x .

Теорема 2. В диффузионном приближении плотность вероятностей длительности использования опционного сырья объёма Q

$$q(t) = \frac{Q}{\sqrt{2\pi\sigma_0^2} t^{3/2}} \exp\left(-\frac{m_0^2}{2\sigma_0^2} \left(t - \frac{Q}{m_0}\right)^2\right), \quad t > 0.$$

Доказательство. Рассмотрим преобразование Лапласа от плотности вероятностей величины $\tau(x)$

$$g(s, x) = E\{e^{-s\tau(x)}\}.$$

Возьмем момент времени $t + \Delta t$. Тогда за промежуток времени Δt процесс $X(\cdot)$ приобретёт приращение Δx и время, оставшееся до достижения процессом $X(\cdot)$ порогового значения Q станет равным $\tau(x + \Delta x)$. Получаем соотношение

$$\begin{aligned} g(s, x) &= E\{e^{-s\tau(x)}\} = E\{e^{-s(\Delta t + \tau(x + \Delta x))}\} = \\ &= e^{-s\Delta t} E_{\Delta x}\{g(s, x + \Delta x)\}, \end{aligned}$$

где $E_{\Delta x}$ означает усреднение по величине Δx .

Разложим экспоненту и функцию $g(s, \cdot)$ в ряд:

$$e^{-s\Delta t} = 1 - s\Delta t + o(\Delta t),$$

$$g(s, x + \Delta x) = g(s, x) + \frac{\partial g(s, x)}{\partial x} \Delta x +$$

$$+ \frac{1}{2} \frac{\partial^2 g(s, x)}{\partial x^2} (\Delta x)^2 + o((\Delta x)^2).$$

Усредняя по Δx , получим

$$g(s, x) =$$

$$= (1 - s\Delta t) E_{\Delta x} \left\{ g(s, x) + \frac{\partial g(s, x)}{\partial x} \Delta x + \right. \\ \left. + \frac{1}{2} \frac{\partial^2 g(s, x)}{\partial x^2} (\Delta x)^2 \right\} + o(\Delta t) =$$

$$= (1 - s\Delta t) \left\{ g(s, x) + m_0 \Delta t \frac{\partial g(s, x)}{\partial x} + \right. \\ \left. + \frac{\sigma_0^2}{2} \frac{\partial^2 g(s, x)}{\partial x^2} \Delta t \right\} + o(\Delta t) =$$

$$= g(s, x) + \left\{ -s g(s, x) + m_0 \frac{\partial g(s, x)}{\partial x} + \right. \\ \left. + \frac{\sigma_0^2}{2} \frac{\partial^2 g(s, x)}{\partial x^2} \right\} \Delta t + o(\Delta t).$$

Поделив на Δt и перейдя к пределу при $\Delta t \rightarrow 0$, получим, что $g(s, x)$ удовлетворяет следующему дифференциальному уравнению:

$$\frac{\partial^2 g(s, x)}{\partial x^2} + \frac{2m_0}{\sigma_0^2} \frac{\partial g(s, x)}{\partial x} - \frac{2s}{\sigma_0^2} g(s, x) = 0 \quad (6)$$

с граничным условием $g(s, Q) = 1$ (так как $\tau(Q) = 0$).

Характеристическое уравнение $z^2 + 2 \frac{m_0}{\sigma_0^2} z - \frac{2s}{\sigma_0^2} = 0$ имеет корни

$$z_1(s) = \sqrt{\frac{m_0^2}{\sigma_0^4} + \frac{2s}{\sigma_0^2}} - \frac{m_0}{\sigma_0^2} > 0$$

$$\text{и } z_2(s) = -\sqrt{\frac{m_0^2}{\sigma_0^4} + \frac{2s}{\sigma_0^2}} - \frac{m_0}{\sigma_0^2} < 0.$$

Общее решение уравнения (6) имеет вид

$$g(s, x) = C_1(s) e^{z_1(s)x} + C_2(s) e^{z_2(s)x}.$$

При $x \rightarrow -\infty$ функция $g(s, x)$ должна стремиться к нулю, поскольку $\tau(x)$ должно бесконечно возрастать. Поэтому следует считать $C_2(s) \equiv 0$ и рассматривать решение в виде $g(s, x) = C_1(s) e^{z_1(s)x}$. Так как $g(s, Q) = 1$, то $g(s, x) = e^{z_1(s)(x-Q)}$.

Процесс производства начинается в момент времени $t=0$ со значения $x=0$. Поэтому нас интересует

$$g(s, 0) = e^{-z_1(s)Q} = \exp \left(\frac{m_0}{\sigma_0^2} Q - Q \sqrt{\frac{m_0^2}{\sigma_0^4} + \frac{2s}{\sigma_0^2}} \right). \quad (7)$$

Для нахождения плотности вероятностей $q(\cdot)$ длительности использования опционного сырья надо найти обратное преобразование Лапласа от $g(s, 0)$. Для этого запишем (7) в виде

$$g(s, 0) = e^{-z_1(s)Q} = \exp \left(\frac{m_0}{\sigma_0^2} Q - \sqrt{\frac{2}{\sigma_0^2}} Q \sqrt{s + \frac{m_0^2}{2\sigma_0^2}} \right).$$

Пользуясь свойствами преобразования Лапласа и формулой [9, Ф. 204], получим явный вид плотности $q(\cdot)$, что и требовалось доказать.

Найдём среднее $E\{\tau\}$ и дисперсию $D\{\tau\}$. Имеем

$$\psi(s) = \ln g(s, 0) = \frac{m_0}{\sigma_0^2} Q - Q \sqrt{\frac{m_0^2}{\sigma_0^4} + \frac{2s}{\sigma_0^2}}.$$

Отсюда

$$\psi'(s) = - \left[\frac{m_0^2}{\sigma_0^4} + \frac{2s}{\sigma_0^2} \right]^{-1/2} \frac{Q}{\sigma_0^2},$$

поэтому

$$E\{\tau\} = -\psi'(0) = \frac{Q}{m_0} = \frac{Q}{a_1 m_T} T.$$

Далее

$$\psi''(s) = \left[\frac{m_0^2}{\sigma_0^4} + \frac{2s}{\sigma_0^2} \right]^{-3/2} \frac{Q}{\sigma_0^4},$$

поэтому

$$D\{\tau\} = \psi''(0) = \frac{\sigma_0^2}{m_0^3} Q = \frac{m_T(a_2 - a_1^2) + \sigma_T^2 a_1^2}{a_1^3 m_T^3} Q T^2.$$

Найдём приближенное выражение для $q(\cdot)$ в случае, когда

$$\frac{E\{\tau\}}{\sqrt{D\{\tau\}}} = \sqrt{\frac{Q m_0}{\sigma_0^2}} \gg 1. \quad (8)$$

Условие (8) можно рассматривать как одно из условий стабильной работы производства.

Разлагая $g(s, 0)$ в ряд по малому параметру

$$\varepsilon = \sqrt{\frac{\sigma_0^2}{Q m_0}},$$

получим

$$g(s, 0) \approx \exp \left(-s \frac{Q}{m_0} + s^2 \frac{\sigma_0^2 Q}{m_0^3} \right),$$

что соответствует гауссовскому распределению

$$N \left(\frac{QT}{a_1 m_T}, \frac{m_T(a_2 - a_1^2) + \sigma_T^2 a_1^2}{a_1^3 m_T^3} Q T^2 \right).$$

Обозначим

$$a_1 m_T = m_x, \quad m_T(a_2 - a_1^2) + \sigma_T^2 a_1^2 = \sigma_x^2.$$

Тогда

$$\tau \sim N \left(\frac{QT}{m_x}, \frac{\sigma_x^2}{m_x^3} Q T^2 \right). \quad (9)$$

Статистическое оценивание числовых характеристик системы

Как видно из (5) и (9), оптимальный объём опционного сырья и распределение длительности ис-

пользования опционного сырья зависят от параметров m_x и σ_x^2 . Эти параметры можно определить на основе опытных данных. Рассмотрим один из возможных способов такого определения.

В течение каждого производственного цикла возможны два варианта:

1. Будет использовано $X < Q$ опционного ресурса. Пусть таких циклов было N , и мы получили выборку $x_1, x_2, x_3, \dots, x_N$.
2. Будет использован весь ресурс, длительность его использования $\tau < T$. Пусть таких циклов было M , и мы имеем выборку $t_1, t_2, \dots, t_M$.

В общем, мы получили выборку объёма $N+M$: $(x_1, x_2, x_3, \dots, x_N; t_1, t_2, \dots, t_M)$. Оценим параметры m_x и σ_x методом моментов.

Найдём $E\{X|X < Q\}$ – условное математическое ожидание величины X при условии, что в конце производственного цикла сырьё останется, т. е. $X < Q$.

Вероятность условия равна

$$\frac{1}{\sigma_x \sqrt{2\pi}} \int_{-\infty}^Q \exp\left(-\frac{(x-m_x)^2}{2\sigma_x^2}\right) dx = \Phi\left(\frac{Q-m_x}{\sigma_x}\right).$$

Далее

$$\begin{aligned} & \frac{1}{\sigma_x \sqrt{2\pi}} \int_{-\infty}^Q x e^{-\frac{(x-m_x)^2}{2\sigma_x^2}} dx = \\ & = \frac{1}{\sigma_x \sqrt{2\pi}} \int_{-\infty}^Q (x-m_x) e^{-\frac{(x-m_x)^2}{2\sigma_x^2}} dx + \\ & + m_x \frac{1}{\sigma_x \sqrt{2\pi}} \int_{-\infty}^Q e^{-\frac{(x-m_x)^2}{2\sigma_x^2}} dx = \\ & = m_x \Phi\left(\frac{Q-m_x}{\sigma_x}\right) - \frac{\sigma_x}{\sqrt{2\pi}} \exp\left(-\frac{(Q-m_x)^2}{2\sigma_x^2}\right). \end{aligned}$$

Обозначим

$$\phi(z) = \frac{e^{-z^2/2}}{\sqrt{2\pi}\Phi(z)}.$$

Тогда

$$E\{X | X < Q\} = m_x - \sigma_x \phi\left(\frac{Q-m_x}{\sigma_x}\right).$$

Так как длительность использования опционного ресурса имеет асимптотически нормальное распределение $\tau \sim N\left(\frac{QT}{m_x}, \frac{\sigma_x^2}{m_x^3} QT^2\right)$, то аналогично

$$E\{\tau | \tau < T\} = \frac{QT}{m_x} - \sqrt{\frac{\sigma_x^2 QT^2}{m_x^3}} \phi\left(\frac{T-QT/m_x}{\sqrt{\sigma_x^2 QT^2/m_x^3}}\right).$$

Найдём выборочные средние значения

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i, \quad \bar{t} = \frac{1}{M} \sum_{j=1}^M t_j.$$

Тогда по методу моментов оценки параметров $\hat{m}_x$ и $\hat{\sigma}_x$ следует искать из следующей системы уравнений:

$$\begin{aligned} & \hat{m}_x - \hat{\sigma}_x \phi\left(\frac{Q-\hat{m}_x}{\hat{\sigma}_x}\right) = \bar{x}, \\ & \frac{QT}{\hat{m}_x} - \sqrt{\frac{\hat{\sigma}_x^2 QT^2}{\hat{m}_x^3}} \phi\left(\frac{T-QT/\hat{m}_x}{\sqrt{\hat{\sigma}_x^2 QT^2/\hat{m}_x^3}}\right) = \bar{t}. \end{aligned} \quad (10)$$

Рассмотрим оценки, полученные по методу максимального правдоподобия.

Учитывая полученные результаты, для совместной плотности вероятностей выборочных значений можно записать функцию правдоподобия

$$\begin{aligned} L = & \prod_{i=1}^N \frac{1}{\sigma_x \sqrt{2\pi}} \exp\left(-\frac{(x_i-m_x)^2}{2\sigma_x^2}\right) \times \\ & \times \prod_{j=1}^M \frac{1}{\sqrt{2\pi\sigma_x^2 QT^2/m_x^3}} \exp\left(-\frac{(t_j-QT/m_x)^2}{2\sigma_x^2 QT^2/m_x^3}\right). \end{aligned}$$

Её логарифм имеет вид

$$\begin{aligned} -2 \ln L = & \sum_{i=1}^N \frac{(x_i-m_x)^2}{\sigma_x^2} + \sum_{j=1}^M \frac{(t_j-QT/m_x)^2}{\sigma_x^2 QT^2/m_x^3} + \\ & + (M+N) \ln \sigma_x^2 - M \ln m_x^3 + \text{const}. \end{aligned}$$

Оценки параметров m_x и σ_x^2 следует искать из системы уравнений

$$\begin{aligned} & \frac{\partial(-2 \ln L)}{\partial(\sigma_x^2)} = -\sum_{i=1}^N \frac{(x_i-m_x)^2}{(\sigma_x^2)^2} - \\ & - \sum_{j=1}^M \frac{(t_j-QT/m_x)^2}{(\sigma_x^2)^2 QT^2/m_x^3} + \frac{N+M}{\sigma_x^2} = 0, \\ & \frac{\partial(-2 \ln L)}{\partial m_x} = -2 \sum_{i=1}^N \frac{x_i-m_x}{\sigma_x^2} + 2 \frac{QT}{m_x^2} \sum_{j=1}^M \frac{t_j-QT/m_x}{\sigma_x^2 QT^2/m_x^3} + \\ & + 3m_x^2 \sum_{j=1}^M \frac{(t_j-QT/m_x)^2}{\sigma_x^2 QT^2} - \frac{3M}{m_x} = 0. \end{aligned} \quad (11)$$

Решение систем (10) и (11) возможно лишь численно.

Заключение

Итак, вопрос управления запасами опционного сырья в рамках рассматриваемой модели теоретически исследован достаточно полно. Полученные результаты носят асимптотический характер, поэтому было бы интересно посмотреть на результаты имитационного моделирования, чтобы оценить практическую скорость сходимости к нормальным законам распределения.

Заметим также, что в модели есть еще один частично настраиваемый параметр: величина c , связанная с дополнительной прибылью предприятия при использовании опционного сырья. Следует, однако, учитывать, что, вообще говоря, величины m_T и σ_T^2 также должны зависеть от c .

Оптимизация средней прибыли предприятия по параметру c и имитационное моделирование могут составить дальнейшее направление работы.

СПИСОК ЛИТЕРАТУРЫ

1. Arrow K.J., Harris Th.E., Marschak J. Optimal Inventory Policy // *Econometrica*. – 1951. – V. 19. – № 3. – P. 250–272.
2. Dvoretzky A., Kiefer J., Wolfowitz J. On the optimal character of the (S, s) policy in inventory theory // *Econometrica*. – 1953. – V. 21. – № 4. – P. 586–596.
3. Chikán A. Inventory models. – Dordrecht: Kluwer Academic Publishers, 1990. – 418 p.
4. Ross Sh.M. Applied probability models with optimization applications. – N.Y.: Dover Publications, 1992. – 198 p.
5. Chopra S., Meindl P. Supply chain management. – London: Prentice Hall, 2001. – 534 p.
6. Beyer D., Cheng F., Sethi S.P., Taksar M. Markovian demand inventory models. – N.Y.: Springer, 2010. – V. 108. – 255 p.
7. Лопухова С.В., Назаров А.А. Исследование рекуррентного потока // *Вестник ТГУ. УВТиИ*. – 2007. – № 1. – С. 67–70.
8. Лопухова С.В., Назаров А.А. Исследование МАР потока методом асимптотического анализа N -го порядка // *Вестник ТГУ*. – 2006. – № 293. – С. 110–115.
9. Диткин В.А., Прудников А.П. Интегральное преобразование и операционное исчисление. – М.: Наука, 1974. – 544 с.

Поступила 11.04.2013 г.

УДК 519.6

ДИАЛоговая ПРОЦЕДУРА РЕШЕНИЯ ЗАДАЧ ВЕКТОРНОЙ ОПТИМИЗАЦИИ С НЕОПРЕДЕЛЕННЫМИ ПАРАМЕТРАМИ

В.И. Рейзлин, В.А. Орлов*

Томский политехнический университет

*ООО «Оптимальное решение», г. Томск

E-mail: vir@tpu.ru

Рассматривается методика решения задач многокритериальной оптимизации с математическими моделями, содержащими множество переменных, значения которых не регулируются лицом, принимающим решение. Вводится понятие толерантности варианта решения, родственное понятиям устойчивости, живучести и т. п.

Ключевые слова:

Многокритериальная оптимизация, критерии оптимальности, оптимальность по Парето, лицо, принимающее решение.

Key words:

Multicriteria optimization, optimality criteria, Pareto optimality, decision-maker.

Любой ситуации принятия решения присущи следующие общие элементы:

- а) Множество переменных, значения которых выбираются лицом, принимающим решение (далее – ЛПР). Будем называть такие переменные **вариантами решения** или просто вариантами.
- б) Множество переменных, значения которых либо по желанию, либо по необходимости не регулируются ЛПР. Будем называть такие переменные **условиями**.
- в) Способ оценивания качества вариантов решения при каждом из условий. Обычно это одна или несколько вещественнозначных функций, зависящих от вариантов и условий.

Перейдем к математической формулировке задачи.

В данной работе предполагается, что варианты решения описываются n -мерными вещественными векторами $x = \langle x_1, \dots, x_n \rangle \in R^n$, условия – также векторные величины $p = \langle p_1, \dots, p_n \rangle \in R^m$. Компоненты векторов x назовем **конструктивными параметрами**, а компоненты векторов p – **внешними параметрами**. Пару векторов $\langle x, p \rangle$ будем называть **ситуацией**, т. е. ситуация – это вариант решения x при условиях p .

Будем предполагать также, что заданы **параметрические ограничения**

$$x_i^{\wedge} \leq x_i \leq x_i^{\vee} \text{ для всех } i \text{ от } 1 \text{ до } n,$$

$$p_i^{\wedge} \leq p_i \leq p_i^{\vee} \text{ для всех } i \text{ от } 1 \text{ до } m,$$

определяющие в пространстве параметров $R^n \times R^m$ прямоугольный блок $B = X \times P$, который будем называть **исходным блоком**, а блоки X и P , соответственно, блоком вариантов и блоком параметров.

Кроме параметрических ограничений рассмотрим функциональные ограничения $0 \leq g_i(x, p) \leq g_i^{\vee}$, $1 \leq i \leq r$, вырезающие в исходном блоке некоторую криволинейную область G , которую назовем **областью возможных ситуаций**. Проекцию области G в пространство конструктивных параметров обозначим через $G|X$. Множество $G|X$ – это множество альтернативных вариантов решения, из которых приходится делать выбор.

Функции $g(x, p)$ предполагаются непрерывными. Определяемая ими область G , вообще говоря, может быть любым замкнутым множеством. Единственное ограничение – ее объем не должен равняться нулю. С математической точки зрения требования, предъявляемые к области G , должны быть более жесткими: область G должна совпадать с замыканием множества своих внутренних точек.

Именно это требование обеспечивает отсутствие в G областей меньшей размерности.

Однако с практической точки зрения последнее требование излишне, т. к. такого положения всегда можно добиться сколь угодно малым изменением величин g_i^v .

Наконец, качество ситуации **вариант + условия** будем оценивать с помощью **целевых функций (критериев)** $f_i(x, p)$, $1 \leq i \leq s$, заданных в блоке B или в некоторой объемлющей его области. Иными словами, при прочих равных условиях ситуация тем лучше, чем больше целевая функция.

Базовым понятием для многокритериальных задач является оптимальность по Парето. Ситуация $\langle x, p \rangle$ считается оптимальной по Парето, если не существует другой ситуации $\langle x^*, p^* \rangle$, такой, что $f_i(x^*, p^*) \geq f_i(x, p)$ для всех i , и хотя бы для одной целевой функции это неравенство является строгим.

Однако цель нашего исследования состоит не в поиске оптимальных ситуаций, хотя и они, конечно, небезынтересны. Нам нужно найти варианты решения, который порождает оптимальные ситуации на широком, в идеале на всем, множестве условий.

Итак, имеем следующую задачу: **в области G нужно выбрать «лучший» вектор x , определяющий выбор варианта, качество которого каким-либо образом согласуется с набором целевых функций.**

Остается понять, что такое «лучшие» векторы.

Рассмотрим сначала упрощенный вариант задачи, когда внешние параметры отсутствуют. Этот вариант важен и сам по себе, и потому, что к нему тем или иным способом сводится решение общей задачи.

Существующие методы решения задач многокритериальной оптимизации можно с некоторой степенью условности разделить на автоматические и диалоговые.

Автоматические методы. Эти методы предусматривают получение в ходе их выполнения одной единственной компромиссной точки (точки множества Парето), которая и считается оптимальной. Для получения этой точки в подавляющем большинстве автоматических методов задача многокритериальной оптимизации преобразуется в задачу с одним критерием эффективности. Наиболее распространенными методами такого типа являются методы главного критерия, обобщенного критерия и целевого программирования.

Метод главного (основного) критерия. Исходная задача многокритериальной оптимизации сводится к задаче оптимизации по одному из критериев $f_k(x)$, который считается самым важным (главным), при условии, что значения остальных критериев должны быть не меньше установленных величин f_i^* [1, 2].

Методы обобщенного критерия и целевого программирования. Эти методы близки друг другу. Сущность их состоит в том, что все критерии $f_i(x)$ свертываются в одну функцию, называемую обобщенным критерием или функцией ценности [3].

В этих методах ЛПР выбирает не конкретное решение, а обобщенный критерий, на основании которого будет выбрано оптимальное решение. В результате этого полученное решение считается оптимальным в смысле выбранного обобщенного критерия. Здесь ЛПР абстрагируется от непосредственных значений критериев и оперирует только с их нормализованными значениями, используя весовые коэффициенты, определяющие важность критерия.

Автоматические методы хороши тем, что дают одно единственное решение. Однако эти методы обладают существенными недостатками. Для метода главного критерия это необходимость:

- выделить из множества критериев наиболее важный, что возможно не во всех практических задачах;
- установить ограничения для всех критериев, кроме главного; установка этих ограничений затруднительна для ЛПР, если они объективно не связаны с постановкой задачи, кроме того, задание ЛПР больших граничных значений критериев часто дает идеальную, т. е. несуществующую точку. Задание малых значений для ограничений приводит задачу многокритериальной оптимизации к тривиальной задаче максимизации одного из критериев, что противоречит самой постановке задачи многокритериальной оптимизации.

Методы обобщенного критерия и целевого программирования создают иллюзию получения единственного оптимального решения, хотя ЛПР, как указано выше, даже не оперирует непосредственными значениями функций, а лишь нормализованными. Доказать правильность выбора того или иного обобщенного критерия, а тем более обосновать, почему полученное решение лучше всех остальных, отвлекаясь от выбранного обобщенного критерия, невозможно.

Автоматические методы появились от неверия в возможность ЛПР самому выбрать нужное ему компромиссное решение. Выбор главного или обобщенного критерия, вообще говоря, произволен. «Исследователю не нравится произвол в выборе решения; считая математизацию за всеобщее благо, он постулирует постановку задачи и целевую функцию, а дальше все идет в соответствии с канонами математической науки, т. е. точно решаются задачи, произвольно поставленные. Произвол переносится из области выбора решения в область выбора математической модели и целевой функции. А этот произвол еще хуже, так как создает видимость научного обоснования там, где его по существу нет» [4. С. 93].

Недостатки автоматических методов все больше убеждают в необходимости привлечения человека к принятию решений. Поэтому в последнее время в большинстве практических работ, использующих многокритериальную оптимизацию, применяются диалоговые методы.

Интерактивные методы. Сущность всех интерактивных методов одна: ЛПР осуществляет направленный перебор компромиссных точек с целью выбора точки, наилучшим образом удовлетворяющей его предпочтениям. Для получения компромиссных точек на каждом шаге процедуры используются уже рассмотренные выше автоматические методы. ЛПР получает компромиссную точку, оценивает ее, и на основании этой оценки ищется другая компромиссная точка. Процедура заканчивается, когда одна из точек удовлетворяет ЛПР. Эта точка и считается оптимальной. Интерактивные методы постоянно развиваются и совершенствуются в направлении облегчения ЛПР принятия решения, упрощения формы получаемой от ЛПР информации. Однако анализ существующих интерактивных методов показал наличие парадокса, связанного с предположениями о знаниях ЛПР: с одной стороны, ЛПР само выбирает лучшую компромиссную точку, т. е. обладает достаточными знаниями для ее выбора, а с другой стороны, предполагается, что ЛПР не в состоянии четко указать, какое же решение оно желает получить [5].

В работах [5, 6] предлагаются методы решения задачи многокритериальной оптимизации, которые требуют от ЛПР задания желаемых значений целевых функций и в случае отсутствия компромиссных точек, удовлетворяющих заданным требованиям, предписывают ЛПР уточнять заданные значения критериев. На основании такого уточнения область компромиссных точек сужается до тех пор, пока не превратится в точку. Предлагаемые процедуры достаточно гибки, но требуют для своего исполнения решения очень большого количества однокритериальных экстремальных задач и поэтому эффективны лишь для задач с малым числом критериев.

В работах [7–9] предлагается интерактивный метод, основанный на зондировании пространства параметров, т. е. на замене исходной непрерывной области ее дискретным представлением в виде конечного равномерно расположенного в этой области множества точек, и на активном диалоге ЛПР с компьютером. Этот метод не свободен от недостатков, но обладает и весьма существенными достоинствами, которые можно распространить на решение задач с внешними параметрами.

Но прежде рассмотрим основные применяемые в практике способы снятия неопределенности в случае задач с внешними параметрами. Фактически их только два.

Минимаксный подход. Выбор считается оптимальным, если он оптимален для наиболее неблагоприятного случая. Другими словами из задачи исключаются внешние параметры заменой целевых функций $f_i(x, p)$ критериями $f_i^*(x) = \min_p f_i(x, p)$, а затем решается задача оптимизации в условиях определенности. Достоинство этого подхода заключается в том, что достигается гарантированный результат, правда лишь в том случае, когда решение существует. Этот подход очень негибок, требует

больших объемов вычислительной работы и, кроме того, вряд ли разумно планировать крайности, если это, конечно, не необходимо.

Статистический подход. Выбор считается оптимальным, если он оптимален в среднем. В основе этого подхода лежат представления о внешних параметрах как о случайных величинах с теми или иными вероятностными характеристиками. Неопределенность снимается заменой целевых функций их математическими ожиданиями, рассчитанными с использованием известных распределений внешних параметров. Однако для реализации статистического подхода необходимо построить распределение вероятностей внешних параметров, что не всегда легко, а часто и невозможно сделать.

В настоящей работе предлагается **прагматический подход**. Задачи многокритериальной оптимизации возникают и решаются не как чисто математические конструкции, а для удовлетворения вполне конкретных потребностей проектирования, планирования и т. п. Чтобы достичь практических результатов, часто достаточно найти вариант решения, имеющий, может быть, и не самые лучшие, но достаточно приемлемые качества, и сохраняющий их на возможно более широком множестве внешних факторов. Поэтому в каждой частной ситуации имеет вполне реальный смысл следующая конструкция.

Для каждой целевой функции $f(x, p)$ назовем пороговые значения – «худшие» показатели качества, с которыми ЛПР готово согласиться. Другими словами, мы назначаем **целевые** или **критериальные** ограничения. Нас будут устраивать ситуации $\langle x, p \rangle$, для которых выполнены все неравенства $f_i(x, p) \leq f_i^*$.

Множество $F = \{ \langle x, p \rangle \in G : f_i(x, p) \leq f_i^* \}$ назовем областью эффективных ситуаций или просто **областью эффективности**.

Введем новое понятие, в некотором смысле родственное таким понятиям как живучесть, устойчивость, нечувствительность и т. п. В связи с тем, что перечисленные термины уже заняты и широко применяются, мы будем употреблять слово толерантность, используя те оттенки его значения, которые означают терпимость, способность соглашаться, безразличие.

Пусть M некоторое множество с ненулевым объемом, лежащее в исходном блоке $B = X \times P$, x – вариант решения.

Множество $P(x, M) = \{ p \in P : \langle x, p \rangle \in M \}$ назовем **множеством толерантности** варианта x на множестве M .

Отношение объема множества $P(x, M)$ к объему блока P $T(x, M) = \frac{|P(x, M)|}{|P|}$ назовем **толерантностью** варианта x на множестве M .

Толерантность характеризует способность варианта x сохранять ситуацию в пределах множества M при изменении условий, или, другими словами, это показатель устойчивости варианта к изменениям условий.

Применительно к нашей задаче толерантность $T(x, F)$ — это характеристика способности варианта x сохранять значения целевых функций внутри исходной области G в пределах, заданных вектором $f^{\wedge} = \langle f_1^{\wedge}, \dots, f_s^{\wedge} \rangle$. Вариант с наибольшей толерантностью и есть прагматическое решение оптимизационной задачи.

Очевидно, что вариант решения с толерантностью, равной 1, является приближенно оптимальным для минимаксного подхода. Со статистической точки зрения толерантность можно интерпретировать, как вероятность варианта удерживать ситуацию в пределах допустимой области. Таким образом, понятие толерантности предоставляет возможность обобщить и минимаксный, и стохастический подход к решению оптимизационных задач с неопределенными параметрами.

Заметим, что так как множество F является частью области G , то толерантность $T(x, F)$ является функцией не только вектора x , но и векторов $f^{\wedge}$ и $g^{\vee}$, определяющих соответственно критериальные и функциональные ограничения.

Введенное понятие толерантности, как легко показать, обладает следующими свойствами.

Пусть функции $f_i(x, p)$ и $g_k(x, p)$ непрерывны в области определения. Тогда функция $T(x, f^{\wedge}, g^{\vee})$:

- по переменной x — кусочно-непрерывная,
- по переменной $f^{\wedge}$ — монотонно убывающая и кусочно-непрерывная,
- по переменной $g^{\vee}$ — неубывающая и кусочно-непрерывная;
- непрерывная, если все функции $f_i(x, p)$ и $g_k(x, p)$ не равны константе на множестве ненулевого объема, т. е. не имеют областей постоянства.

К сожалению, вычисление толерантности для различных вариантов решений аналитическими методами возможно только для областей, имеющих простое геометрическое строение, определяемое алгебраическими уравнениями первого или второго порядков. А ведь требуется еще и найти вариант с наибольшей толерантностью. С формальной точки зрения имеем две различные задачи: задачу вычисления меры множеств и задачу поиска оптимума. И ту и другую задачи можно пытаться решать, используя достаточно хорошие равномерно распределенные последовательности [10, 11].

Будем полагать, что мы умеем строить в блоках X и P начальные участки некоторых равномерно распределенных последовательностей $\{X_k\}$ и $\{P_k\}$.

Пусть A — некоторое совершенное (совпадающее со своим замыканием) подмножество блока P , а $P_1, P_2, \dots, P_N$ — начальные точки последовательности $\{P_k\}$, — множество тех из них, которые попали в

$$A, \text{ тогда } |A| = \lim_{N \rightarrow \infty} \frac{|A_N|}{N} |P|.$$

Это известное положение теории равномерно распределенных последовательностей. При доста-

$$\text{точно больших } N \quad |A| \approx \frac{|A_N|}{N} |P|.$$

Мы будем использовать последнее приближение для вычисления толерантности.

Возможны три принципиально различных случая постановки общей задачи многокритериальной оптимизации

1. Ни целевые функции, ни ограничения не зависят от условий.
2. Часть целевых функций зависит от условий, а часть не зависит.
3. Все целевые функции зависят от условий.

Рассмотрим подробно каждый вариант и предложим схему решения соответствующей задачи.

Случай 1. Это классическая задача оптимизации в условиях определенности. Приведем методику ее решения, предложенную в [5].

Во множестве G , совпадающем с областью возможных вариантов, выбираем N пробных точек $G_1, \dots, G_N$ из некоторой равномерно распределенной в X последовательности. В каждой пробной точке вычисляются значения всех критериев $f_1, \dots, f_N$. По каждому критерию составляется таблица испытаний, в которой значения расположены в порядке возрастания качества, т. е. худшие значения предшествуют лучшим.

При больших N крайние значения таблицы испытаний для каждого кусочно-непрерывного критерия близки к его минимуму и максимуму. Таблица испытаний позволяет оценивать не только экстремальные значения критериев, но и дает возможность судить о частоте тех или иных значений.

Просматривая таблицы испытаний, ЛПР назначает пороговые значения критериев, т. е. задает вектор $f^{\wedge}$. После того как пороги назначены, происходит отбор эффективных точек — точек, удовлетворяющих критериальным ограничениям. Может оказаться, что для заданного N не найдется ни одной эффективной точки. Это означает, что если множество эффективных вариантов не пусто, то его объем имеет порядок $|G|/N$. В этой ситуации можно поступить двояко: либо изменить вектор $f^{\wedge}$, ослабив какие-либо ограничения, либо, если ограничения менять не хочется, увеличить число пробных точек. Если при неоднократном увеличении числа N множество эффективных вариантов все же пусто, то есть основание полагать, что ограничения несовместны. Конечно, нельзя исключить, что существует эффективная точка, но если это и так, то ее окрестность, в которой действуют все ограничения, имеет очень маленький объем, и практически соответствующий этой точке вариант решения будет неустойчивым, т. е. небольшие нарушения допусков приведут к потере эффективности.

После того как найдено некоторое множество эффективных вариантов, ЛПР может удовлетвориться достигнутым, считая любой найденный вариант решением задачи, но может пойти и дальше, выбирая в построенном множестве точки, оптимальные по Парето.

В табл. 1 рассматривается схема решения задач первого типа. Слева приведены операции, справа — оперирующая сторона.

Таблица 1. Схема решения задач 1-го типа

1. Назначаем N – число пробных точек	ЛПР
2. Вычисляем значения f и g в пробных точках	Компьютер
3. Назначаем ограничения $f^{\wedge}$ и $g^{\vee}$	ЛПР
4. Находим множество F_N эффективных вариантов	Компьютер
5. Если F_N имеет малый объем, переходим к шагу 1 или шагу 3	ЛПР
6. В множестве F_N выбираем оптимальные по Парето варианты	Компьютер

Случай 2. Это наиболее общий случай задач многокритериальной оптимизации. Вектор целевых функций состоит из двух частей $f(x,p) = \langle \varphi(x), \psi(x,p) \rangle$.

В блоках X и P строим множества X_N и P_K пробных точек. Составляем таблицы испытаний для всех функций из f и g , рассчитывая их в точках множества $X_N \times P_K$. Назначаем критериальные и функциональные ограничения. Для критериев, не зависящих от условий, находим эффективные варианты, выбираем из них оптимальные по Парето и уже для последних рассчитываем толерантности. Вариант с наибольшей толерантностью и будет являться решением.

В табл. 2 рассматривается схема решения задач второго типа. Слева приведены операции, справа – оперирующая сторона.

Таблица 2. Схема решения задач 2-го типа

1. Назначаем N и K – число пробных точек в блоках X и P	ЛПР
2. Вычисляем значения f и g в пробных точках	Компьютер
3. Назначаем ограничения $f^{\wedge}$ и $g^{\vee}$	ЛПР
4. Находим множество Φ_N эффективных вариантов	Компьютер
5. Если Φ_N имеет малый объем, переходим к шагу 1 или шагу 3	ЛПР
6. В множестве Φ_N выбираем φ -оптимальные решения	Компьютер
7. Среди них находим вариант с наибольшей толерантностью	Компьютер

Случай 3. Это другой крайний случай, в каком-то смысле противоположный первому. Здесь все целевые функции зависят от условий.

В блоках X и P строим множества X_N и P_K пробных точек. Составляем таблицы испытаний для всех функций из f и g , рассчитывая их в точках множества $X_N \times P_K$, и назначаем критериальные и функциональные ограничения. Для каждого варианта из области эффективных вариантов рассчитываем толерантности. Вариант с наибольшей толерантностью и будет являться решением.

В табл. 3 рассматривается схема решения задач третьего типа.

Для проверки работоспособности предлагаемого прагматического подхода к диалоговому процессу многокритериальной оптимизации были проведены численные эксперименты на моделях, предложенных в [8, 9]. Прагматический подход сравнивался с интерактивным методом, предложенным в этих же работах. В [8, 9] строится однокритериаль-

ная целевая функция, основанная на получении информации о предпочтениях ЛПР посредством диалога. В качестве автоматического метода, оптимизирующего эту функцию, используется метод деформируемых конфигураций [8]. В прагматическом подходе применялась рассмотренная выше схема 3 и квазиравномерные последовательности точек [11]. В обоих случаях использовалась размерность задачи 10, а число обращений к целевой функции ограничивалось 1000.

Таблица 3. Схема решения задач 3-го типа

1. Назначаем N и K – число пробных точек в блоках X и P	ЛПР
2. Вычисляем значения f и g в пробных точках	Компьютер
3. Назначаем ограничения $f^{\wedge}$ и $g^{\vee}$	ЛПР
4. Находим множество F_N эффективных вариантов	Компьютер
5. Если F_N имеет малый объем, переходим к шагу 1 или шагу 3	ЛПР
6. В множестве F_N находим вариант с наибольшей толерантностью	Компьютер

Временные затраты, связанные с вычислением компьютером целевой функции в прагматическом подходе, оказываются в 4–7 раз выше, чем в подходе [9]. Это связано с тем, что метод деформируемых конфигураций является более быстрым по сравнению со стохастическим. Однако эти затраты несравнимо меньше, чем время диалога с ЛПР. Кроме того, в прагматическом подходе ЛПР вступает в диалог только после серии итераций, в то время как в методе [9] диалог требуется после каждой итерации. В результате прагматический подход оказывается в 2–3 раза более эффективным (эти оценки зависят в общем случае от размерности задачи и числа критериев).

Однако более важной при сравнении методов является как раз толерантность найденного решения. В подходе [8] погрешность нахождения оптимума в модельных задачах не превосходит 4 %, в то время как в прагматическом подходе – 12 %. В то же время проведенный анализ чувствительности найденных решений показал, что подход [8] приводит к очень узкой области работоспособности (порядка 3...4 % от диапазона параметров), в то время как прагматический подход дает широкий коридор в 10...15 %, что приводит к высокой устойчивости найденных решений.

Итак, в настоящей работе приводится методика решения задач многокритериальной оптимизации с математическими моделями, содержащими множество переменных, значения которых не регулируются лицом, принимающим решение. Эта методика основана на вновь введенном понятии толерантности варианта решения. Приведены схемы решения задач 3-х разных типов. Результаты исследований показывают работоспособность прагматического подхода, его приемлемые точность и эффективность, и, главное, устойчивость решений к изменениям найденных оптимальных параметров

Работа выполнена в рамках госзадания «Наука», рег. номер НИР 8.596.2011.

СПИСОК ЛИТЕРАТУРЫ

1. Подиновский В.В. Математическая теория выработки решений в сложных ситуациях. — М.: Министерство обороны СССР, 1981. — 212 с.
2. Дубов Ю.А., Травкин С.И., Якимец В.Н. Многокритериальные модели формирования и выбора вариантов систем. — М.: Наука, 1986. — 296 с.
3. Машунин Ю.К. Методы и модели векторной оптимизации. — М.: Наука, 1986. — 144 с.
4. Вентцель Е.С. Исследование операций. Методологические аспекты. — М.: Наука, 1972. — 105 с.
5. Степанов А.В. Человеко-машинная процедура принятия решений в задачах векторной оптимизации // Математическое моделирование. — 1991. — Т. 3. — № 5. — С. 61–73.
6. Соболев И.М., Статников Р.Б. Выбор оптимальных параметров в задачах со многими критериями. — М.: Наука, 1981. — 112 с.
7. Михалевич В.С., Волкович В.Л. Вычислительные методы исследования и проектирования сложных систем. — М.: Наука, 1982. — 278 с.
8. Рыков А.С. О диалоговых методах деформируемых конфигураций // Доклады РАН. — 2000. — Т. 375. — № 2. — С. 39–46.
9. Рыков А.С., Калашников А.Е. Диалоговый метод деформируемых конфигураций для многокритериальной оптимизации технологических процессов // Современные сложные системы управления (СССУ/HTCS 2003): Труды Междунар. конф. — Воронеж: ВГАСУ, 2003. — Т. 2. — С. 185–188.
10. Кейперс Л., Нидеррайтер Г. Равномерное распределение последовательностей. — М.: Наука, 1985. — 408 с.
11. Орлов В.А., Рейзлин В.И. Новое семейство квазислучайных последовательностей // Известия Томского политехнического университета. — 2012. — Т. 320. — № 2. — С. 24–26.

Поступила 14.03.2013 г.

УДК 519.6:004.93

АНАЛИЗ ПУТЕЙ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ РАСЧЕТОВ ЧАСТОТНО-ВРЕМЕННОЙ КОРРЕЛЯЦИОННОЙ ФУНКЦИИ

Е.Е. Лунева, В.С. Аврамчук

Томский политехнический университет

E-mail: lee@tpu.ru, avs@tpu.ru

Рассмотрены технологии повышения эффективности математических расчетов на многопроцессорных системах с использованием графических процессоров NVIDIA CUDA. Показано, что графические процессоры превосходят по скорости процессоры общего назначения при решении задач, связанных с вычислениями. Эффективность использования технологии Microsoft.NET Framework целиком зависит от числа вычислительных ядер в процессоре. Применение технологии CUDA позволяет привлечь уникальную вычислительную архитектуру графических процессоров и существенно сократить общее время выполнения расчетов.

Ключевые слова:

Цифровой сигнал, корреляционный анализ, Microsoft.NET Framework, NVIDIA CUDA.

Key words:

Digital signal, correlation analysis, Microsoft.NET Framework, NVIDIA CUDA.

Цифровая обработка сигналов в настоящее время является важным и перспективным направлением развития современной науки и техники. По мере развития вычислительных мощностей современных ЭВМ увеличиваются объемы обрабатываемой информации, создаются совершенно новые алгоритмы обработки сигналов, расширяются сферы применения цифровой обработки сигналов. К создаваемым способам и программному обеспечению предъявляются более жесткие требования по быстродействию, возможности использования в режиме реального времени и точности. Удовлетворение этих требований невозможно без привлечения передовых вычислительных устройств и технологий. Цель данной работы заключается в исследовании возможных путей повышения эффективности использования аппаратных ресурсов ЭВМ при решении задач корреляционного анализа сигналов.

Корреляционный анализ сигналов находит широкое применение при решении задач неразрушающего контроля и диагностики, определения

характеристик электрических систем, обработки цифровых изображений [1]. Функции корреляции достаточно просто определяются через дискретное преобразование Фурье (ДПФ) [1]. Эффективность вычисления в данном случае зависит от способа реализации ДПФ. Максимальная эффективность достигается при использовании быстрого преобразования Фурье (БПФ). Основным недостатком такого способа расчета корреляционных функций является отсутствие информации о связи сигналов в частотной области. Этого недостатка лишен метод частотно-временного корреляционного анализа [2], использование которого позволяет существенно повысить информативность проводимого анализа. Однако применение данного подхода сопряжено с большими вычислительными затратами, так как данный метод вычисления требует многократного выполнения процедур БПФ. Количество необходимых преобразований напрямую зависит от количества формируемых копий сигнала. Для оценки трудоемкости вычислений указанным

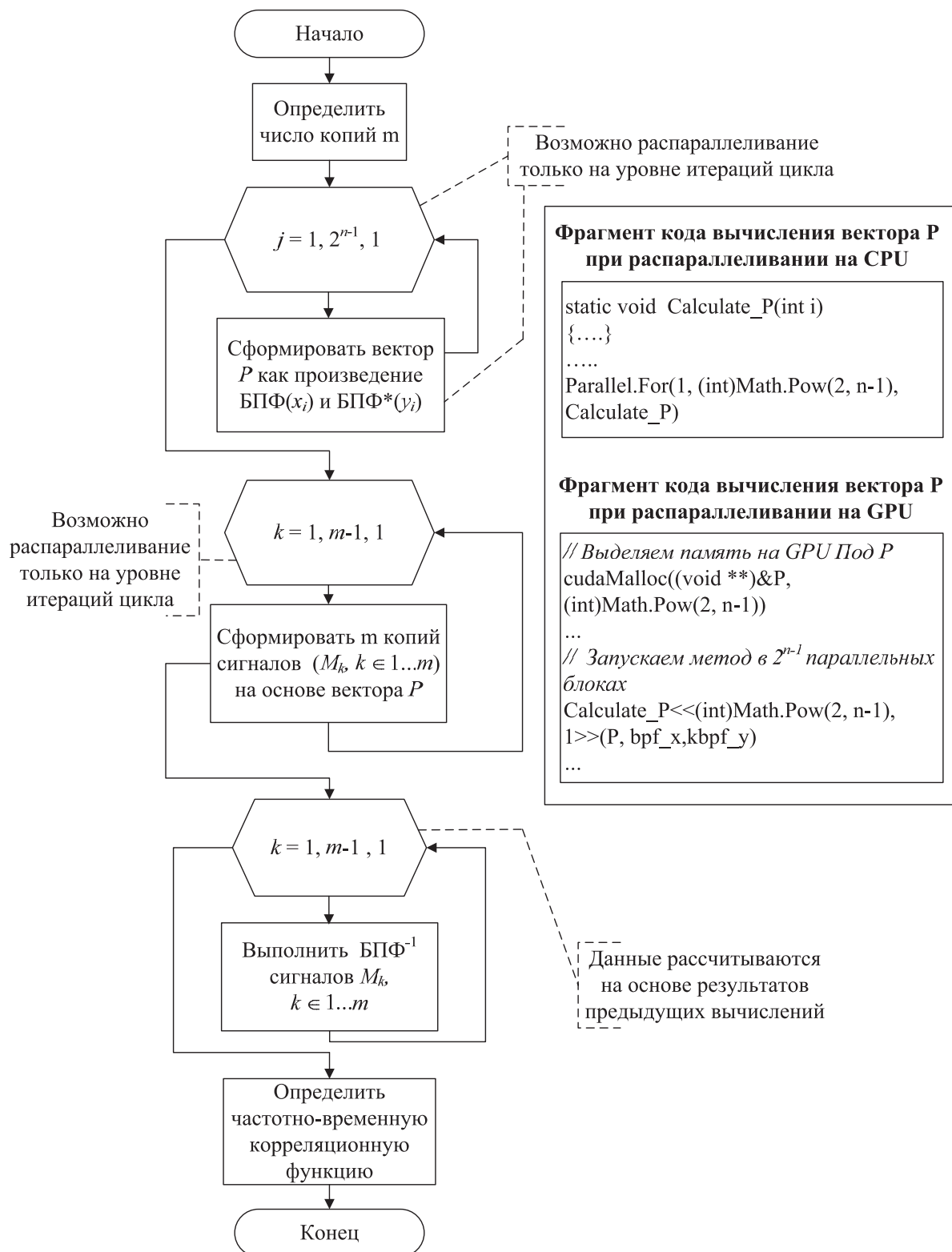


Рисунок. Алгоритм вычисления частотно-временной корреляционной функции: m – число формируемых копий сигнала; x_i, y_i – дискретные отсчеты сигналов; БПФ* – комплексно-сопряженное значение результатов прямого ДПФ; БПФ⁻¹ – обратное ДПФ; P – вектор произведения результатов прямого ДПФ сигнала x_i с комплексно-сопряженными значениями прямого ДПФ сигнала y_i

методом на рисунке приведен укрупненный алгоритм необходимых математических расчетов.

Несмотря на хорошую алгоритмируемость и наличие большого числа оптимизированных библиотек вычисления БПФ, отмеченная выше проблема является узким местом при использовании данного подхода и создаваемого на его основе программного обеспечения. В этом случае эффективность создаваемых программ зависит от использования специализированных технологий.

Для многопроцессорных систем, в которых все вычисления выполняются исключительно на центральном процессоре (CPU), повысить эффективность можно с использованием технологий распараллеливания вычислительных процессов. Корпорацией Microsoft разработан универсальный набор инструментов параллельного выполнения заданий ParallelExtensions, входящих в состав набора Microsoft.NET Framework 4.0 [3]. Данный инструментальный позволяет автоматически использовать все доступные процессоры для выделенных блоков кода, пригодных для распараллеливания, в существующем последовательном коде. Разметка задач осуществляется посредством вызова предназначенных для этого статических методов класса Parallel, являющегося частью инструмента ParallelExtensions. В частности, методы «Parallel.For ()», «Parallel.ForEach ()» преобразуют обычный последовательно исполняемый цикл в параллельный. Произвольные участки кода размечаются путем оформления их в отдельные методы и последующего вызова с помощью метода «Parallel.Invoke ()».

Однако сложность разработки приложений на основе инструментария ParallelExtensions увеличивается при распараллеливании задач с использованием одних и тех же данных, что приводит к необходимости детального управления распределением массивов и витков циклов между потоками, а также обменом сообщениями между ними. Например, это возможно при разработке наиболее популярного типа приложений «WindowsForms», при обработке данных в фоновых потоках и графической визуализации этих данных в основном потоке. Также использование универсального инструментария ParallelExtensions не позволяет максимально эффективно использовать преимущества конкретной параллельной архитектуры. Кроме этого, учитывая, что в настоящее время повышение производительности центральных процессоров затруднено фундаментальными ограничениями при производстве интегральных схем, следует рассматривать и другие решения повышения вычислительной мощности системы.

В литературе [4] отмечено, что приложения, использующие графические процессоры (GPU) NVIDIA для неграфических вычислений, продемонстрировали существенный прирост эффективности вычислений, по сравнению с реализациями, построенными на базе одних лишь центральных процессоров. Технология CUDA (Compute Unified Device Architecture) представляет собой програм-

мно-аппаратную архитектуру, позволяющую применять графический процессор компании NVIDIA для неграфических расчетов на основе Runtime API.

Runtime API представляет собой расширение к языку C/C++. Данное расширение позволяет выделить память на GPU при помощи функции «cudaMalloc ()», передать GPU на обработку участки кода (функция «cudaMemcpy ()»), оформленные как функции. Для параллельной обработки массивов данных может быть использован механизм блоков, позволяющий параллельно обрабатывать элементы массивов. Использование типа «dim3» CUDA позволяет работать с сетками – параллельными потоками для обработки матриц. Значимым достоинством CUDA является возможность прямого обращения к OpenGL и DirectX [4], что, несомненно, необходимо использовать в задачах с выводом изображений.

На основе рассмотренных технологий было создано программное обеспечение расчета частотно-временных корреляционных функций. В качестве алгоритма вычисления БПФ был выбран наиболее распространенный алгоритм Кули–Тьюки с фиксированным основанием 2, обладающий простотой реализацией, наглядностью и поддающийся эффективному распараллеливанию [5]. Сравнение эффективности рассмотренных технологий было проверено на ряде тестовых примеров. Результаты экспериментов были получены на размерах выборок: 2048, 4096, 8192, 16384, 32768 отсчетов. Количество формируемых копий $m=1121$. Результаты проведенных экспериментов сведены в таблицу. Максимальное время выполнения преобразований было получено на процессоре без реализации задач параллелизма вычислений. Для наглядности оценки эффективности применения рассмотренных технологий в таблице приведено отношение затраченного времени выполнения расчетов к максимальному времени вычислений, полученному на фиксированном размере выборки, выраженное в процентах.

Таблица. Результаты экспериментов

Размер выборки, отсчет	CPU Intel Celeron E1500 2 Core без Framework, %	CPU Intel Celeron E1500 2 Core + Framework, %	CUDA NVIDIA GeForce GTX640, %
2048	100	67	3,2
4096	100	61	2,9
8192	100	54	2,7
16384	100	52	2,5
32768	100	51	2,4

Исходя из полученных результатов можно констатировать, что применение технологии Microsoft.NET Framework на двухъядерном процессоре позволило сократить время выполнения преобразований. Максимальная эффективность была получена при длине выборки 32768 отсчета, а минимальная – при 2048. Это различие легко объясняет-

ся тем, что время создания потока может быть сопоставимо, а иногда и превосходить время выполнения преобразования. Параллельная обработка данных в соответствии с представленным на рисунке (фрагменты кода) алгоритмом может быть выполнена только на уровне итераций цикла, а также при вычислении прямого и обратного преобразования Фурье, что требует привлечения значительных ресурсных затрат CPU на создание и завершение потоков.

Использование технологии CUDA привело к существенному сокращению времени обработки сигнала за счет высоких скоростных характеристик создания сеток и блоков потоков.

Выводы

Рассмотрены технологии увеличения эффективности проведения расчетов, позволяющих вычислить частотно-временную корреляционную функцию. За счет своей специализированной вычислительной архитектуры графические процессоры превосходят по скорости центральные процес-

соры общего назначения при решении задач, связанных с вычислениями.

Показано, что использование технологии Microsoft.NET Framework целиком зависит от числа вычислительных ядер в центральном процессоре. Предельное сокращение времени выполнения расчетов будет соответствовать количеству этих ядер, что подтверждается полученными результатами.

Применение технологии CUDA позволяет привлечь уникальную вычислительную архитектуру графических процессоров и существенно сократить общее время выполнения преобразований. Значительный эффект достигается при обработке большого количества данных, так как время создания потока может превосходить время выполнения преобразования. В то же время эффективность вычислений будет зависеть от конкретной реализации алгоритма быстрого преобразования Фурье.

Технология CUDA в настоящее время дает возможность создания эффективного программного обеспечения за счет своей уникальной вычислительной архитектуры.

СПИСОК ЛИТЕРАТУРЫ

1. Айфичер Э.С., Джервис Б.У. Цифровая обработка сигналов: практический подход. 2-е изд. – М.: Вильямс, 2008. – 992 с.
2. Аврамчук В.С., Чан Вьет Тъяу. Частотно-временной корреляционный анализ цифровых сигналов // Известия Томского политехнического университета. – 2009. – Т. 315. – № 5. – С. 112–115.
3. Рихтер Д. CLR via C#. Программирование на платформе Microsoft.NET Framework 4.0 на языке C#. 3-е изд. – СПб.: Питер, 2012. – 928 с.
4. Сандерс Д., Кэндрот Э. Технология CUDA в примерах: введение в программирование графических процессов. – М.: ДМК Пресс, 2011. – 232 с.
5. Нуссбаумер Г. Быстрое преобразование Фурье и алгоритмы вычисления свертки. – М.: Радио и связь, 1985. – 248 с.

Поступила 25.09.2013 г.

МОДЕЛЬ УГРОЗ СИСТЕМЫ ФИЗИЧЕСКОЙ ЗАЩИТЫ ЛИНЕЙНОЙ ЧАСТИ МАГИСТРАЛЬНОГО НЕФТЕПРОВОДА

И.А. Кукало, А.П. Кшнянкин, С.Н. Гривцов*

Томский государственный университет систем управления и радиоэлектроники

*ОАО «Центрсибнефтепровод», г. Томск

E-mail: i@kukalo.ru

Отмечена особенность управления безопасностью протяженных объектов большого, национального масштаба, типа линейной части магистрального нефтепровода. Предложена иерархическая модель угроз типовой линейной части магистрального нефтепровода. Приведена посуточная оценка вероятности реализации угроз его физической безопасности с применением метода, основанного на статистических данных для линейной части магистрального нефтепровода ОАО «Центрсибнефтепровод». Определены группы факторов, оказывающих влияние на указанную модель, а также весовые коэффициенты, учитывающие вклад каждого фактора. Дана иллюстративная оценка погонного распределения вероятности реализации угроз безопасности линейной части магистрального нефтепровода «Александровское–Анжеро-Судженск».

Ключевые слова:

Угрозы физической безопасности, магистральный нефтепровод.

Key words:

Physical security threats, oil pipeline.

Основные меры, обеспечивающие безопасность, в том числе антитеррористическую защищенность объектов топливно-энергетического комплекса (ТЭК) в Российской Федерации, устанавливаются Федеральным законом «О безопасности объектов топливно-энергетического комплекса» [1]. Типовым объектом такого рода является линейная часть магистрального нефтепровода (ЛЧ МН) – пространственно-протяженная, сложная структурная составная часть магистрального нефтепровода, предназначенная для транспортировки нефти и нефтепродуктов [2]. Указанная особенность инфраструктуры ЛЧ МН осложняет задачу обеспечения ее безопасности, не позволяя использовать для этой цели стандартные методики защиты, пригодные для точечных объектов.

Известные подходы к управлению безопасностью сложных и масштабных, таких как ТЭК, систем чаще всего основываются на процедуре категорирования опасных объектов. При этом объектам одной категории предъявляются единые требования по обеспечению безопасности – стандартная модель угроз и профиль защиты, т. е. типовой состав комплекса средств и мероприятий, обеспечивающий приемлемый уровень безопасности всего множества объектов данной категории [3]. Альтернативой категорированию, как известно, является процедура построения моделей угроз, оценка уязвимостей, построение моделей защиты и систем требований не только для типовых объектов каждой категории, но и по каждому объекту системы. Такой подход, основанный на исследовании модели актуальных угроз и модели событий рисков, при которых эти угрозы могут быть реализованы, и построении модели защиты объекта дан, например, в работе [4]. Однако предложенная авторами [4] итерационная модель управления рисками применима лишь к структурированным локализованным объектам и не может непосредственно применяться

для проектирования и оптимизации систем физической защиты (СФЗ) протяженных объектов типа ЛЧ МН.

Целью настоящей работы является создание модели угроз физической безопасности (ФБ) пространственно-протяженных объектов, которая, как известно, является основой для построения модели следующего уровня – общей модели управления рисками систем, в т. ч. таких, как СФЗ типа ЛЧ МН.

В основу описания модели объекта защиты положим подход авторов [4], предложенный ими для моделирования эффективности систем защиты критически важных объектов (КВО). Для этого обозначим совокупность объектов ЛЧ МН как системное множество S составляющих его компонентов O_j , где $i \in I^S$ – множество индексов всех компонентов S : $S = \{O_j\}$.

Компоненты системы S , с которыми не связаны какие-либо угрозы безопасности, в модель объекта защиты включаться не будут. Следуя далее [4], структурируем объект защиты и выделим J уровней иерархии его компонентов. Часть системы, относимую к j -му уровню иерархии, обозначим как S^j , $j \in J = \{1 \dots J\}$. При этом уровень системы в целом будет соответствовать 1-му уровню иерархии, самый низкий уровень иерархии будет J -м. В этом случае система может быть представлена в виде структурной модели:

$$S = \{O_{js}^j\},$$

где j – иерархические уровни компонентов системы S .

Множество объектов системы ЛЧ МН предлагаем структурировать в соответствии с данными табл. 1. Количество и состав уровней объектов системы определяется существующей организационно-правовой структурой транспортных организаций системы магистральных нефтепроводов ОАО

«АК «Транснефть», с учетом требований, сформированных службой безопасности ОАО «Центрсибнефтепровод» в процессе осуществления своей деятельности.

Таблица 1. Перечень иерархических уровней системы S ЛЧ МН

j	Наименование	Обозначение	Примеры объектов уровня
1	Система МН	$S = \{O_1^j\}$	Система МН ОАО «АК «Транснефть»
2	Подсистема МН	$S^2 = \{O_2^j\}$	МН ОАО «Центрсибнефтепровод»
3	МН	$S^3 = \{O_3^j\}$	МН «Александровское-Анжеро-Судженск»
4	Километровые участки МН	$S^4 = \{O_4^j\}$	132 километр МН «Александровское-Анжеро-Судженск» (включая запорную и иную арматуру, переходы через естественные и искусственные препятствия), вдольтрассовое обо

Отдельные компоненты, определенные на множестве $\{O_i^j\}$, могут принадлежать к разным иерархическим уровням, однако, следуя [4], будем считать, что если компонент находится на более высоком уровне иерархии, он может включать в себя компоненты более низкого иерархического уровня. То есть для любого объекта при $j < J$ будет существовать множество

$$O_{i^s}^j \supset \{O_{i^o}^{j+1}\}, i^o \in I_{i^s}^{O_i^j},$$

где $I(O_i^j)$ – множество индексов компонентов $(j+1)$ -го уровня иерархии, входящих в состав компонента O_i^j .

Сформируем модель угроз ФБ ЛЧ МН как математическое, описательное представление свойств или характеристик угроз, реализуемых при совершении актов незаконного вмешательства ЛЧ МН [5, 6]. Для этого указанную совокупность представим множеством Y^S , связанным с системой S и со всеми ее компонентами, как:

$$Y^S = \{Y^{O_i^j}\}.$$

Здесь каждому объекту O_i^j сопоставляется некоторое множество угроз $Y^{O_i^j}$, которое и представляет собой формальную модель угроз для рассматриваемой системы S . Источником угроз Y^S является множество злоумышленников, которое может быть представлено как:

$$N^S = \{N^{Y^{O_i^j}}\}.$$

Возможная модель злоумышленника представлена в табл. 2.

Рассмотрим один из возможных вариантов задания элементов множества Y^S для ЛЧ МН, представленный в табл. 3.

Ниже, наряду с данными табл. 3, в качестве основной характеристики элементов множества Y^S будем использовать вероятности реализации угроз $\bar{P}_{Y_i^S}$.

Таблица 2. Модель злоумышленника

N_i^S	Наименование злоумышленника	Описание злоумышленника
N_1^S	Группа преступных элементов	Лицо или группа лиц, существующая в течение определённого периода времени и действующая согласованно с целью совершения одного или нескольких актов незаконного вмешательства с тем, чтобы получить, прямо или косвенно, материальную или иную выгоду
N_2^S	Внутренние нарушители	Лицо или группа лиц, обладающих правом доступа на объект защиты и к материальным ценностям в силу выполнения служебных или иных обязанностей
N_3^S	Диверсионно-террористическая группа	Подразделение, используемое для совершения диверсионно-террористических актов, уничтожения или временного вывода из строя важнейших объектов топливно-энергетического комплекса

Таблица 3. Модель угроз безопасности системы S

Y_i^S	Наименование угрозы	Злоумышленники, реализующие угрозу	Описание угрозы
Y_1^S	Врезка в ЛЧ МН	N_1^S, N_2^S	Установка патрубка с задвижкой с последующим нарушением целостности трубопровода для несанкционированной кражи нефти
Y_2^S	Взлом вдольтрассового обслуживания и получение доступа к нефти ЛЧ МН	N_1^S, N_2^S	Взлом вантузов, камер приема-пуска средств очистки и диагностики, запорной арматуры и других технологических устройств с целью несанкционированной кражи нефти
Y_3^S	Установка взрывчатых устройств на ЛЧ МН	N_3^S	Реализация террористического акта с целью нанесения материального, экологического и политического ущерба объекту топливно-энергетического комплекса

В качестве количественных методов оценки вероятности событий рассматриваются следующие: детерминированные, статистические, детерминированно-статистические, вероятностные, статистико-вероятностные, детерминированно-вероятностные и др. [7]. В настоящее время в системе промышленной безопасности объектов топливно-энергетического комплекса наиболее распространенными являются статистические методы [2]. Для их применения производится обобщение статистической информации организаций, осуществляющих эксплуатацию ЛЧ МН, о частоте возникновения актов незаконного вмешательства.

Для определения вероятности реализации рисков событий ФБ на ЛЧ МН воспользуемся по-

казателем $\bar{\lambda}_{\text{МН}_i}^j$ – интенсивность инцидентов нарушения ФБ (год⁻¹) на объектах $S^3=\{O_i^3\}$ для угрозы Y_j^s .

Поскольку в практике планирования мероприятий по снижению вероятностей реализации угроз ОАО «Центрсибнефтепровод» в качестве единичного интервала времени используются одни сутки, то именно данную единицу будем использовать в дальнейшем при описании вероятности реализации угроз ФБ. В общем случае значение вероятности суточной реализации угроз будет определяться по формуле точечного определения вероятности [8], где общее количество испытаний – это количество дней в году, а число появления событий равно $\bar{\lambda}_{\text{МН}_i}^j$, поэтому:

$$\begin{cases} \bar{p}_{y_j^s} = 1, & \text{если } \bar{\lambda}_{x_i}^j > K, \\ \bar{p}_{y_j^s} = \frac{\bar{\lambda}_{x_i}^j}{K}, & \text{если } \bar{\lambda}_{x_i}^j \leq K, \end{cases} \quad (1)$$

где $K=365$ – количество дней в году; $x \in X$ – множество статистических показателей интенсивности реализации угроз; $j \in J, J$ – множество индексов угроз безопасности; $i \in I, I$ – множество индексов объектов, для которых определены статистические показатели; $t \in T, T$ – множество индексов месяцев в году.

Вероятность реализации угроз ФБ может быть определена на различных уровнях иерархии объекта защиты S . При этом чем ниже уровень иерархии, на котором будет оцениваться вероятность реализации угрозы, тем большее количество влияющих факторов эта вероятность может учитывать. Поэтому, учитывая иерархическую вложенность подсистем S рассматриваемой модели объекта, оценку вероятности реализации угроз ФБ для участка ЛЧ МН будем вычислять на основе значения $\bar{p}_{y_j^s}$.

Вероятность $\bar{p}_{y_j^s}$ реализации угрозы O_i^3 найдем по формуле полной вероятности [8], при условии появления одного из несовместимых событий врезки на произвольном километровом участке $O_i^4 - N_j, j \in J=\{1...L_o\}$, где L_o – длина ЛЧ МН O_i^3 (км).

Так как события $N_1, N_2, ..., N_n$ образуют полную группу, то:

$$p(N) = p(N_1) + p(N_2) + ... + p(N_n) = 1. \quad (2)$$

Заметим далее, что любой из объектов O_i^4 , независимо от его местоположения, диаметра трубы и т. д., позволит злоумышленнику получить одинаковую выгоду, поэтому вероятность выбора злоумышленником объектов O_i^4 при реализации угрозы носит случайный характер. На этом основании выражение для оценки $p(N_j, \text{ км})$ произвольного ЛЧ МН O_i^3 можно записать как:

$$p(N_j) = \frac{1}{L_o^3}. \quad (3)$$

Далее по формуле полной вероятности получим:

$$\begin{aligned} \bar{p}_{y_j^s} &= p(N_1)p_{N_1}(y_j^s) + p(N_2)p_{N_2}(y_j^s) + ... + \\ &+ p(N_n)p_{N_n}(y_j^s), \end{aligned} \quad (4)$$

где $p_{N_j}(y_j^s)$ – вероятность реализации угрозы врезки на конкретном километровом участке O_i^4 .

Для того чтобы учесть факторы, влияющие на вероятность реализации угрозы на том или ином километровом участке, введем коэффициенты k_{N_j} , определяющие степень опасности конкретного участка ЛЧ МН:

$$p_{N_j}(y_j^s) = k_{N_j} \bar{p}_{y_j^s}. \quad (5)$$

Для определения коэффициентов k_{N_j} в формуле (4) для каждого километрового участка ЛЧ МН воспользуемся методом балльной оценки факторов [2]. На первоначальном этапе определим группы факторов влияния $Гр_i$ с весовыми коэффициентами ρ_i , определяющими вероятность их влияния на реализацию угрозы ФБ. В пределах каждой группы $Гр_i$ определим факторы влияния в количестве J_i . Каждый фактор имеет буквенно-цифровое обозначение F_{ij} , где i – номер группы, j – номер фактора в группе. Относительный вклад фактора F_{ij} внутри своей группы в изменение вероятности актов незаконного вмешательства на рассматриваемом участке ЛЧ МН учитывается с помощью весового коэффициента (доли) q_{ij} .

Расчет коэффициента k_{N_j} производился нами с использованием балльной оценочной системы, при которой каждому значению фактора F_{ij} ставится в соответствие определенное, назначаемое на основании расчета или экспертной оценки, количество баллов B_{ij} (по 10-балльной шкале), отражающее интенсивность его влияния. При рассмотрении конкретного n -го участка трассы последовательно оценивается степень влияния каждого из факторов. Полученные для всех факторов влияния балльные оценки $\{B_{ij}, i=1, ..., I, j=1, ..., J_i\}$ подставляются в формулу:

$$k_{N_j} = \frac{B_n}{B_{\text{ср}}},$$

где B_n – балльная оценка n -го участка трассы МН, определяемая как:

$$B_n = \sum_{i=1}^I \sum_{j=1}^{J(i)} \rho_i q_{ij} B_{ij},$$

где $B_{\text{ср}}$ – средняя балльная оценка трассы МН (среднее арифметическое), полученная на основе балльной оценки каждого участка трассы ($n=1, ..., N$), а N – общее количество рассматриваемых участков трассы ЛЧ МН.

Из-за отсутствия необходимых статистических данных для определения коэффициентов ρ_i и q_{ij} применим метод экспертной оценки. Поскольку количество долей факторов невелико ($n \leq 3$), воспользуемся методом непосредственной оценки экспертной группой с использованием метода Дельфи [9]. В результате опроса и экспертной оценки рассматриваются группы факторов влияния, представленные в табл. 4. В состав экспертной группы службой безопасности включены сотрудники ОАО «Центрсибнефтепровод» различного

го уровня — от охранников и начальников команд службы безопасности до управленческого персонала организации.

Таблица 4. Группы факторов влияния

Обозначение и наименование группы факторов		Доля группы, ρ_i		
		Угроза Y_1^s	Угроза Y_2^s	Угроза Y_3^s
$Гр_1$	Предпосылки акта незаконного вмешательства	0,20	0,20	0,80
$Гр_2$	Реализация акта незаконного вмешательства	0,40	0,40	0,10
$Гр_3$	Извлечения выгоды в результате акта незаконного вмешательства	0,40	0,40	0,10

В результате повторных раундов опроса и экспертной оценки внутри каждой из групп были получены факторы влияния и их весовые коэффициенты, представленные в табл. 5.

Таблица 5. Факторы влияния

Обозначение и наименование фактора влияния		Доля в группе q_{ij}		
		Угроза Y_1^s	Угроза Y_2^s	Угроза Y_3^s
Группа 1: Предпосылки незаконного вмешательства				
F_{11}	Криминогенная обстановка	0,20	0,30	0,70
F_{12}	Населенность прилегающих территорий	0,60	0,40	0,20
F_{13}	Уровень занятости населения прилегающих территорий	0,20	0,30	0,10
Группа 2: Реализация незаконного вмешательства				
F_{21}	Защищенность объекта	0,10	0,10	0,60
F_{22}	Наличие подъездных путей	0,60	0,60	0,10
F_{23}	Наличие специального оборудования, персонала и транспорта для реализации акта незаконного вмешательства	0,30	0,30	0,30
Группа 3: Извлечения выгоды в результате незаконного вмешательства				
F_{31}	Наличие на прилегающих территориях пунктов сбыта похищенной продукции	0,90	0,80	0
F_{32}	Репутационные, экологические и политические последствия	0,10	0,20	1

В качестве примера в табл. 6 приведены значения отдельных составляющих фактора F_{31} и соответствующие им балльные оценки $B_{31}^{(m)}$, где m — номер составляющей. Итоговая балльная оценка для данного фактора рассчитывается как сумма балльных оценок нижеприведенных пяти составляющих. Если сумма баллов превышает 10, то $B_{31}=10$.

Поскольку необходимые статистические данные для анализа коэффициентов ρ_i и q_{ij} отсутствуют, проверка их адекватности была выполнена с помощью метода ранжирования. Метод ранжи-

рования является более достоверным, чем метод непосредственной оценки факторов влияния, с помощью которого первоначально были определены коэффициенты ρ_i и q_{ij} [10]. В результате ранжирования факторов экспертной группой того же состава значения рангов факторов влияния совпали с порядками произведения коэффициентов $\rho_i \cdot q_{ij}$, расположенных в порядке возрастания. Поскольку значение коэффициента координации $0,92 \geq 0,5$, можно утверждать о согласованности экспертов и адекватности оценки коэффициентов ρ_i и q_{ij} .

Таблица 6. Балльная оценка фактора B_{31}

m	Наименование составляющей m , В районе пролегания участка нефтепровода	$B_{31}^{(m)}$
1	Находятся потребители, использующие нефть в качестве топлива (малые котельные, теплицы, хлебо-заводы, асфальтобитумные заводы и др.)	5
2	Имеются сведения о наличии нелегальных нефтеперерабатывающих заводов	10
3	Располагаются частные малые нефтеперерабатывающие заводы	7
4	Располагаются крупные нефтеперерабатывающие заводы	2
5	Осуществляют предпринимательскую деятельность юридические лица, занимающиеся торговлей нефтепродуктами	3

Результаты использования вышеописанной методики для оценки вероятностей $p_{N_j}(y_i^s)$ реализации угроз ФБ (в течение года) километровых участков ЛЧ МН, эксплуатируемых ОАО «Центрсибнефтепровод», представлены на рисунке. Полученные данные погонного распределения $p_{N_j}(y_i^s)$ участков определенного ЛЧ МН могут быть использованы для планирования мероприятий по укреплению защищенности конкретных участков, а также для менеджмента рисков ФБ ЛЧ МН.

Заметим, что вероятности реализации угроз $\bar{p}_{y_i^s}$ в формулах (1)–(5) составляют коммерческую тайну ОАО «Центрсибнефтепровод». Поэтому данные, использованные для расчета описанной выше модели угроз ЛЧ МН «Александровское–Анжеро-Судженск», являются гипотетическими и носят чисто иллюстративный характер.

Аналогов разработанной модели угроз ФБ и предложенной методики оценки вероятности для линейных объектов ТЭК авторами найдено не было. Реализация разработанной модели и предложенной методики включена в технические требования по созданию автоматизированной системы контроля защищенности ЛЧ МН в ОАО «Центрсибнефтепровод». Предложенная модель угроз в результате изменения перечня злоумышленников, угроз и переоценки коэффициентов факторов влияния может быть применена для моделирования угроз ФБ других линейных объектов ТЭК: электрических сетей, магистральных газопроводов, нефтепродуктопроводов.

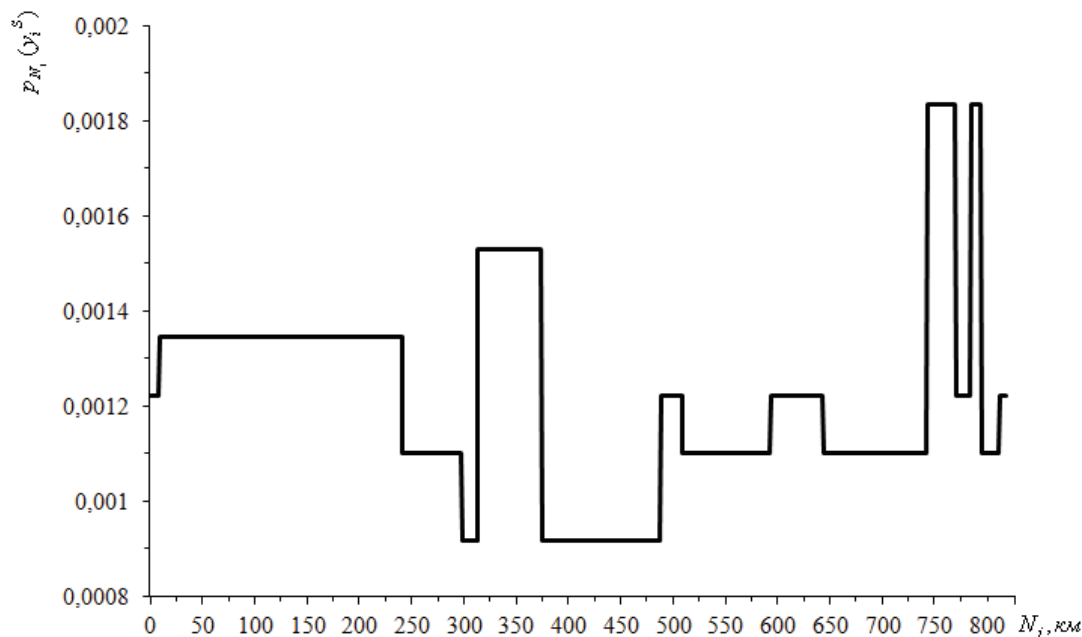


Рисунок. Вероятностная характеристика МН «Александровское–Анжеро-Судженск»

Выводы

Предложенная выше многоуровневая иерархическая модель пространственно-протяженной инфраструктуры ЛЧ МН как объекта защиты использовалась нами для формирования модели угроз ФБ ЛЧ МН. В качестве основной характеристики угроз ФБ использована вероятность их реализации для нижнего уровня декомпозиции объекта защиты. Полученные соотношения (1)–(5) являются основой для проведения последующей процедуры менеджмента рисков и оптимизации соответствующих реальных систем ФБ распределенных систем.

С помощью указанных соотношений установлены факторы влияния на уровни угроз ЛЧ МН, а также определены весовые коэффициенты, учитывающие вклад каждого фактора. В качестве модельной иллюстрации произведена оценка вероятности реализации угроз ФБ для участков МН «Александровское–Анжеро-Судженск», а также соответствующие вероятностные характеристики.

Выражаем благодарность за помощь в подготовке статьи Геннадию Наумовичу Глазову, ведущему научному сотруднику ЗАО «НПФ «Микран».

СПИСОК ЛИТЕРАТУРЫ

1. Федеральный закон Российской Федерации от 21 июля 2011 г. N 256-ФЗ «О безопасности объектов топливно-энергетического комплекса» // Российская газета – издание Правительства Российской Федерации, официальный публикатор документов. 2013. URL: <http://www.rg.ru/2011/07/26/tek-dok.html> (дата обращения: 04.02.2013).
2. Безопасность России. Правовые, социально-экономические и научно-технические аспекты. Анализ риска и проблем безопасности: в 4-х ч. Ч. 4. Научно-методическая база анализа риска и безопасности / науч. руковод. К.В. Фролов. – М.: МГОФ «Знание», 2007. – 864 с.
3. Кононов А.А., Стилиславский А.Б., Цыгичко В.Н. Управление рисками нарушения транспортной безопасности. – М.: АС-Траст, 2008. – 210 с.
4. Кононов А.А., Черныш К.В., Гуревич Д.С., Поликарпов А.К. Оценка рисков в иерархических структурах критически важных объектов // Управление рисками и безопасностью / под ред. Д.С. Черешкина. – М.: ЛЕНАНД, 2010. – Т. 52. – С. 5–15.
5. ГОСТ Р 50922-2006. Защита информации. Основные термины и определения // Каталог национальных стандартов – Росстандарт. 2013. URL: <http://protect.gost.ru/document.aspx?control=7&baseC=6&page=0&month=1&year=2008&search=&id=129024> (дата обращения: 04.02.2013).
6. Факты незаконных врезок по системе «Транснефть» // ОАО «АК «Транснефть». 2013. URL: <http://www.transneft.ru/dagestan/263/> (дата обращения: 04.02.2013).
7. Безопасность России. Правовые, социально-экономические и научно-технические аспекты. Анализ рисков и управление безопасностью: методические рекомендации / науч. руковод. К.В. Фролов. – М.: МГОФ «Знание», 2008. – 672 с.
8. Гмурман В.Е. Теория вероятностей и математическая статистика. – М.: Изд-во «Юрайт», 2013. – 479 с.
9. Безопасность России. Правовые, социально-экономические и научно-технические аспекты. Анализ рисков и управление безопасностью: в 4-х ч. Ч. 3. Прикладные вопросы анализа рисков критически важных объектов / науч. руковод. К.В. Фролов. – М.: МГОФ «Знание», 2007. – 816 с.

Поступила 04.03.2013 г.

УДК 519.688:622.279.23

ИДЕНТИФИКАЦИЯ ЭВОЛЮЦИОННЫХ ПРОЦЕССОВ ЖИЗНЕННОГО ЦИКЛА СИСТЕМ С УЧЕТОМ АПРИОРНОЙ ИНФОРМАЦИИ

А.Г. Наймушин, В.Л. Сергеев

Томский политехнический университет

E-mail: SergeevVL@ignd.tpu.ru

Рассматриваются проблемы параметрической идентификации эволюционных процессов, для решения которых предлагается использовать интегрированные системы феноменологических моделей с учетом априорной информации. Приводятся примеры решения задачи идентификации и прогноза жизненного цикла накопленной продукции в процессе разработки нефтяного месторождения.

Ключевые слова:

Идентификация, феноменологические модели, процессы эволюции, жизненный цикл, априорная информация, нефтяное месторождение, извлекаемые запасы.

Key words:

Identification, phenomenological models, processes of evolution, life cycle, a-priori information, oil deposit, recoverable reserves.

Введение

В настоящее время в связи с возрастающей ролью системного подхода при проектировании и управлении в условиях неопределенности актуальным является решение задач идентификации и прогноза эволюционных процессов жизненного цикла сложных технических и социально-экономических систем. Именно проблема идентификации занимает исключительно важную роль, поскольку является наиболее «узким местом» при проектировании наукоемких и интеллектуальных систем управления и принятия решений [1, 2].

Для прогнозирования жизненного цикла большое внимание уделяется феноменологическим моделям

$$\frac{dy}{dt} = f(t, y, \alpha) \quad (1)$$

эволюционных процессов

$$y_t^* = y_t + \xi_t = y(t_0) + \int_{t_0}^t f(\tau, y, \alpha) d\tau + \xi_t, \quad t = \overline{1, n}, \quad (2)$$

отражающих их целостные системные свойства. Здесь y_t^* , y_t , $t = \overline{1, n}$ – фактические и вычисленные на основе модели значения процесса в различные моменты времени t ; $f(t, y, \alpha)$ – известные с точностью до вектора параметров $\alpha = (\alpha_j, j = \overline{1, m})$ функции; ξ_t , $t = \overline{1, n}$ – различного рода случайные факторы, представляющие погрешности расчета траектории процесса, ошибки измерения y_t , неточности при выборе модели процесса и т. д.

Следует отметить, что наряду с описанием эволюционного процесса в пространстве «вход–выход» (2) известна и другая форма представления феноменологических моделей эволюционных процессов в пространстве состояний, заданная двумя системами моделей, в принципе, сводимая к одной системе [2].

Примерами эволюционных моделей (1), (2) являются текущая емкость рынка инновационного товара, накопленная добыча нефти и газа в про-

цессе разработки месторождений углеводородов, забойные давления при гидродинамических исследованиях скважин на неустойчивых режимах фильтрации [3–5].

Однако при решении задачи параметрической идентификации, связанной с определением параметров феноменологической модели жизненного цикла (2), возникают проблемы обеспечения устойчивости, повышения точности оценок, учета дополнительной априорной информации, особенно на начальной стадии развития процесса, когда объем исходных данных n мал.

В данной работе для решения отмеченных выше проблем идентификации эволюционных процессов предлагается использовать интегрированные системы феноменологических моделей с учетом дополнительной априорной информации.

Модели и алгоритмы

идентификации эволюционных процессов

Основной задачей идентификации эволюционных процессов жизненного цикла является интегрированная система феноменологических моделей вида

$$\begin{cases} y_t^* = f_0(t, y, \alpha) + \xi_t, \quad t = \overline{1, n}, \\ \bar{x}_j = f_{aj}(y_t, \alpha) + \eta_j, \quad j = \overline{1, d}, \end{cases} \quad (3)$$

где первая система из n уравнений – стохастическая модель исследуемого эволюционного процесса жизненного цикла (2), а вторая система из d уравнений представляет модели объектов аналогов, позволяющих учитывать дополнительную априорную информацию $\bar{x}_j$, известную к моменту времени t . В качестве априорной информации могут быть использованы данные о параметрах эволюционного процесса $\bar{x}_j = \bar{\alpha}_j$, $j = \overline{1, m}$, известные к моменту времени t с погрешностью η_j , будущие значения траектории эволюционного процесса $\bar{x}_{(n+j)} = \bar{y}_{n+j}$, $j = \overline{1, l}$, в том числе и его предельные значения (аттракторы $\bar{x}_\infty = y_\infty$). Модели объектов аналогов $f_{aj}(y, \alpha)$ могут

представлять функции, функционалы, а в общем виде операторы f_{ij} от переменных y_i в классах линейных, нелинейных параметрических либо непараметрических моделей [1]; ξ_i, η_j – случайные неконтролируемые факторы.

Рассмотрим примеры интегрированных систем феноменологических моделей с учетом априорной информации. При использовании априорной информации о параметрах моделей эволюционного процесса $\bar{\alpha}_t$, полученных к моменту времени t , системе моделей (3) можно представить в виде

$$\begin{cases} y_t^* = f(t, y_t, \alpha) + \xi_t, \quad t = \overline{1, n}, \\ \bar{\alpha}_t = \alpha + \eta. \end{cases} \quad (4)$$

В случае логистической модели роста [3] и априорных сведений о ее аттракторе $\bar{x}_\infty$ модель (4) принимает достаточно простой вид

$$\begin{cases} y_t^* = y(t_0) + \int_{t_0}^t (\alpha_1 y_\tau - \alpha_2 y_\tau^2) d\tau + \xi_t, \quad t = \overline{1, n}, \\ \bar{x}_\infty = \alpha_1 / \alpha_2 + \eta. \end{cases} \quad (5)$$

При использовании априорной информации о параметрах феноменологической модели $\bar{\alpha}$ и экспертных оценок будущих значений эволюционного процесса $\bar{y}_{n+j}$ система моделей (3) принимает вид

$$\begin{cases} y_t^* = f_o(t, y_t, \alpha) + \xi_t, \quad t = \overline{1, n}, \\ \bar{\alpha} = \alpha + \eta, \quad \bar{y}_{n+j} = f(y_{n+j}, \alpha) + \nu_j, \quad j = \overline{1, l}. \end{cases} \quad (6)$$

Следует отметить, что уравнения вида (3)–(6) представляют простейшую интегрированную систему моделей первого уровня. По аналогии с [1] имеют место и многоуровневые иерархические интегрированные системы феноменологических моделей.

С позиции системного подхода процесс параметрической идентификации эволюционных моделей (3) можно представить как решение оптимизационных задач вида [1]

$$\alpha_n^*(\beta) = \arg \min_{\alpha} \Phi(\alpha, \beta), \quad (7)$$

$$\beta_n^* = \arg \min_{\beta} J_0(\alpha_n^*(\beta)), \quad (8)$$

где запись $\arg \min f(x)$ означает точку минимума x^* функции $f(x)$ ($f(x^*) = \min_x f(x)$); $\Phi(\alpha, \beta) = \Phi(J_0(\alpha), \beta_k J_k(\alpha_n), k = \overline{1, l})$ – комбинированный показатель качества системы моделей (3), представляющий заданную функцию (функционал) Φ от частного показателя качества $J_0(\alpha_n)$ модели эволюционного процесса жизненного цикла и взвешенных весами β_k частных показателей качества $J_k(\alpha_n)$ моделей дополнительных априорных данных и экспертных оценок.

Следует отметить, что рассматриваемая технология выбора оптимальных решений (7), (8) позволяет синтезировать достаточно широкий спектр известных и новых алгоритмов идентификации для линейных и нелинейных интегрированных систем моделей эволюционных процессов (3), а также для различных показателей качества и методов решения оптимизационных задач. Например, для линейной

по параметрам α интегрированной системы моделей эволюционных процессов жизненного цикла

$$\begin{cases} y^* = F_0 \alpha + \xi, \\ \bar{x} = F_a \alpha + \eta \end{cases} \quad (9)$$

и комбинированного показателя качества Φ , выбранного в виде суммы взвешенных частных квадратичных показателей качества

$$\Phi(\alpha, \beta) = \|y^* - F_0 \alpha\|_W^2 + \|\bar{x} - F_a \alpha\|_{W(\beta)}^2, \quad (10)$$

оптимизационная задача (7) сводится к решению систем линейных уравнений вида

$$\begin{aligned} (F_0^T W F_0 + F_a^T W(\beta) F_a) \alpha^*(\beta) = \\ = (F_0^T W y^* + F_a^T W(\beta) \bar{x}), \end{aligned} \quad (11)$$

где запись $\|X\|_W^2$ означает квадратичную форму $X^T W X$; $y^* = (y_i^*, i = \overline{1, n})$ – вектор фактических значений эволюционного процесса; $\bar{x} = (\bar{x}_j, j = \overline{1, d})$ – вектор дополнительных априорных сведений и экспертных оценок; $F_0 = (\varphi_j(t_{i-1}, y_i), j = \overline{1, m}, i = \overline{1, n})$ – матрица известных функций $\varphi_j(t_{i-1}, y_i)$ в модели исследуемого объекта размерности $(m \times n)$; $F_a = (\varphi_{aj}(y_i), j = \overline{1, d}, k = \overline{1, m})$ – матрица известных функций $\varphi_{aj}(y_i)$ в модели объектов аналогов размерности $(d \times m)$; $W(\beta) = \text{diag}(\beta_k, k = \overline{1, m})$ – диагональная матрица весовых функций, определяющих значимость (вес) априорных данных $\bar{x}$; W – известная матрица вероятностно-статистических характеристик случайных величин $\xi_i, i = \overline{1, n}$. Для получения системы линейных уравнений (11) достаточно взять частные производные по параметрам α от комбинированного функционала (10) и приравнять их к нулю.

Для нелинейной интегрированной системы моделей эволюционных процессов жизненного цикла вида

$$\begin{cases} y^* = f_0(\alpha) + \xi, \\ \bar{x} = f_a(\alpha) + \eta, \end{cases} \quad (12)$$

и комбинированного показателя качества Φ

$$\Phi(\alpha, \beta) = \|y^* - f_0(\alpha)\|_W^2 + \|\bar{x} - f_a(\alpha)\|_{W(\beta)}^2$$

оптимизационная задача (7) при использовании метода Гаусса–Ньютона [6] сводится к последовательному решению систем линейных уравнений вида

$$\begin{cases} \alpha_i^* = \alpha_{i-1}^* + h_i \Delta \alpha_{i-1}^*, \quad i = 1, 2, 3, \dots, \\ (D_0^T W D_0 + D_a^T W(\beta) D_a)_{i-1} \Delta \alpha_{i-1}^* = \\ = (D_0^T W e_0 + D_a^T W(\beta) e_a)_{i-1}, \end{cases} \quad (13)$$

где

$$\begin{aligned} D_0 &= \left(\frac{\partial f_o(t_i, y_i, \alpha)}{\partial \alpha_j}, \quad i = \overline{1, n}, j = \overline{1, m} \right)_{nm}, \\ D_a &= \left(\frac{\partial f_{a,k}(t, y, \alpha)}{\partial \alpha_j}, \quad k = \overline{1, l}, j = \overline{1, m} \right)_{lm} \end{aligned}$$

— матрицы частных производных от моделей исследуемого процесса и моделей объектов аналогов; $\mathbf{e}_0 = (\mathbf{y}^* - \mathbf{f}_0(\boldsymbol{\alpha}))$, $\bar{\mathbf{e}}_a = (\bar{\mathbf{x}} - \mathbf{f}_a(\boldsymbol{\alpha}))$ — векторы невязок, h_i — параметр шага.

Отметим, что приведенные выше оценки параметров интегрированных систем феноменологических моделей (9), (12) при определенных значениях управляющих параметров и дополнительных априорных сведений соответствуют многим традиционным методам идентификации. Например, из (11) при нулевых значениях априорных данных $\bar{\mathbf{x}}_j = 0, j = \overline{1, d}$ и $F_a = W = I, \beta_k = \beta, k = \overline{1, m}$ (где I — единичная матрица) следуют известные Ridge-приближения и регуляризирующие по Тихонову оценки параметров линейных регрессионных моделей [6]

$$\boldsymbol{\alpha}^* = (F_0^T F_0 + I \beta)^{-1} F_0^T \mathbf{y}^*, \quad (14)$$

позволяющие получать устойчивые решения при вырожденности либо плохой обусловленности матрицы $F_0^T F_0$. Следует отметить, что оценки (14) являются оптимальными приближениями параметров, доставляющих минимум стабилизирующего функционала А.Н. Тихонова [7]

$$\Phi(\boldsymbol{\alpha}, \beta) = \|\mathbf{y}^* - F_0 \boldsymbol{\alpha}\|_w^2 + \beta \|\boldsymbol{\alpha}\|^2,$$

следующего из комбинированного показателя качества (10).

При аналогичных условиях оценки параметров (13) для нелинейных моделей совпадают с устойчивыми оценками метода Левенберга—Марквардта [6]

$$\begin{cases} \boldsymbol{\alpha}_i^* = \boldsymbol{\alpha}_{i-1}^* + h_i \Delta \boldsymbol{\alpha}_{i-1}^*, \quad i = 1, 2, 3, \dots, \\ (D_0^T D_0 + I \beta)_{i-1} \Delta \boldsymbol{\alpha}_{i-1}^* = (D_0^T \mathbf{e}_0 + W(\boldsymbol{\beta}) \bar{\mathbf{e}}_a)_{i-1}, \end{cases} \quad (15)$$

а при $\beta = 0$ — с методом Гаусса—Ньютона

$$\begin{cases} \boldsymbol{\alpha}_i^* = \boldsymbol{\alpha}_{i-1}^* + h_i \Delta \boldsymbol{\alpha}_{i-1}^*, \quad i = 1, 2, 3, \dots, \\ (D_0^T D_0)_{i-1} \Delta \boldsymbol{\alpha}_{i-1}^* = (D_0^T \mathbf{e}_0)_{i-1}. \end{cases} \quad (16)$$

Следует также отметить, что задача (8) по определению оптимальных значений вектора управляющих параметров $\boldsymbol{\beta}^*$ не имеет аналитического решения и решается методами последовательных приближений.

Идентификация эволюционного процесса накопленной добычи нефти

На рис. 1–3 и в таблице приведены результаты решения актуальной задачи анализа и контроля разработки лицензионных участков нефтяных месторождений, прогноза накопленной добычи нефти и оценки извлекаемых запасов.

Для решения задачи интеграции геологической и технологической информации и идентификации эволюционного процесса использовалась интегрированная система моделей накопленной добычи нефти вида

$$\begin{cases} y_i^* = f(t, \boldsymbol{\alpha}) + \xi_i = \\ = y(t_0) + \alpha_1 \int_{t_0}^t \exp(-\alpha_2 \tau) \tau^{\alpha_3} d\tau + \xi_i, \quad t = \overline{1, n}, \\ \bar{z}_\infty = \bar{\kappa} Q + \eta = f(\infty, \boldsymbol{\alpha}) + \eta, \end{cases} \quad (17)$$

где $y_i^*, t = \overline{1, n}$ — фактические значения добычи нефти объектов разработки за период времени Δt ; $y(t_0)$ — накопленная к начальному моменту времени t_0 добыча нефти; $\bar{z}_\infty$ — априорная информация об извлекаемых запасах; Q — геологическая оценка балансовых запасов; $\bar{\kappa}$ — экспертная оценка коэффициента извлечения.

Фактические значения накопленной добычи нефти пласта Ю₂ нефтяного месторождения Томской области за 23 года разработки приведены на рис. 1–3 сплошной линией 1. Линии 2–5 представляют оценки прогнозной добычи нефти $y(t_n + \tau)$, начиная со второго года разработки

$$\begin{aligned} \hat{y}(t_n + \tau) &= f(t_n + \tau, \boldsymbol{\alpha}_n^*(\beta_n^*)), \\ t_n &= n, n = \overline{1, 5}, \tau = \overline{1, 23 - n}, \end{aligned} \quad (18)$$

где $\boldsymbol{\alpha}_n^*(\beta_n^*)$ — оценки параметров модели (17), полученные на основе (13) при соответствующих значениях матриц частных производных от модели годовой добычи нефти, векторов невязок и при оптимальном значении управляющего параметра $\beta_n^*(3)$, полученного с использованием метода золотого сечения.

Точные значения извлекаемых запасов за все время разработки нефтяного пласта Ю₂ составили $z_\infty = 7,4 \cdot 10^6$ тонн. Априорная информация об извлекаемых запасах выбиралась равной $\bar{z}_\infty = 5 \cdot 10^6$ тонн с ошибкой порядка 30 %.

Корректировка априорной информации $\bar{z}_\infty$ в модели (17) проводилась по схеме

$$\bar{z}_{\infty, n} = f(T, \boldsymbol{\alpha}_n^*(\beta_n^*)), n = \overline{2, 3}, \dots,$$

где T соответствует времени завершения разработки нефтяного пласта ($T = 43$).

В таблице приведены значения относительных ошибок оценок извлекаемых запасов за первые 7 лет разработки нефтяного пласта

$$\delta_n = \text{abs}((f(T, \boldsymbol{\alpha}_n^*(\beta_n^*)) - z_\infty) / z_\infty), n = \overline{2, 7},$$

полученные на основе (13) и условно названные методом интегрированных моделей (МИМ), методом Гаусса—Ньютона (МГН) (16) и методом Левенберга—Марквардта (МЛМ) (15).

Из таблицы видно, что метод интегрированных моделей позволил улучшить экспертную оценку извлекаемых запасов с 30 до 5 % ошибки, что отражено и на графиках рис. 1 (линии 2–4), демонстрирующих достаточно быструю сходимость прогнозных значений накопленной добычи нефти (18) к их фактическим значениям.

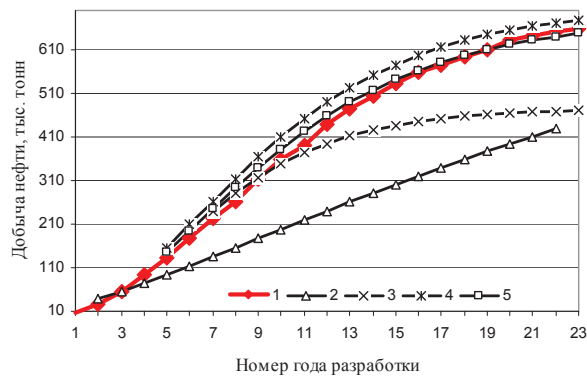


Рис. 1. Фактические (линия 1) и прогнозные значения добычи нефти (линии 2–5) с учетом априорной информации о запасах

Таблица. Относительная ошибка оценок извлекаемых запасов, %

Методы	Длительность разработки (номер года)					
	2	3	4	5	6	7
МИМ	0,461	0,244	0,173	0,086	0,059	0,046
МГН	–	8,405	6,537	0,369	0,469	0,319
МЛМ	3,483	0,363	0,312	0,343	0,349	0,301

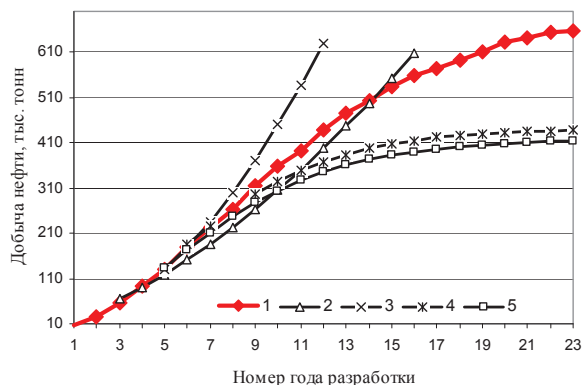


Рис. 2. Фактические (линия 1) и прогнозные значения добычи нефти (линии 2–5) с использованием метода Гаусса–Ньютона

СПИСОК ЛИТЕРАТУРЫ

- Сергеев В.Л. Интегрированные системы идентификации. — Томск: Изд-во ТПУ, 2011. — 198 с.
- Анфилов В.С., Емельянов А.А., Кукушкин А.А. Системный анализ в управлении. — М.: Финансы и статистика, 2002. — 368 с.
- Кожухова В.Н., Семенов Е.В. Методы идентификации логистической динамики и жизненного цикла продукта моделью Верхульста // Экономика и математические методы. — 2012. — Т. 48. — № 2. — С. 108–115.
- Кемеров П.А., Сергеев В.Л., Ананьев А.С. Адаптивная идентификация и интерпретация нестационарных гидродинамиче-

Для сравнения качества прогноза на рис. 2 и 3 приведены прогнозные значения накопленной добычи нефти, полученные с использованием метода Гаусса–Ньютона, начиная с третьего года разработки, и метода Левенберга–Марквардта, — со второго года разработки.

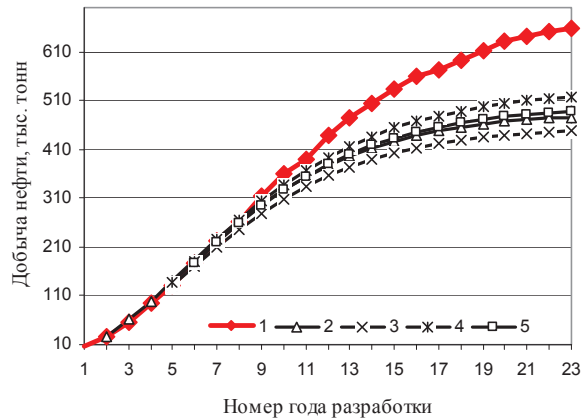


Рис. 3. Фактические (линия 1) и прогнозные значения добычи нефти (линии 2–5) с использованием метода Левенберга–Марквардта

Выводы

- Для идентификации эволюционных процессов жизненного цикла систем предложено использовать интегрированные системы феноменологических моделей с учетом априорной информации.
- Приведен метод синтеза и примеры оптимальных, в смысле квадратичных показателей качества, оценок параметров линейных и нелинейных интегрированных систем феноменологических моделей.
- На примере идентификации эволюционного процесса накопленной добычи нефти показано, что предлагаемые модели и алгоритмы, условно названные методом интегрированных моделей, позволяют при малом объеме промысловых данных получить более точные оценки извлекаемых запасов по сравнению с классическими алгоритмами Гаусса–Ньютона и Левенберга–Марквардта.

ских исследований с учетом притока продукции в скважине // Известия Томского политехнического университета. — 2011. — Т. 319. — № 4. — С. 43–46.

- Хасанов М.М., Карачурин Н.А., Тяжев Е.А. Оценка извлекаемых запасов на основе феноменологических моделей // Вестник инженерного центра ЮКОС. — 2001. — № 2. — С. 3–7.
- Демиденко Е.З. Линейная и нелинейная регрессии. — М.: Финансы и статистика, 1981. — 300 с.
- Тихонов А.Н., Арсенин В.Я. Методы решения некорректных задач. — М.: Наука, 1979. — 392 с.

Поступила 24.12.2012 г.

УДК 681.5

СИНТЕЗ ВСТРАИВАЕМОЙ ОДНОКОНТУРНОЙ СИСТЕМЫ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ С САМОНАСТРАИВАЮЩИМСЯ РЕГУЛЯТОРОМ И ОЦЕНКА ЕЁ РОБАСТНОСТИ

В.В. Курганкин, С.В. Замятин, В.М. Замятин, М.И. Пушкарев

Томский политехнический университет

E-mail: kurgankin@tpu.ru

Рассмотрена возможность применения принципа динамической компенсации для синтеза встраиваемой одноконтурной системы автоматического управления с самонастраивающимся регулятором. Приведены результаты натурных экспериментов. Проведён анализ свойств робастности синтезированной системы.

Ключевые слова:

Система автоматического управления, одноконтурная система, встраиваемая система, синтез регулятора, принцип динамической компенсации, робастность, микроконтроллер.

Key words:

Automatic control system, single-circuit system, embedded system, regulator design, dynamic compensation principle, robustness, microcontroller.

Синтез регулятора системы автоматического управления (САУ) является одной из ключевых задач теории управления. Как правило, исходными данными для решения задачи синтеза являются математическое описание объекта управления и некоторые требования к функционированию синтезируемой системы. Зачастую подобные требования предъявлены только к одному или двум показателям качества функционирования системы. Синтез регулятора по таким исходным данным рассматривается в работах [1–6]. Широко распространенные инженерные методы синтеза регуляторов сводятся к аппроксимации уравнения синтеза, но не к его точной реализации. Такая аппроксимация направлена на упрощение структуры регулятора. Это приводит к снижению точности, а иногда и к невозможности решения. При этом приходится дополнительно разрабатывать алгоритмы и методы аппроксимации уравнения синтеза, что значительно усложняет задачу.

В работе рассматривается реализация алгоритмов автоматической настройки регулятора встраиваемой системы управления и её экспериментальные исследования с объектом, в качестве которого выступает двигатель постоянного тока (ДПТ).

САУ может быть представлена в виде структурной схемы (рис. 1).

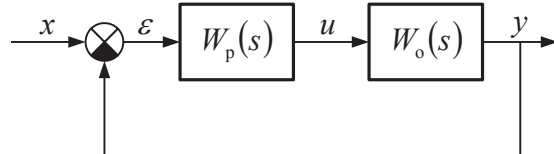


Рис. 1. Схема одноконтурной САУ: $W_o(s)$, $W_p(s)$ – передаточные функции объекта и регулятора; x и y – входной и выходной сигналы; ε – сигнал рассогласования; u – сигнал управления

Для класса стационарных систем задачу синтеза можно разделить на два этапа [1]:

- выбор эталонной передаточной функции (ПФ) системы, удовлетворяющей поставленным техническим требованиям;

- определение параметров элементов, входящих в систему управления, из условия равенства эталонной и реальной передаточной функции.

Задача состоит в автоматическом определении передаточной функции регулятора $W_p(s)$ встроеной САУ.

В математическом плане для системы, представленной на рис. 1, задача заключается в составлении и решении уравнения синтеза

$$W_{\text{ж}}(s) \cong \frac{W_p(s)W_o(s)}{1 + W_p(s)W_o(s)}, \quad (1)$$

где $W_{\text{ж}}(s)$ – желаемая ПФ замкнутой системы управления.

Уравнение сформировано на основе равенства желаемой ПФ САУ $W_{\text{ж}}(s)$ и модели синтезируемой системы, представленной в правой части (1).

Для решения необходимо иметь ПФ объекта управления $W_o(s)$ и желаемую ПФ системы $W_{\text{ж}}(s)$. Первая из них известна или может быть найдена по заданному объекту. Вторая ПФ должна быть определена проектировщиком на основе требований, предъявляемых к системе.

Таким образом, составляющие формулы (1) $W_{\text{ж}}(s)$ и $W_o(s)$ известны, поэтому принципиально возможен расчёт оператора $W_p(s)$, который сводится к определению структуры регулятора и численных значений его параметров

$$W_p(s) = \frac{W_{\text{ж}}(s)}{W_o(s)(1 - W_{\text{ж}}(s))}. \quad (2)$$

Рассматриваемый в работе подход применяется для полной нейтрализации влияния динамических характеристик модели объекта на оператор замкнутой системы. Такой подход основан на принципе динамической компенсации [1, 5]. Формально зависимость (2) даёт точное решение задачи синтеза регулятора. Несмотря на это, принцип динамической компенсации не получил широкого распространения в связи с усложнением структуры САУ. Бурное развитие электроники и микропроцессор-

ной техники позволило снять вычислительные ограничения, возникающие при реализации сложных и ресурсоемких алгоритмов управления, в том числе и основанных на принципе динамической компенсации, средствами встраиваемых САУ.

Алгоритмы формирования желаемых ПФ, применяемые во встраиваемой системе управления скоростью ДПТ, основаны на методе Коновалова–Огурка [7]. Модель объекта по экспериментальным характеристикам формируется на основе подхода, изложенного в [8].

Функциональная схема установки представлена на рис. 2.

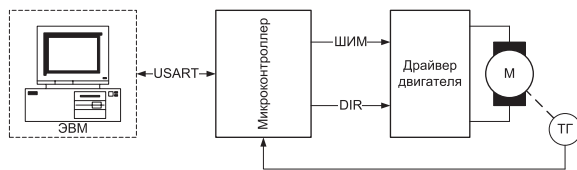


Рис. 2. Функциональная схема установки: М – ДПТ; ТГ – тахогенератор (датчик скорости); ЭВМ – электронно-вычислительная машина; USART – интерфейс связи микроконтроллера и ЭВМ; DIR – сигнал, определяющий направление вращения вала ДПТ; Широтно-импульсная модуляция (ШИМ) – управляющий сигнал, задающий скорость вращения вала ДПТ

Устройство реализовано на базе микроконтроллера STM32F103. Основные характеристики: 32-разрядное ядро процессора; максимальная частота процессора 72 МГц; 512 кб ПЗУ; 64 кб ОЗУ; 12-разрядный АЦП; максимальное время выборки АЦП 1 мкс.

В качестве объекта управления используется ДПТ «Dynamo» Sliven PIVT6–25/3A со встроенным тахогенератором. Основные характеристики двигателя: напряжение питания 30 В; скорость вращения 3000 об/мин; момент вращения 0,11 Н·м; пусковой ток 4,5 А; электрохимическая постоянная времени 25 мс; электрическая постоянная времени 2 мс; момент инерции якоря двигателя $0,051 \cdot 10^{-3}$ кг·м²; константа тахогенератора 3 В/1000 мин⁻¹.

В качестве драйвера двигателя используется микросхема Pololu High-Power Motor Driver 18v15. Основные характеристики драйвера: диапазон рабочих напряжений 5,5...30 В; рабочий ток ≤ 15 А; максимальная частота ШИМ 40 кГц; величина «мертвого времени» 3 мкс.

На основе предложенных алгоритмов разработано программное обеспечение, реализованное на вычислительной платформе STM32F103.

Формально процедура синтеза регулятора САУ выглядит следующим образом.

1. На объект подается тестовый сигнал.
2. Формируются векторы входных и выходных значений объекта.
3. На основе полученных векторов осуществляется структурно-параметрическая идентификация объекта с помощью вещественного интерполяционного метода [3, 8].
4. На основе заданных показателей качества формируется желаемая ПФ системы с помощью метода Коновалова–Огурка.

5. На основе выражения (2) определяется структура и параметры регулятора.

Рассмотрим работу реализованной встраиваемой системы управления.

В момент запуска устройства формируется тестовый ступенчатый сигнал, уровень которого равен 50 % от максимального (коэффициент заполнения ШИМ – 50 %, среднее значение напряжения на якоре двигателя – 15 В). Устройство запоминает вектор выходных значений скорости вращения двигателя, сформированных в результате подачи тестового сигнала. На основании полученных входных и выходных векторов данных и алгоритма структурно-параметрической идентификации формируется математическое описание объекта.

Для оценки адекватности модели используется выражение [8]

$$\Delta y = \frac{|y_m(kT_0) - y(kT_0)|}{y_{уст}} 100 \%,$$

где $y(t)$ и $y_m(t)$ – функции времени выходного сигнала объекта и модели соответственно; T_0 – период квантования; $k=1,2, \dots$ – номер отсчета, взятый на выходных характеристиках модели и объекта; $y_{уст}$ – установившееся значение снятой характеристики.

Для заданной структуры модели критерием ее адекватности является $\max(\Delta y)$.

В результате процедуры идентификации ДПТ получена следующая модель объекта

$$W_{ol}(s) = \frac{5,054 \cdot 10^{-2}}{2,773 \cdot 10^{-2} s + 1}.$$

На рис. 3 представлены результаты идентификации объекта управления.

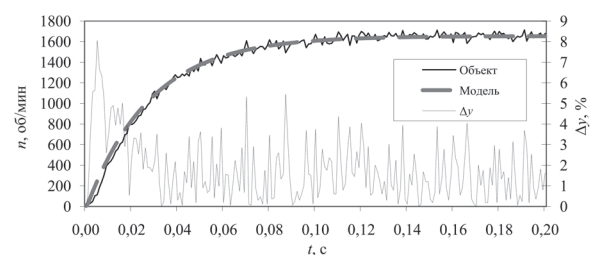


Рис. 3. Результаты идентификации объекта управления

Ошибка идентификации по заданному критерию равна 8,04 %. Расчетное время регулирования составило 0,081 с. Время, затраченное устройством для идентификации объекта, составило менее 2 с.

Следующим шагом является формирование желаемой модели синтезируемой системы по прямым показателям качества. В эксперименте задано 2 набора требований к системе, которые представлены в таблице. На их базе сформированы желаемые ПФ систем и рассчитаны регуляторы. На основании ПФ объекта, желаемой ПФ системы и (2) получаем регулятор полного порядка.

Далее осуществляется переход от непрерывной модели регулятора к дискретной на основе билинейных преобразований [9]. Это обусловлено тем,

что микроконтроллер работает дискретно. Поэтому формирование управляющего сигнала с микроконтроллера осуществляется с помощью дискретного регулятора.

На рис. 4, 5 представлены реакции систем 1, 2 (таблица) на ступенчатое воздействие.

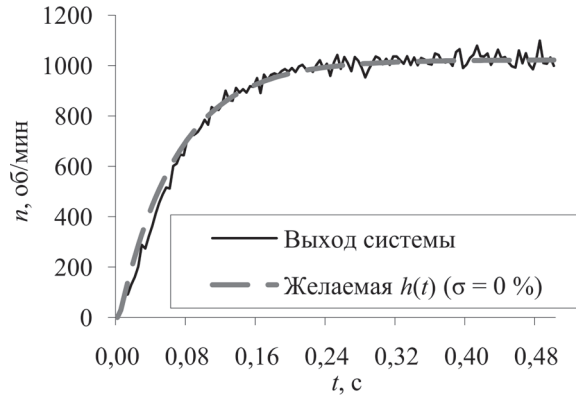


Рис. 4. Результаты синтеза регулятора системы 1

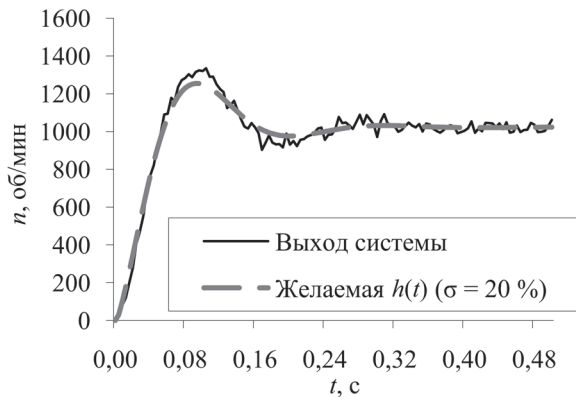


Рис. 5. Результаты синтеза регулятора системы 2

Из приведенных графиков можно сделать вывод, что для заданного объекта регулирования используемые алгоритмы первоначальной автонастройки на основе метода динамической компенсации дают удовлетворительные результаты. Получаемый регулятор полного порядка способен не только обеспечить устойчивость системы, но и удовлетворительные показатели качества переходного процесса, близкие к заданным.

При использовании регулятора полного порядка, помимо получения сложной структуры регуля-

тора, возникает опасность синтеза системы, не обладающей свойствами робастности. Для оценки таких свойств проведены эксперименты над синтезированной системой управления. Для изменения параметров объекта управления использована переменная нагрузка на валу двигателя. В качестве переменной нагрузки использовались металлические диски. Момент инерции одного диска составляет $0,458 \cdot 10^{-3} \text{ кг} \cdot \text{м}^2$.

Для оценки робастных свойств идентификационный эксперимент проводился для «номинально нагруженного» объекта, суммарный момент инерции на валу которого составляет $2,068 \cdot 10^{-3} \text{ кг} \cdot \text{м}^2$. В результате идентификации получена следующая модель объекта:

$$W_{o2}(s) = \frac{4,321 \cdot 10^{-3} s + 5,126 \cdot 10^{-2}}{6,206 \cdot 10^{-2} s^2 + 1,130 s + 1}.$$

Результаты идентификации представлены на рис. 6.

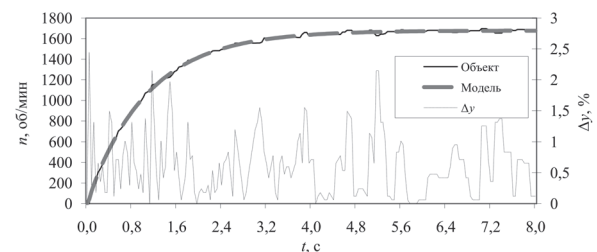


Рис. 6. Результаты идентификации «номинально нагруженного» объекта

Ошибка идентификации по заданному критерию равна 2,44 %. Расчетное время регулирования составило 3,431 с. Время, затраченное устройством для идентификации объекта, составило менее 2 с.

Для анализа робастности реализованной встраиваемой системы управления в эксперименте задано 2 набора требований к системе, которые представлены в таблице (системы 3, 4). На основе этих требований сформированы желаемые ПФ систем и рассчитаны регуляторы. После чего для рассчитанного регулятора снималась реакция системы на ступенчатое воздействие. Далее момент инерции на валу увеличивается, а затем уменьшается на 44 % относительно момента инерции «номинально нагруженного» объекта и снимается переходная характеристика системы.

Таблица. Требования к системам

№ системы	t_p , с	σ , %	Желаемая ПФ	ПФ регулятора
1	0,2	0	$\frac{6,67 \cdot 10^{-2} s + 1}{4,44 \cdot 10^{-3} s^2 + 1,33 \cdot 10^{-1} s + 1}$	$\frac{1,85 \cdot 10^{-3} s^2 + 9,44 \cdot 10^{-2} s + 1}{2,25 \cdot 10^{-4} s^2 + 3,37 \cdot 10^{-3} s}$
2	0,2	20	$\frac{1,38 \cdot 10^{-2} s + 1}{9,24 \cdot 10^{-4} s^2 + 2,77 \cdot 10^{-2} s + 1}$	$\frac{3,84 \cdot 10^{-4} s^2 + 4,16 \cdot 10^{-2} s + 1}{4,67 \cdot 10^{-5} s^2 + 7,01 \cdot 10^{-4} s}$
3	4,0	0	$\frac{1,33 s + 1}{1,78 s^2 + 2,67 s + 1}$	$\frac{8,28 \cdot 10^{-2} s^3 + 1,57 s^2 + 2,46 s + 1}{7,68 \cdot 10^{-3} s^3 + 9,69 \cdot 10^{-2} s^2 + 6,84 \cdot 10^{-2} s}$
4	4,0	20	$\frac{2,77 \cdot 10^{-1} s + 1}{3,70 \cdot 10^{-1} s^2 + 5,54 \cdot 10^{-1} s + 1}$	$\frac{1,72 \cdot 10^{-2} s^3 + 3,75 \cdot 10^{-1} s^2 + 1,41 s + 1}{1,60 \cdot 10^{-3} s^3 + 2,01 \cdot 10^{-2} s^2 + 1,42 \cdot 10^{-2} s}$

Обозначения: t_p – желаемое время регулирования; σ – желаемое перерегулирование.

На рис. 7, 8 представлены реакции систем 3, 4 (таблица) на ступенчатое воздействие.

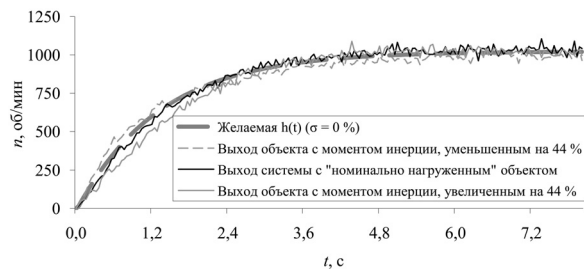


Рис. 7. Результаты работы синтезированного регулятора системы 3



Рис. 8. Результаты работы синтезированного регулятора системы 4

СПИСОК ЛИТЕРАТУРЫ

1. Методы классической и современной теории автоматического управления: в 5-ти т.; 2-е изд., перераб. и доп. Т. 3. Синтез регуляторов систем автоматического управления / под ред. К.А. Пупкова, Н.Д. Пупкова. — М.: Изд-во МГТУ им. Н.Э. Баумана, 2004. — 616 с.
2. Ключев А.С., Карпов В.С. Синтез быстродействующих регуляторов для объектов с запаздыванием. — М.: Энергоатомиздат, 1990. — 176 с.
3. Алексеев А.С., Антропов А.А., Гончаров В.И., Замятин С.В., Рудницкий В.А. Вещественный интерполяционный метод в задачах автоматического управления. — Томск: Изд-во ТПУ, 2008. — 217 с.
4. Белихмайер М.Я., Гончаров В.И. Синтез корректирующих устройств систем автоматического управления на основе равномерного приближения // Автоматика и телемеханика. — 1997. — № 5. — С. 3–11.
5. Ротач В.Я. Теория автоматического управления. 5-е изд., перераб. и доп. — М.: Издательский дом МЭИ, 2008. — 396 с.
6. Nuapett Sarasiri, Arthit Srikaew, Sarawut Sujitjorn. Dynamic Compensation of Hard-Disk R/W Head and Head-Stack // WSEAS Transactions on Systems. — 2010. — V. 9. — Iss. 7. — P. 764–773.
7. Анализ и оптимальный синтез на ЭВМ систем управления / под ред. А.А. Воронова, И.А. Огурка. — М.: Наука, 1984. — 343 с.
8. Курганкин В.В., Замятин С.В., Алексеев А.С. Применение встраиваемых систем управления для решения задачи идентификации // Известия Томского политехнического университета. — 2011. — Т. 319. — № 5. — С. 39–42.
9. Стародубцев Д.О. Билинейное преобразование // Википедия — свободная энциклопедия. 2006 URL: http://ru.wikipedia.org/wiki/Билинейное_преобразование (дата обращения: 26.11.2012)

Поступила 03.10.2012 г.

Анализ графиков (рис. 7, 8) показал, что разработанная система и алгоритмы, заложенные в ней, способны решать задачу не только первоначальной автонастройки системы, но и задачи управления объектами с переменными параметрами.

Разработанная встраиваемая САУ с функцией начальной автонастройки, реализующей принцип динамической компенсации, может успешно применяться для управления широким классом технических объектов на различных производствах и в народном хозяйстве.

Выводы

В результате серии натурных экспериментов над системой управления, объектом в которой выступает двигатель постоянного тока, можно сделать вывод, что регулятор полного порядка, реализованный на базе микроконтроллера, способен успешно решать задачу стабилизации динамических свойств систем с постоянными и переменными параметрами объекта. Параметры объекта могут варьироваться в широком диапазоне значений ($\pm 44\%$ от номинальных).

Работа выполнена в рамках ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009–2013 гг.

УДК 622.333.03:556.3

АВТОМАТИЗАЦИЯ ПРОЦЕССА НАГНЕТАНИЯ ПРИ ГИДРОДИНАМИЧЕСКИХ ИССЛЕДОВАНИЯХ ФИЛЬТРАЦИОННЫХ ХАРАКТЕРИСТИК УГОЛЬНЫХ ПЛАСТОВ

И.Г. Однокопылов, В.В. Гнеушев*, Д.А. Сизиков*, В.В. Шишляев*

Томский политехнический университет

*ОАО «Газпром промгаз», г. Москва

E-mail: odivan@yandex.ru

Совершенствование методик исследований пластов трещино-поровых коллекторов средней и низкой проницаемости является актуальной задачей при выборе технологий разработки метаноугольных месторождений. Рассматриваются технические решения автоматизации процесса управления нагнетанием флюида при проведении гидродинамических исследований угольных пластов на репрессии (инжект-тест).

Ключевые слова:

Гидродинамические исследования скважин, метаноугольные месторождения, угольные пласты, автоматизация, инжект-тест, асинхронный электропривод.

Key words:

Well test analysis, coalbed methane reservoirs, coal seams, automation, injection-falloff test, induction motor drive.

Угольный пласт представляет собой нетрадиционный коллектор трещиновато-пористого типа с двойной пористостью. Микропористость матрицы угля, содержащая на поверхности сорбированный метан, характеризуется значениями проницаемости в 10^{-9} мкм², макропористая проницаемость, сформированная системой трещин эндогенной отделимости и слоистости, — кливажем. Размеры пустот колеблются от нескольких ангстрем до миллиметров и определяют широкий диапазон значений проницаемости углей, достигающих 0,1 мкм². При рассмотрении конкретного геолого-структурного элемента метаноугольного месторождения можно сделать вывод, что на значения проницаемости оказывают влияние много факторов, таких как напряженно-деформированное состояние массива, метаморфизм, микрокомпонентный состав и физико-механические свойства углей.

В условиях открытых гидрогеологических систем, к которым относится большинство метаноугольных месторождений Кузбасса, пластовое давление близко к гидростатическому. При снижении давления в угольном пласте до критического давления десорбции начинается процесс фазового перехода метана из сорбированного состояния в свободное. Газ диффузионным потоком направляется в трещины различного порядка с последующей фильтрацией к скважине. Постепенно понижая уровень воды в скважине ниже наиболее глубокого продуктивного пласта, формируется депрессионная воронка, в пределах которой основным флюидом в пластах является метан. Поэтому фильтрационные свойства вскрытых пластов угля определяют зону влияния скважины, динамику выхода скважины на максимальные дебиты и, как следствие, применяемые технологии разработки месторождений.

Определение проницаемости угольных пластов и оценка состояния околоскважинного пространства на всех этапах поисково-разведочных работ и

разработки метаноугольных месторождений является одной из основных геолого-промысловых задач. Предпочтительным методом получения необходимой информации о фильтрационных характеристиках угольных пластов являются гидродинамические исследования скважин (ГДИС) [1–3].

Данные, полученные при гидродинамических исследованиях, используются как исходные для гидродинамического моделирования, позволяют выбирать технологии интенсификации притока газа и осуществлять выбор оптимального режима работы технологического оборудования скважины на периоды ее освоения и эксплуатации.

При исследовании большого количества объектов в разведочных скважинах важно выбрать оптимальный метод ГДИС, который позволит получить в сжатые сроки корректные значения фильтрационных характеристик продуктивных горизонтов. На начальных этапах разработки месторождений до вызова притока флюида из угольных пластов, когда основная система трещинных каналов заполнена пластовой жидкостью, эффективным методом исследования фильтрационных характеристик угольных пластов является инжект-тест. Целью данной статьи является рассмотрение особенностей проведения гидродинамических исследований на репрессии в средне- и низкопроницаемых коллекторах, а также составление возможных вариантов автоматизации процесса нагнетания, что повысит эффективность и точность инъекционного теста.

Инжект-тест реализуется нагнетанием жидкости в пласт на протяжении определенного периода времени и последующим закрытием скважины и регистрацией кривой изменения забойного давления. В научных работах [4, 5] зарубежные специалисты отмечают, что при проведении таких испытаний фильтрационный поток в системе трещин считается однофазным, интерпретация таких исследований значительно упрощается по сравне-

нию с технологиями ГДИС при многофазном потоке, которые учитывают относительные фазовые проницаемости. Схема компоновки подземного оборудования, используемого при проведении инжест-теста, представлена на рис. 1.

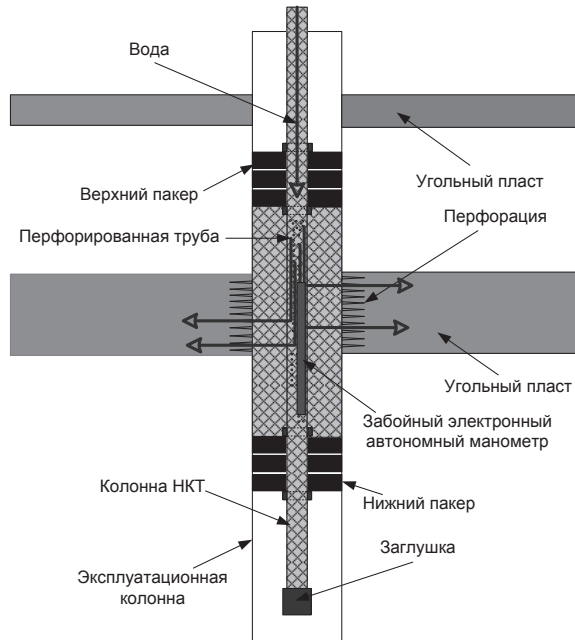


Рис. 1. Схема компоновки подземного оборудования, используемого при проведении инжест-теста

Для осуществления правильной интерпретации гидродинамических исследований, проводимых в скважинах, необходимо иметь надежные входные параметры, определяющие физические свойства горного массива, вмещающего угольные пласты и фильтрующиеся в нем флюиды. При планировании ГДИС необходимо определить время нагнетания жидкости в угольный пласт, время регистрации кривой падения давления, расходы жидкости при закачке.

При математическом описании гидродинамических процессов, происходящих в пласте во время испытания, предполагается, что пласт является горизонтальным с постоянной мощностью, бесконечным по протяженности, однородным и изотропным по своей структуре и без нарушений (экранов, сбросов и т. д.) в зоне исследования.

Первым шагом при планировании инжест-теста является оценка времени влияния объема ствола скважины.

Для снижения влияния объема ствола скважины необходимо проводить заполнение колонны насосно-компрессорных труб (НКТ) (рис. 1) технической жидкостью до устья. При полностью заполненной НКТ, постоянная ствола в скважине определяется по формуле:

$$C = 0,785 D_{\text{НКТ вн.}}^2 L_{\text{пл}} c_{\text{ж}}, \quad (1)$$

где $D_{\text{НКТ вн.}}$ — внутренний диаметр НКТ, м; $L_{\text{пл}}$ — длина компоновки оборудования до исследуемого пласта, м; $c_{\text{ж}}$ — коэффициент сжимаемости флюида, 1/атм.

Для расчета времени, когда эффект влияния объема ствола скважины при регистрации падения давления перестанет быть значимым, применяется следующее уравнение:

$$t = \frac{22105C \exp(0,14s)}{(k \cdot h / \mu)}, \quad (2)$$

где C — коэффициент влияния ствола скважины, м³/атм (1); k — проницаемость пласта, мД; h — мощность пласта, м; μ — вязкость жидкости, сПз; s — скин-фактор.

Для того чтобы уменьшить риск получения непредставительных данных вследствие влияния объема ствола скважины или образования трещин при закачке в уравнение (2) необходимо подставить минимально-ожидаемое значение проницаемости. Длительность периода падения давления должна быть, по возможности, в 5–10 раз больше рассчитанного времени влияния объема ствола скважины.

Для того чтобы определить максимальный расход жидкости при закачке можно воспользоваться следующим уравнением:

$$q_{\text{ж}} = \frac{kh(P_{\text{з}} - P_{\text{пл}})}{18,41B\mu[\ln(R_{\text{др}} / r_{\text{скв}}) + s - 3/4]}, \quad (3)$$

где $P_{\text{пр}}$ — пластовое давление, атм; $P_{\text{з}}$ — максимальное возможное забойное давление в период закачки, атм; $R_{\text{др}}$ — радиус дренирования скважины при исследовании, м.

Необходимо отметить, что если расход жидкости подобран правильно и будет поддерживаться постоянным, то устьевое давление в скважине при проведении испытания будет оставаться избыточным, что устранил изменение коэффициента влияния ствола скважины, а самое главное, упростит последующую интерпретацию данных ГДИС.

В период закачки при создании возмущения в прискважинной зоне необходимо проводить непрерывный мониторинг устьевого и забойного давления и поддерживать постоянный расход жидкости. Поскольку угольные пласты обладают низкой проницаемостью и высокой сжимаемостью порово-трещинного пространства (в пределах от $1,5 \cdot 10^{-4}$ до $0,5 \cdot 10^{-4}$ 1/кПа), то даже незначительная закачка жидкости в исследуемый объект приводит к раскрытию трещинных каналов, в результате чего происходит изменение скорости фильтрации потока жидкости в системе скважина–пласт и замечаются изменения в работе насосного оборудования.

Существующие способы проведения инжест-теста, используемые в настоящее время, предполагают участие оператора, который должен в течение длительного времени с помощью вентиля поддерживать необходимый расход и давление в системе. Та как ранее решения подобных задач не рассматривались, для автоматизации процесса нагнетания при гидродинамических исследованиях авторами были разработаны схемы с использованием мехатронных узлов, которые позволяют с высокой точностью автоматизировать процесс инжестиро-

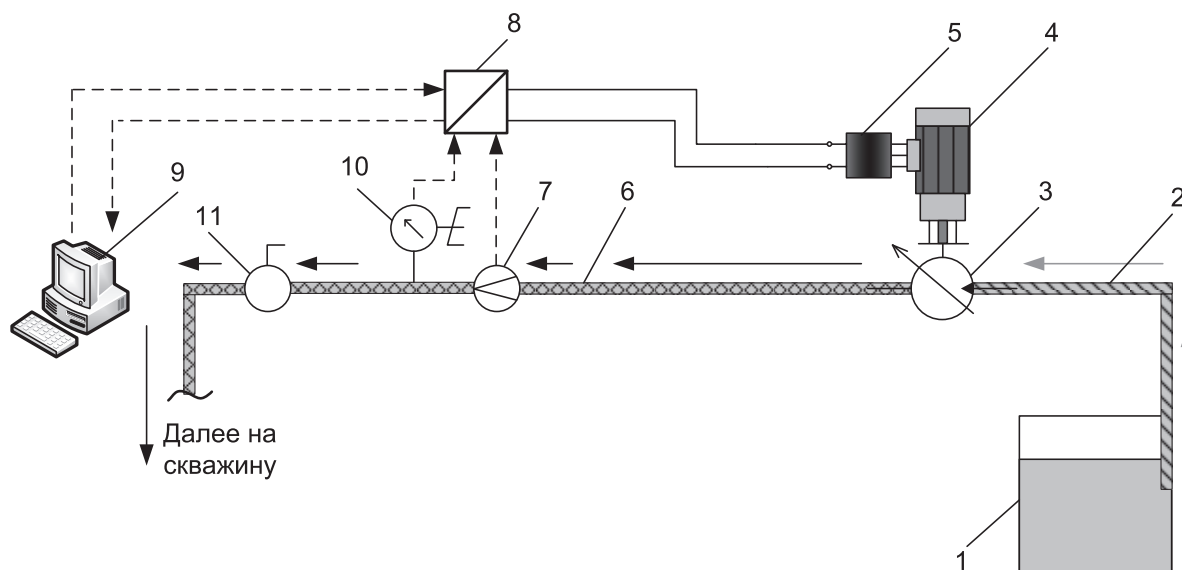


Рис. 2. Схема автоматизации процесса нагнетания при гидродинамических исследованиях скважин методом репрессии с использованием регулируемого насоса высокого давления

вания. Рассмотрим автоматизацию процесса нагнетания подробнее.

Одним из простых решений автоматизации гидродинамических исследований скважин на репрессии является использование регулируемого насоса высокого давления. Схема автоматизации представлена на рис. 2.

Из емкости 1 по трубопроводу низкого давления 2 вода подводится к регулируемому насосу высокого давления 3, приводимого в движение асинхронным двигателем (АД) 4, скорость которого регулируется преобразователем частоты 5. Далее по трубопроводу высокого давления

6 жидкость направляется через расходомер 7 в скважину. Данные с расходомера 7 снимаются, обрабатываются программируемым логическим контроллером 8, в котором формируется задание для преобразователя частоты 5. Для визуального отображения информации в реальном времени и записи данных в определенном формате используется персональный компьютер 9. После достижения в скважине необходимого давления, снимаемого электронным манометром 10, нагнетательный трубопровод закрывается при помощи шарового крана 11. Процесс инжек-тирования закончен.

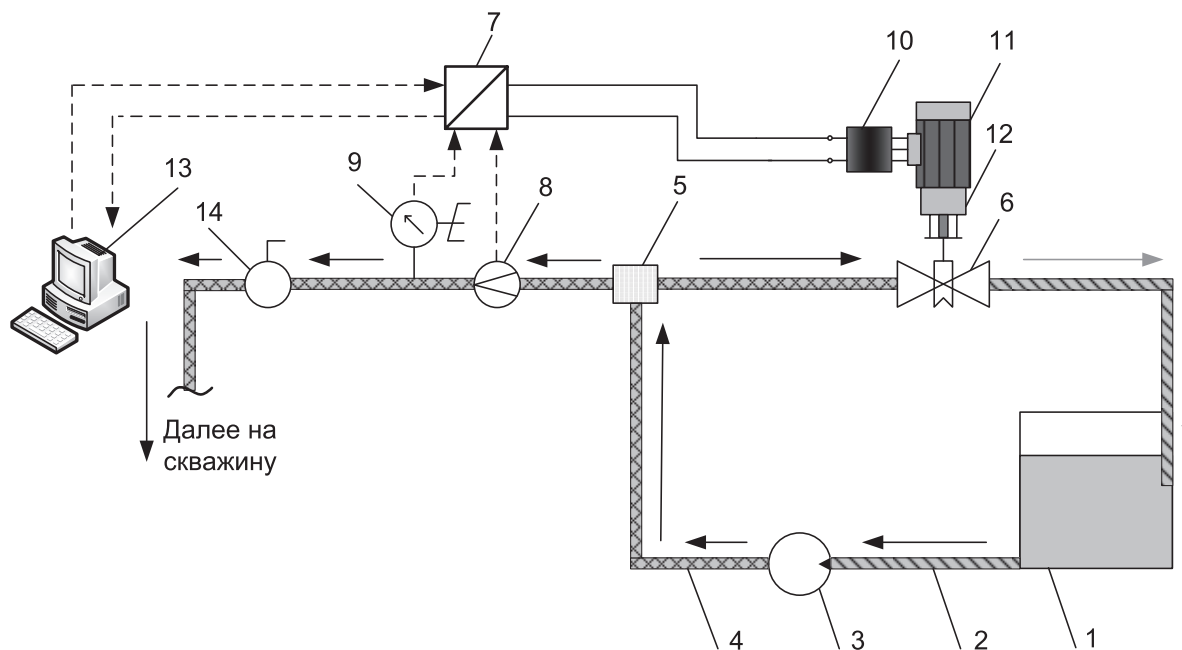


Рис. 3. Схема автоматизации процесса нагнетания при гидродинамических исследованиях скважин (инжект-тест) с использованием проходного регулировочного клапана

Данная схема является простой и логичной, с одной стороны, но она обладает существенным недостатком. Существующие общепромышленные насосы не смогут обеспечить малый расход при высоком давлении. Расход должен фиксироваться расходомером 7 и составлять 50...300 л/ч при давлении более 10 МПа. Проектирование глубоко регулируемого электропривода насоса высокого давления является трудоемкой и неоправданной задачей.

На рис. 3 представлена схема автоматизации гидродинамических исследований скважин методом инжектирования с использованием проходного регулирующего клапана. Схема работает следующим образом.

Из емкости 1 по трубопроводу низкого давления 2 жидкость подводится к нерегулируемому насосу высокого давления 3, далее по трубопроводу высокого давления 4 жидкость подводится к тройнику 5, разделяющему жидкость на два потока – рабочий и сбросной.

Рабочий поток жидкости после тройника направляется через расходомер в скважину. Так как производительность нерегулируемого насоса больше требуемого расхода нагнетания жидкости в скважину, то часть жидкости направляется по обратному трубопроводу через проходной регулировочный клапан 6 в емкость 1. Таким образом, изменяя количество сбрасываемой жидкости, можно поддерживать расход нагнетания жидкости в скважину на определенном уровне.

Контроль расхода нагнетания жидкости, поступающей в скважину, обеспечивает программируемый логический контроллер 7, основываясь на данных, получаемых с расходомера 8 и датчика давления 9. Сигналы снимаются, обрабатываются,

и формируется задание на преобразователь частоты 10, который, в свою очередь, управляет асинхронным двигателем 11. Далее двигатель через понижающий редуктор 12 поворачивает шток регулировочного клапана 6, тем самым изменяя количество проходящей жидкости. Для визуального отображения информации в реальном времени и записи данных используется персональный компьютер 13.

После достижения в скважине необходимого давления, снимаемого электронным манометром 9, нагнетательный трубопровод закрывается при помощи шарового крана 14.

Предложенная схема обладает рядом недостатков, таких как низкий КПД системы, т. к. большая часть жидкости, подаваемая насосом 3, сбрасывается обратно в емкость 1. Расходомер 8 должен быть точным прибором, позволяющим измерять небольшой расход жидкости (50...300 л/ч) при высоком давлении (более 10 МПа). В этом случае приведенная схема позволяет с высокой точностью проводить процесс инжектирования, а электро- и гидрооборудование, необходимое для его реализации, является доступным.

Для исключения дорогостоящего расходомера, который включен в трубопроводе высокого давления, можно видоизменить схему путем переноса расходомера в сбросной трубопровод. Измененная схема представлена на рис. 4. Данная схема позволяет использовать более дешевый расходомер 8, который будет измерять более высокий расход с меньшим давлением, но информацию о реальном расходе жидкости, поступающей в скважину, будем узнавать косвенно. Косвенное измерение расхода несколько снизит точность процесса нагнетания.

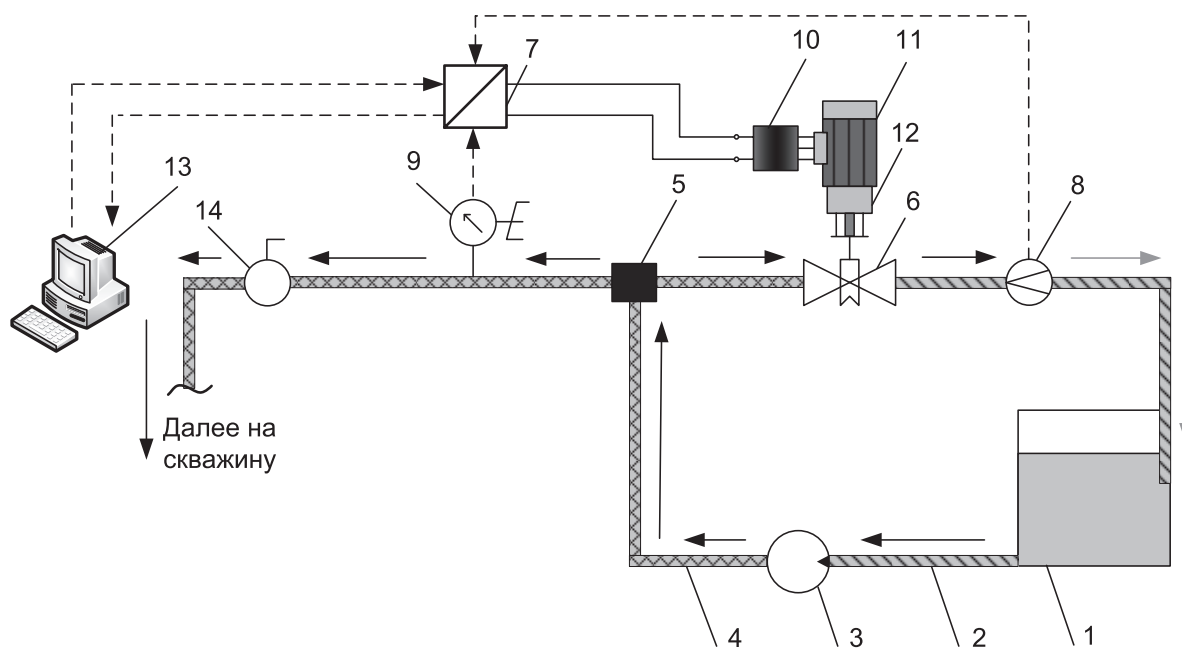


Рис. 4. Схема автоматизации процесса нагнетания при гидродинамических исследованиях скважин (инжект-тест) с использованием проходного регулировочного клапана

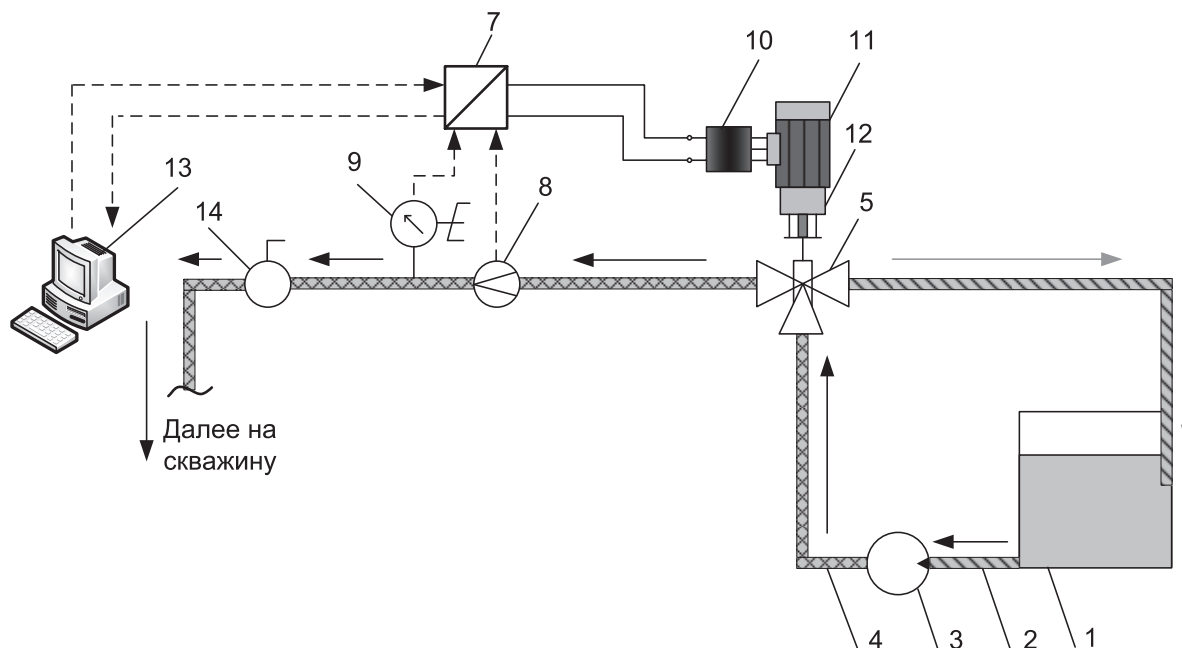


Рис. 5. Схема автоматизации процесса нагнетания при гидродинамических исследованиях скважин (инжект-тест) с использованием регулировочного трехходового клапана

На рис. 5 представлена схема автоматизации гидродинамических исследований скважин нагнетанием с использованием регулировочного трехходового клапана. Тройник 5 и регулировочный клапан 6 (рис. 4) совмещены в одном устройстве – трехходовой клапан 5 (рис. 5), который регулирует разделение потока жидкости на рабочий и сбросной. Данный механический узел облегчает схему автоматизации процесса инжектирования в целом, но является дорогостоящим и уступает схемам, приведенным на рис. 3 и 4.

Выводы

В низкопроницаемых трещинно-поровых коллекторах метаноугольных месторождений для определения проницаемости хорошо зарекомендовал себя инжект-тест, позволяющий наиболее достоверно определять фильтрационные параметры пластов со значительной экономией времени проведения опыта и затрат на исследование.

По опыту интерпретации данных гидродинамических исследований, наиболее успешными признаются те испытания, при которых в период закачки расход жидкости поддерживается постоянным на протяжении всего теста.

Анализ разработанных схем автоматизации гидродинамических исследований скважин методом инжектирования показал в первом приближении, что наиболее простым, но в тоже время эффективным, является использование проходного регулировочного клапана, что позволит с достаточной точностью управлять величиной расхода при проведении испытаний. Для этого необходимо разработать регулируемый электропривод, который позволит бы с высокой точностью изменять и поддерживать количество проходящей жидкости в скважину в ходе испытаний. Применение программируемого логического контроллера позволит вести сбор и обработку информации с датчиков и сформировать соответствующее задание для регулирования положения клапана.

СПИСОК ЛИТЕРАТУРЫ

1. Кузнецов С.В., Кругман Р.Н. Природная проницаемость угольных пластов и методы ее определения. – М.: Недра, 1978. – 122 с.
2. Saulsberry J.L., Schafer P.S., Schraufnagel R.A. A Guide to Coalbed Methane Reservoir Engineering // Gas Research Institute Report GRI-94/0397. – Chicago, Illinois, 1996. – P. 342–345.
3. Останин В.А., Пешков В.Е., Крохалев И.В. Обоснование рационального комплекса гидродинамических исследований водоносных горизонтов // Известия Томского политехнического университета. – 2012. – Т. 321. – № 1. – С. 129–132.
4. Hopkins C.W., Frantz J.H., Flumerfelt R.W., Spivey J.P. Pitfalls of Injection/Falloff Testing in Coalbed Methane Reservoirs // Permian Basin Oil and Gas Recovery Conference. – Midland, Texas, 23–26 March, 1998. – P. 16–20.
5. Taco G., Kamenar A., Edgoose J. Comparison of Permeabilities and Skin Damages Results from Flow Build-up and Injection Falloff Tests in CBM Reservoirs // Asia Pacific Oil and Gas Conference and Exhibition. – Brisbane, Queensland, Australia, 18–20 October, 2010. – P. 6–10.

Поступила 25.01.2013 г.

УДК (004.352.242)

ИССЛЕДОВАНИЕ МЕТОДА НЕПАРАМЕТРИЧЕСКИХ ОЦЕНОК ДЛИНЫ ПЕРИОДА И ПЕРИОДИЧЕСКОЙ СОСТАВЛЯЮЩЕЙ СИГНАЛА

В.В. Марченко, О.Г. Берестнева¹, Д.В. Девярых¹, Е.Ф. Суханова¹

ЗАО «Элекард Девайсез», г. Томск

¹Томский политехнический университет

E-mail: ogb6@yandex.ru

Рассматриваются методы выделения скрытой периодической составляющей сигнала. В качестве основного и наиболее подходящего для использования в данной работе был выбран метод непараметрических оценок длины периода и периодических составляющих, так как именно этот метод позволяет выделять скрытые составляющие из любых периодических сигналов, а не только гармонических. Исследование метода проводилось на модельных сигналах, показана зависимость относительной погрешности метода от количества скрытых периодов и шага дискретизации.

Ключевые слова:

Статистические оценки, показатель разброса, показатель размаха, скрытые периодические составляющие, погрешность.

Key words:

Statistical estimates, spread index, range index, hidden periodic components, inaccuracy.

Скрытые периодические процессы протекают практически во всех областях жизни и затрагивают множество сфер человеческой деятельности. Так, например, задачи выделения скрытых периодических составляющих возникают в геологии, сейсмологии, экономике, медицине. На сегодняшний день изучение методов поиска периодичностей, скрытых в наблюдаемых процессах, является по-прежнему актуальной задачей. Статья является продолжением цикла работ по данной тематике [1–3].

Основными параметрами ритмов являются амплитуда, период и фаза. Выход параметров ритмов за пределы нормы или появление их там, где они раньше не обнаруживались, как правило, связано с изменением свойств и возможной неисправностью системы. Существуют различные методы выделения частотных ритмов, такие как метод Стокса, спектральное представление сигнала с помощью ряда Фурье, корреляционный анализ и метод непараметрических оценок длины периода и периодических составляющих [4].

В работе рассмотрен метод непараметрических оценок длины периода и периодических составляющих, так как он позволяет выделять скрытые составляющие из сигналов любой периодической природы, а не только гармонической.

Целью исследования является изучение свойств и ограничений метода статистической оценки скрытой периодической составляющей на модельных сигналах. Объект исследования – сигналы, предполагающие наличие скрытой периодической составляющей, предмет исследования – периодические свойства сигналов, выделенные методом непараметрических оценок.

Кратко опишем суть выбранного метода. Рассмотрим достаточно широкий класс практически полезных непараметрических оценок длины периода и периодической составляющей во временных рядах. Во многих прикладных задачах рассматривают временной ряд (или случайный процесс)

$$y(t)=x(t)+e(t),$$

где $x(t)$ – детерминированная периодическая функция от времени t , т. е. $x(t)=x(t+T)$ при некотором T , где T – длина периода, а $e(t)$ – «шумы», случайные погрешности, искажающие периодический сигнал. Требуется оценить (минимальную) длину периода $T=T_0$ и периодическую составляющую $x(t)$. При этом не предполагается, что функция $x(t)$ входит в какое-либо параметрическое семейство, например конечных сумм синусов и косинусов, т. е. рассматривается задача непараметрического оценивания (минимальной) длины периода и периодической составляющей сигнала [5].

Пусть рассматриваемые функции $y(t)$, $x(t)$, $e(t)$ определены на отрезке $[0; A]$. При фиксированном T рассмотрим «куски» сигнала $y(t)$ на последовательных отрезках длины T , т. е. на отрезках $[0; T]$, $[T; 2T]$, $[2T; 3T]$,... Удобно ввести последовательность функций на отрезке $[0; T]$, полученную сдвигами этих кусков к началу координат:

$$y_1(t)=y(t), y_2(t)=y(t+T), y_3(t)=y(t+2T), \dots$$

Все они определены на отрезке $[0; T]$. Число этих функций равно числу полных периодов длины T , укладывающихся на отрезке $[0; A]$, т. е. равно целой части числа A/T . В дальнейшем из всех периодов будем рассматривать и оценивать только наименьший [5].

Существуют разные ситуации поведения функции и периодической составляющей:

1. Если $T=T_0$ – истинный период (или кратный ему), и погрешности $e(t)$ отсутствуют, то все введенные в предыдущем абзаце функции совпадают между собой и с периодической составляющей:

$$x(t)=y_1(t)=y_2(t)=y_3(t)=\dots$$

при всех t из $[0; T]$. При наличии погрешностей полного совпадения не будет. Однако отклонения определяются лишь шумами в различные моменты времени. При этом в качестве оценки периодической составляющей $x(t)$ естественно

- взять среднее арифметическое $y_{cp}(t)$ функций $y_1(t), y_2(t), y_3(t), \dots$ [5].
2. Если же T отличается от истинного периода T_0 (и кратных ему величин), то различия функций $y_1(t), y_2(t), y_3(t), \dots$ между собой определяются также и различием значений $x(t)$ в точках, отстоящих друг от друга на интервалы, длина которых кратна T [5].
 3. Если число T/T_0 иррационально, то можно показать, что значения $t+mT \pmod{T_0}$, где m – натуральные числа такие, что $t+mT < A$, асимптотически (при росте A) равномерно заполняют отрезок $[0; T_0]$, а потому при выполнении соответствующих условий регулярности, например непрерывности периодической составляющей сигнала, функция $y_{cp}(t)$ приближается к константе – среднему значению периодического сигнала $x(t)$ [5].
 4. Если же число T/T_0 рационально, то наблюдаем промежуточный случай, в котором $y_{cp}(t)$, как можно показать, приближается к периодической функции с периодом $T = T_0/n$ при некотором натуральном n . Эта функция получена усреднением n последовательных участков длины T_0/n периодического сигнала $x(t)$. Она не является константой, хотя разброс ее значений меньше, чем для исходного периодического сигнала, поскольку T_0 – минимальная длина периода [5].

Для оценивания T целесообразно ввести два показателя: показатель разброса $F(T; Y) = F(T; y_1(t), y_2(t), y_3(t), \dots)$ множества функций $y_1(t), y_2(t), y_3(t), \dots$ на отрезке $[0; T]$ и показатель размаха $G(T; Y) = G(T; y_{cp}(t))$ функции $y_{cp}(t)$ на отрезке $[0; T]$ [5].

Символ Y означает здесь, что показатели разброса и размаха строятся по функции $y(t)$. При этом показатель разброса нацелен на оценку различий в значениях семейства функций при одном и том же значении аргумента. А показатель размаха – на различие значений одной и той же функции при разных значениях аргумента [5].

Для дискретного времени их можно адаптировать двумя способами: либо заменив $\sup$ на $\max$, а интеграл на сумму; либо расширив область определения используемых функций на весь отрезок, например, соединив соседние точки отрезками или используя для заполнения пропусков сплайны более высокого порядка [5].

Ввести показатели разброса $F(T; Y) = F(T; y_1(t), y_2(t), y_3(t), \dots)$ можно разными способами. Пусть $k = [A/T]$. Можно использовать различные функционалы супремумного типа (здесь и далее число слагаемых k не будем указывать в обозначении функционалов). Первым рассмотрим максимальный разброс непосредственно между значениями функций:

$$F_1(T, Y) = \sup \{ |y_i(t) - y_j(t)|, i, j = 1, 2, \dots, k, 0 \leq t \leq T \} \quad [5].$$

Второй функционал супремумного типа будет учитывать не произвольные отклонения, а только отклонения от «средней функции», т. е. иметь вид

$$F_2(T, Y) = \sup \{ |y_i(t) - y_{cp}(t)|, i = 1, 2, \dots, k, 0 \leq t \leq T \}.$$

Третий функционал показывает, какую зону «заметают» значения функций:

$$F_3(T, Y) = \sup \{ |y_i(t)|, i = 1, 2, \dots, k, 0 \leq t \leq T \} - \inf \{ |y_i(t)|, i = 1, 2, \dots, k, 0 \leq t \leq T \}.$$

В качестве показателя разброса представляется полезным рассмотреть то или иное отклонение совокупности функций $Y(q), i = 1, 2, \dots, k$ друг относительно друга. Можно сказать, что эти функции заполняют некую «трубку», которая тоньше всего при истинном значении периода T , а внутри нее проходит периодическая составляющая $X(q) = x(t) = x(qT)$. Естественно рассмотреть различные функционалы интегрального типа. Например, можно проинтегрировать максимум модулей попарных разностей:

$$F_4(T, Y) = \int_0^1 \max \{ |Y_i(q) - Y_j(q)|, i, j = 1, 2, \dots, k \} dq \quad [5].$$

Вместо максимума можно проинтегрировать сумму:

$$F_5(T, Y) = \int_0^1 \sum_{i,j=1}^k |Y_i(q) - Y_j(q)| dq.$$

Как и для функционалов супремумного типа, естественно рассмотреть показатели разброса относительно «средней функции»:

$$F_6(T, Y) = \int_0^1 \max \{ |Y_i(q) - Y_{cp}(q)|, i = 1, 2, \dots, k \} dq,$$

$$F_7(T, Y) = \int_0^1 \sum_{i,j=1}^k |Y_i(q) - Y_{cp}(q)| dq \quad [5].$$

Список показателей разброса можно существенно расширить.

Показатели размаха также можно ввести самыми различными способами. Например, можно рассмотреть такой показатель:

$$G_1(T, Y) = \sup \{ |y_{cp}(t)|, 0 \leq t \leq T \} - \inf \{ |y_{cp}(t)|, 0 \leq t \leq T \}.$$

При практическом использовании рассматриваемых алгоритмов целесообразно учитывать дополнительные особенности реальных временных рядов. В частности, обратим внимание на неустойчивость супремумов по отношению к выбросам (резко выделяющимся наблюдениям) сравнительно с функционалами интегрального типа. Во многих ситуациях временные ряды дают резкие выбросы (всплески), которые затем, как правило, сглаживаются. Поэтому целесообразно в качестве показателей разброса и размаха использовать функционалы интегрального типа [5].

Выберем для реализации показатели разброса и размаха F_1 и G_1 .

Приступим к исследованию выбранного метода на модельных сигналах и, прежде всего, убедимся в его способности находить ритмические составляющие в различных периодических сигналах.

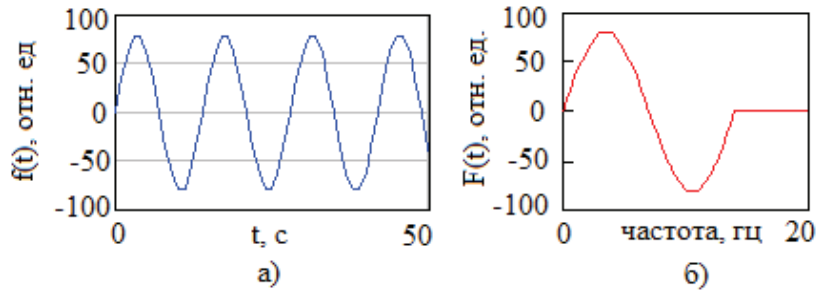


Рис. 1. а) сигнал синуса; б) оценка периодической составляющей сигнала синуса

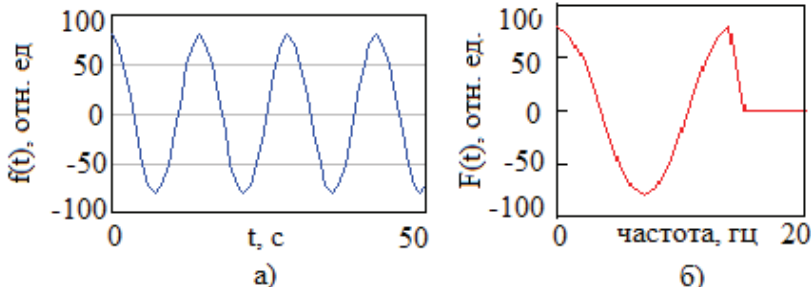


Рис. 2. а) сигнал косинуса; б) оценка периодической составляющей сигнала косинуса

Для этого возьмем гармонические сигналы синуса и косинуса, а также негармонические в виде функций «ступенька», «пила» и тангенса.

Пусть сигнал задан в виде $f(t) = A \sin(2\pi\omega_x t)$, где A – амплитуда сигнала, ω_x – частота, заданная равноотстоящими по аргументу t значениями (рис. 1, а). Выберем следующие значения: $A=80$, $\omega_x=9$ и проверим метод на данном сигнале. Оценка частоты сигнала равна 9,6 Гц (рис. 1, б).

Пусть исследуемая функция $f(t) = A \cos(2\pi\omega_x t)$, где A – амплитуда сигнала, ω_x – частота, задана равноотстоящими по аргументу t значениями. Вы-

берем следующие значения: $A=80$, $\omega_x=9$ (рис. 2, а). Оценка частоты сигнала равна 9,6 Гц (рис. 2, б).

Пусть исследуемая функция задана следующим образом

$$f(t) = \begin{cases} A, & \text{если } \text{mod}(t, T/2) > 0,5; \\ A/2, & \text{если } \text{mod}(t, T/2) \leq 0,5, \end{cases}$$

где A – амплитуда сигнала, $\text{mod}(t, T/2)$ – остаток от деления, заданный равноотстоящими по аргументу t значениями (рис. 3, а). Выберем значения $A=10$, $T=1/9$. Оценка частоты сигнала равна 9,9 Гц (рис. 3, б).

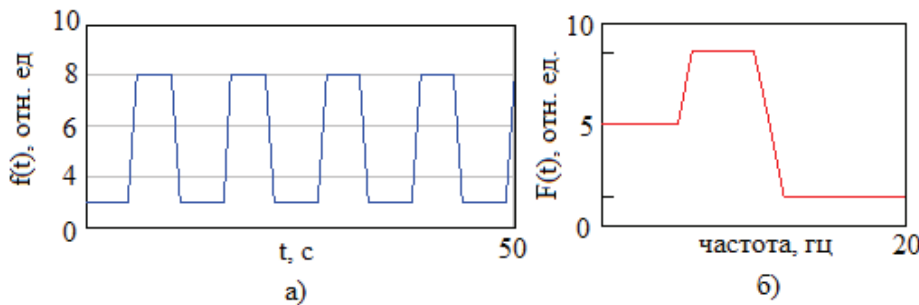


Рис. 3. а) сигнал ступенчатой функции; б) оценка периодической составляющей сигнала ступенчатой функции

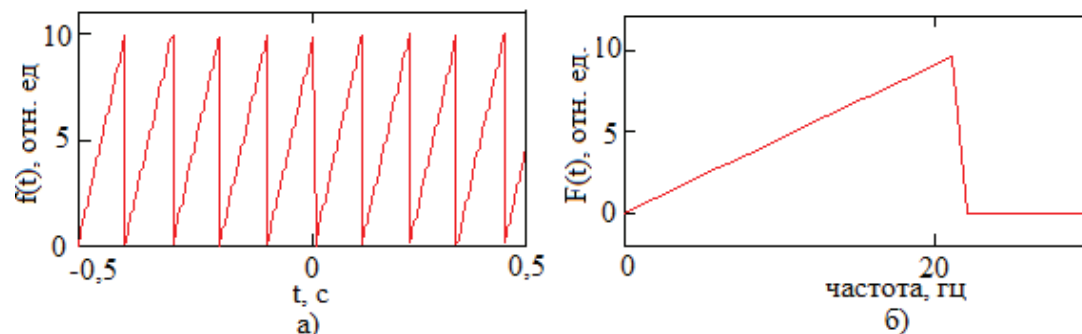


Рис. 4. а) сигнал функции «пила»; б) оценка периодической составляющей сигнала функции «пила»

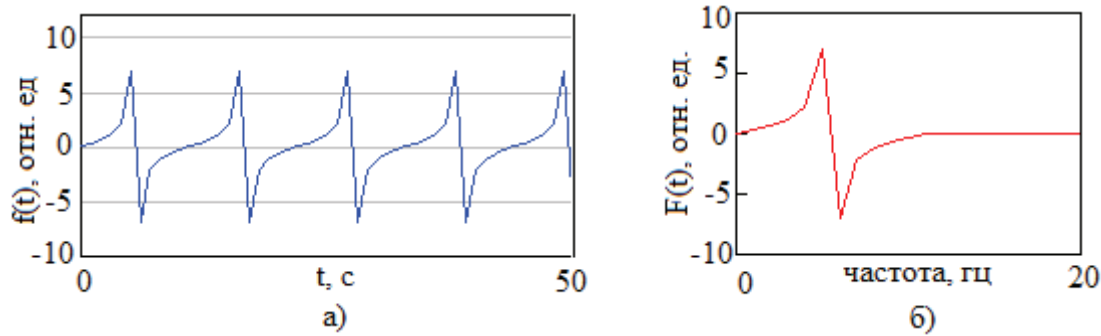


Рис. 5. а) сигнал функции тангенс; б) оценка периодической составляющей сигнала функции тангенс

Пусть исследуемая функция задана в виде «пи-лы» $f(t) = A \cdot \text{mod}(t, T)$, где A – амплитуда сигнала, $\text{mod}(t, T)$ – остаток от деления, заданный равноотстоящими по аргументу t значениями (рис. 4, а). Выберем следующие значения: $A=10$, $T=1/9$. Оценка частоты сигнала равна 9,4 Гц (рис. 4, б).

Пусть исследуемая функция $f(t) = A \cdot \text{tg}(2\pi\omega_x t)$, где A – амплитуда сигнала, ω_x – частота, заданная равноотстоящими по аргументу t значениями (рис. 5, а). Выберем следующие значения: $A=10$, $\omega_x=9$. Оценка частоты сигнала равна 9,9 Гц (рис. 5, б).

Проведем исследование зависимости точности обнаружения скрытых параметров от количества периодов, содержащихся в реализации процесса, и шага дискретизации.

Так как гармонические сигналы является наиболее распространенными среди периодических сигналов, встречающихся в повседневной жизни, для дальнейших исследований возьмем функцию синуса.

Исследуем функцию $f(t) = A \cdot \sin(2\pi\omega_x t)$, где A – амплитуда сигнала, ω_x – частота, заданная равноотстоящими по аргументу t значениями. Выберем следующие значения: $A=80$, $\omega_x=9$.

Известно, что для дискретизации сигнала без потери информации согласно теореме Котельникова частота отсчетов должна быть в 2 раза выше верхней граничной частоты спектра сигнала. В качестве оптимального шага, позволяющего, не сохраняя лишних данных, восстановить сигнал с любой заданной точностью, возьмем значение $\Delta t_{\text{опт}} = 0,5(f_{\text{max}})^{-1}$, где f_{max} – максимальная частота, содержащаяся в спектре сигнала, измеренная в герцах.

Из табл. 1 видно, что при уменьшении шага дискретизации величина относительных погрешностей по частоте также уменьшается. При шаге дискретизации в 5–6 раз меньше оптимального, метод достигает оптимальной точности ~90 %.

Очевидно, что длительность анализируемого сигнала влияет на количество периодов, содержащихся в реализации. Поэтому интересно выяснить, как зависит точность обнаружения скрытых параметров от количества периодов, содержащихся в исследуемой реализации процесса.

При найденном оптимальном шаге дискретизации определим минимальную длину реализации. Результаты исследования представлены в табл. 2.

Таблица 1. Зависимость относительной погрешности метода непараметрических оценок от шага дискретизации

Шаг дискретизации	Погрешность по частоте, %
$\Delta t = \Delta t_{\text{опт}}$	33,33
$\Delta t = \Delta t_{\text{опт}}/2$	33,33
$\Delta t = \Delta t_{\text{опт}}/3$	20
$\Delta t = \Delta t_{\text{опт}}/4$	14,28
$\Delta t = \Delta t_{\text{опт}}/5$	11,11
$\Delta t = \Delta t_{\text{опт}}/6$	9,08
$\Delta t = \Delta t_{\text{опт}}/7$	7,69
$\Delta t = \Delta t_{\text{опт}}/8$	6,67
$\Delta t = \Delta t_{\text{опт}}/9$	5,88
$\Delta t = \Delta t_{\text{опт}}/10$	5,27
$\Delta t = \Delta t_{\text{опт}}/11$	4,77

Анализируя полученные результаты можно сказать, что для точного выделения скрытой гармоник в исходной реализации должно содержаться, по крайней мере, два скрытых периода. Таким образом, следуя значениям табл. 2, можно сделать вывод, что погрешность по частоте метода непараметрических оценок составляет 9,09 %.

Таблица 2. Зависимость относительной погрешности метода непараметрических оценок от количества скрытых периодов

Количество периодов	Погрешность по частоте, %
1	200
2	9,09
3	9,09
4	9,09
5	9,09

Стоит также отметить, что следует рациональным образом выбирать длину анализируемой зависимости. Помимо общего увеличения объема вычислений, анализ излишне длинного участка невыгоден еще и потому, что при неустойчивом периоде смешиваются в кучу все участки, где период изменяется, что приводит к неубедительным результатам.

Основные результаты, полученные в работе.

1. Выделены основные свойства метода: универсальность в смысле применимости к различным сигналам, а также гибкость за счет использования различных показателей разброса и размаха.

2. Проведено исследование метода на модельном сигнале на зависимость точности обнаружения скрытых параметров:
 - 2.1) шаг дискретизации целесообразно брать в 5–6 раз меньше шага, определяемого по теореме Котельникова, так как дальнейшие уменьшения шага не приводят к существенному уменьшению погрешности;
 - 2.2) для точного выделения скрытой гармоники в исходной реализации должно содержаться, по крайней мере, два скрытых периода, при этом погрешность по частоте метода непараметрических оценок составляет 9,09 %.
3. Наиболее точную оценку периодических сигналов имеет функция «пила». Оценка частоты сигнала данной функции составляет 9,4 Гц. Сигналы синуса и косинуса также имеют достаточно точную оценку, составляющую 9,6 Гц. Оценки частот сигналов тангенса и ступенчатой функции оказались менее точными, их оценки составили 9,9 Гц.
Работа выполнена при финансовой поддержке РГНФ, проект № 12–06–12057в «Создание системы алгоритмических и программных средств обработки, представления и анализа экспериментальных данных в социальных и медицинских исследованиях».

СПИСОК ЛИТЕРАТУРЫ

1. Абдулкина Н.Г., Алайцева С.В., Константинова Л.И., Кохановская Ю.Г., Кочегуров В.А., Марченко В.В., Степаненко Н.П. Применение геометрического метода анализа фазового портрета для оценки биоэлектрической активности головного мозга у подростков с дисфункцией гипоталамо-гипофизарно-тиреоидной системы // Вестник новых медицинских технологий. – 2009. – Т. 16. – № 1. – С. 14–17.
2. Кочегуров В.А., Константинова Л.И., Марченко В.В. Использование фазового отображения ЭЭГ для контроля состояния здоровья пациентов с тиреопатологией // Известия Томского политехнического университета. – 2011 – Т. 319. – № 5. – С. 125–129.
3. Берестнева О.Г., Пеккер Я.С. Выявление скрытых закономерностей в сложных системах // Известия Томского политехнического университета. – 2009. – Т. 315. – № 5. – С. 138–143.
4. Магазинникова А.Л. Основы цифровой обработки сигналов. – Томск: Томский межвузовский центр дистанционного образования, 2002. – 129 с.
5. Орлов А.И. Прикладная статистика. – М.: Изд-во «Экзамен», 2004. – 190 с.

УДК 528.854.2

ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ В ЗАДАЧЕ ПРОСТРАНСТВЕННОГО МОДЕЛИРОВАНИЯ ИЗМЕНЕНИЙ ЛАНДШАФТНОГО ПОКРОВА

А.В. Замятин, А.А. Афанасьев

Томский политехнический университет
E-mail: zamyatin@tpu.ru, afanasyevaa@tpu.ru

Обсуждается задача моделирования изменений ландшафтного покрова, основанная на использовании марковских цепей и аппарата клеточных автоматов, увеличенная производительность решения которой достигается применением различных вариантов параллельной обработки данных при моделировании. Приведены результаты численных экспериментов по оценке различных вариантов параллельной обработки, полученные с использованием дорогостоящего суперкомпьютерного кластера и кластера из недорогих ПЭВМ в локальной сети, включая оценку параллельного ускорения, параллельной эффективности и общей производительности кластеров.

Ключевые слова:

Моделирование изменений ландшафтного покрова, параллельные вычисления, марковская цепь, клеточный автомат.

Key words:

Land cover changes modeling, parallel computing, Markov chain, cellular automata.

Введение

Один из наиболее перспективных и широко применяемых на практике подходов к моделированию, реализующих принципы построения систем с возможностями прогнозирования изменений ландшафтного покрова, основан на стохастических моделях, клеточных автоматах (КА) [1, 2] и аппарате марковских цепей и позволяет реализовать поведение сложных природно-территориальных комплексов и строить прогнозную тематическую карту T' на некоторый момент времени t в будущем на базе разновременных тематических карт T^2 и T^1 на моменты времени t_2 и t_1 , матрицы вероятностей переходов $P=[p_{ij}]$ (стохастической матрицы) из типа (класса) ω_i в тип ω_j , матрицы $M_\Phi=[m_{ij}^\Phi]$ фактического числа переходов элементов на отрезке времени $\Delta t_2 \in [t_2; t_1]$ и матрицы $M_{\text{ож}}=[m_{ij}^{\text{ож}}]$ ожидаемого числа переходов элементов на отрезке $\Delta t_1 \in [t_1; t]$, а также с помощью сравнительно простых правил функционирования КА за десятки и сотни итераций алгоритма моделирования, требующих существенных вычислительных затрат [3–5]. Тематическую карту, на основе которой осуществляются последовательно необходимые преобразования (например, карта T^1 на момент времени t_1), будем называть базовой тематической картой. Период времени, соответствующий Δt_2 , называют наблюдаемым, а Δt_1 – прогнозируемым.

Правила функционирования КА могут быть сформированы различными способами [6]. В данном случае будем их задавать на вероятностной основе с использованием результирующей вероятности $p_{ij}^{\text{рез}}$, определяемой исходя из набора вероятностных компонентов $p_{ij}^{\text{вер}}$, $p_{ij}^{\text{пр}}$ и $p_{ij}^{\text{доп}}$. При допущении независимости событий, задаваемых $p_{ij}^{\text{вер}}$, $p_{ij}^{\text{пр}}$ и $p_{ij}^{\text{доп}}$, вероятность $p_{ij}^{\text{рез}}$ может быть найдена как

$$p_{ij}^{\text{рез}} = p_{ij}^{\text{вер}} \times p_{ij}^{\text{пр}} \times p_{ij}^{\text{доп}}. \quad (1)$$

Вероятностный компонент $p_{ij}^{\text{вер}}$ может быть определен как $p_{ij}^{\text{вер}} = p_{ij}$ или как $p_{ij}^{\text{вер}} = p_{ij} \cdot n_j$ на основе стохастической матрицы $P=[p_{ij}]$, где n_j – число эле-

ментов типа ω_j в окрестности заданного КА, а p_{ij} определяется как

$$p_{ij} = \frac{n_{ij}}{\sum_{j=1}^M n_{ij}}, \quad (2)$$

где n_{ij} – число элементов, перешедших из состояния ω_i в состояние ω_j , а M – число типов ландшафта.

Вероятностный компонент $p_{ij}^{\text{доп}}$ при наличии соответствующей априорной информации о буферных зонах, расстоянии до сетей инженерных коммуникаций и других аналогичных данных может быть сформирован с помощью вероятностных карт соответствия $Ptm_k = \{p_{ij}^k, k=1, 2, \dots, Sm, Sm \text{ – общее количество карт соответствия, } i=1, 2, \dots, Cn, j=1, 2, \dots, Rn, \text{ а } Ptm_k \text{ – матрица размером } [Cn \times Rn]\}$, построенных с использованием функций пространственного анализа. Если принять значимость (вес) каждого фактора влияния, с учетом которого построена каждая карта соответствия, идентичным для всех k , то вероятность $p_{ij}^{\text{доп}}$ будет определяться как

$$p_{ij}^{\text{доп}} = p_{ij}^1 \times p_{ij}^2 \times \dots \times p_{ij}^{Sm}.$$

Определение вероятностного компонента $p_{ij}^{\text{пр}}$ предлагается реализовать с использованием набора пространственных метрик. Один из возможных вариантов практического применения $p_{ij}^{\text{пр}}$ в моделировании изменений ландшафтного покрова изложен в [7].

Практические особенности формирования матриц, определяющих основные тенденции изменения ландшафтного покрова, заключаются в следующем. Матрицу фактических переходов определим как $M_\Phi=[m_{ij}^\Phi]$, где m_{ij}^Φ – число элементов, перешедших из поверхности типа ω_i в тип ω_j на изображениях T^2 и T^1 соответственно, $i, j=1, \dots, M$, M – число типов поверхности на исследуемой территории. С помощью выражения (2) и значений матрицы M_Φ ($n_{ij}=M_\Phi[i, j]$) получим стохастическую матрицу $P=[p_{ij}]$. Сумма элементов каждой строки матрицы M_Φ содержит информацию о количестве пиксе-

лей, принадлежащих соответствующим типам ландшафтного покрова на момент времени t_2 , а сумма элементов каждого столбца – ту же информацию на момент времени t_1 , причем $t_2 < t_1$.

Для того чтобы из $\mathbf{M}_\Phi$ получить матрицу ожидаемых переходов $\mathbf{M}_{\text{ож}}$, необходимо выполнить следующие действия:

1. Транспонировать $\mathbf{M}_\Phi$ и получить $\mathbf{M}_\Phi^T$.
2. Получить вектор $\mathbf{S}=[s_1, s_2, \dots, s_M]^T$, причем

$$s_i = \sum_{j=1}^M (m_{ij})^T = \sum_{j=1}^M m_{ji}.$$

3. Получить $\mathbf{M}_{\text{ож}}=[m_{ij}^{\text{ож}}]=[s_i \times p_{ij}]$, $i, j=1, \dots, M$.

Как отмечено выше, изложенный подход к построению прогнозных карт с использованием моделирования изменений ландшафтного покрова является многоитерационной вычислительно затратной процедурой. Тем более вычислительные затраты возрастут при увеличении объемов используемых исходных данных (например при возрастании геометрических размеров разновременных растровых карт $\mathbf{I}^2$ и $\mathbf{I}^1$ или при увеличении числа M ландшафтных классов на исследуемой территории). Существенное снижение вычислительных затрат алгоритма моделирования может быть достигнуто путем применения *высокопроизводительных параллельных вычислений* [8–11].

Рассмотрим некоторые существующие возможности модификации алгоритма моделирования изменений ландшафтного покрова, позволяющие существенно увеличить производительность вычислений за счет параллельного варианта исполнения.

Организация параллельных вычислений при моделировании

При разработке параллельного варианта исполнения алгоритма моделирования целесообразно ориентироваться на следующие требования:

- сравнительная простота реализации параллельного варианта, не требующая значительных модификаций традиционного последовательного алгоритма;
- минимизация затрат на передачу данных между вычислительными узлами при распределенной обработке, увеличивающая общую производительность кластера [12];
- минимизация величины отклонений результатов параллельного алгоритма моделирования (в идеальном случае – их отсутствие) от результатов, которые были бы получены исходным алгоритмом при традиционном последовательном исполнении.

Реализация распределенного алгоритма предполагает наличие управляющего (головного) вычислительного узла (ВУ) (p_0) и N ВУ для параллельной обработки данных ($p_i, i=1, 2, \dots, N$). Поэтому логику возможных вариантов параллельного алгоритма моделирования будут определять варианты распределения данных между головным и прочими ВУ.

В самом простом варианте алгоритм моделирования оперирует следующими данными:

- *исходными изображениями* (тематическими картами): $\mathbf{T}^{t-2}$ и $\mathbf{T}^{t-1}$, полученными на моменты времени $(t-2)$ и $(t-1)$ соответственно;
- *матрицами тенденций изменений*, рассчитываемыми на основе $\mathbf{T}^{t-2}$ и $\mathbf{T}^{t-1}$: вероятностей переходов – $\mathbf{P}$, фактических переходов – $\mathbf{M}_\Phi$ и ожидаемых переходов – $\mathbf{M}_{\text{ож}}$, а также матрицей ранжирования – $\mathbf{R}$;
- *прочими параметрами моделирования*, задаваемыми пользователем: порядком d анализируемой окрестности КА, критерием останова K_{stop} , а также величинами наблюдаемого Δt_2 и прогнозного Δt_1 интервалов времени;
- *промежуточными результатами моделирования (изображениями)*: $\mathbf{T}_{(k)}$, полученными в результате преобразований на $(k-1)$ -й итерации с использованием исходных изображений $\mathbf{T}^{t-2}$ и $\mathbf{T}^{t-1}$, матриц $\mathbf{P}$, $\mathbf{M}_\Phi$, $\mathbf{M}_{\text{ож}}$ и $\mathbf{R}$, а также с учетом прочих параметров моделирования; на основе $\mathbf{T}_{(k)}$ строится результирующее изображение $\mathbf{T}^t$ на момент времени t .

Возможны два принципиально различных варианта организации параллельных вычислений при моделировании изменений ландшафтного покрова.

Первый вариант предполагает расчет вероятностных компонентов типа (1) на каждом из ВУ без фрагментации (разделения) и уменьшения геометрического размера исходных данных. Результат расчета на каждом ВУ p_i интерпретируется как значение вероятности $p_{ij}^{\text{рез}}$ и применяется в многоитерационном моделировании при определении правил функционирования КА. Результаты обработки каждым из ВУ p_i объединяются на головном ВУ p_0 тем или иным образом, в зависимости от того, в каком виде представлены результаты. Если результаты моделирования отдельными ВУ представлены в виде вероятностных карт для каждого из M типов ландшафта, результирующая карта может быть найдена путем произведения соответствующих вероятностей в каждой точке изображений. В этом случае тип ландшафта, для которого вероятность максимальна, является искомым результатом. Если результаты представлены в виде конечных тематических прогнозных карт, типы ландшафта в каждой точке результирующей прогноз-ной карты могут быть найдены, например, по мажоритарному правилу.

При такой организации параллельных вычислений моделирование изменений осуществляется независимо различными ВУ и достигается простота практической реализации. Однако при этом в процессе моделирования и обработки данных различными ВУ не учитывается взаимное влияние используемых вероятностных компонентов, что может негативно влиять на точность моделирования и адекватность представления информации о процессах и явлениях ландшафтного покрова на полученных прогнозных картах.

Другой вариант параллельной реализации алгоритма моделирования предполагает расчет вероятностей и моделирование таким образом, чтобы сохранить взаимное влияние вероятностных компонентов. В этом случае следует использовать весь доступный набор данных, которыми оперирует алгоритм моделирования, осуществляя на каждом ВУ моделирование для некоторого фрагмента исследуемой территории.

Поиск вариантов параллельной реализации алгоритма моделирования, удовлетворяющих сформулированному выше условию, показывает практическую сложность фрагментации указанных данных и разделения соответствующих параметров алгоритма по ВУ кластера таким образом, чтобы сохранить логику его традиционного последовательного исполнения. Это связано с тем, что использование геометрически уменьшенных фрагментов данных на каждом ВУ не всегда при моделировании (аналогичном тому, что исполняется традиционным последовательным вариантом алгоритма моделирования) позволяет отразить во фрагменте тенденции ландшафта, характерные для всей исследуемой области.

В связи с этим поиск способов адаптации рассматриваемого алгоритма моделирования для возможности параллельной обработки данных позволяет предложить варианты, характеризующиеся результатами моделирования, отличными в той или иной степени от результатов, которые были бы получены исходным алгоритмом при традиционном последовательном исполнении.

Рассмотрим два основных варианта возможной реализации параллельного алгоритма моделирования изменений ландшафтного покрова — с *собственными* и с *разделяемыми матрицами тенденций изменения* на ВУ кластера.

Вариант с собственными матрицами тенденций изменений

Вариант с *собственными матрицами тенденций изменения* предполагает следующий обобщенный порядок работы:

Шаг 1. На головном узле осуществляется разделение исходных изображений T^{t-2} и T^{t-1} на фрагменты T_i^{t-2} , T_i^{t-1} , $i=1, 2, \dots, N$, где N — число ВУ.

Шаг 2. На каждом p_i осуществляется расчет собственных матриц P_i , $M_{\Phi(i)}$, $M_{\text{ок}(i)}$ и $R_{(i)}$, которые используются при моделировании и построении фрагментов T_i^t результирующего изображения T^t на момент времени t .

Шаг 3. На головном узле осуществляется компоновка фрагментов T_i^t в единое результирующее изображение T^t .

Очевидно, что основным ограничением использования этого варианта параллельного алгоритма моделирования является потребность в обработке исходных изображений T^{t-2} и T^{t-1} , которая позволит при фрагментации изображений сохранить на отдельных p_i ландшафтно-классовую структуру, характерную для исходных изображе-

ний. При моделировании на отдельных ВУ это способствует сохранению тенденций изменения ландшафтного покрова, характерных для не фрагментированных исходных разновременных изображений. Это, в свою очередь, позволит снизить потенциальную величину отклонений результатов параллельного алгоритма от результатов, которые были бы получены исходным алгоритмом при традиционном последовательном исполнении. В противном случае, матрицы тенденций изменения, используемые каждым p_i , будут существенно отличаться от матриц, используемых традиционным последовательным алгоритмом, а, следовательно, обеспечивать соответствующие значительные отклонения в результатах моделирования.

Кроме того, независимая обработка данных различными ВУ может негативно влиять на общую производительность вычислений кластера. Действительно, объем вычислений каждым ВУ может существенно отличаться, а компоновка фрагментов T_i^t в единое результирующее изображение T^t будет выполнена только после завершения обработки данных каждым ВУ.

Вариант с разделяемыми матрицами тенденций изменения

В общем случае этот вариант алгоритма может иметь разновидности в зависимости от вида разделяемых матриц (например, разделяемыми всеми ВУ могут быть матрицы P и/или $M_{\text{ок}}$). Вариант с разделяемой матрицей P предполагает следующий обобщенный порядок работы:

Шаг 1. На головном узле p_0 на основе исходных изображений T^{t-2} и T^{t-1} осуществляется расчет матриц P , M_{Φ} , $M_{\text{ок}}$ и R ; выполняется разделение базового изображения T^{t-1} на фрагменты T_i^{t-1} , формируются матрицы ранжирования $R_i \in R$. Данные (P, T_i^{t-1}, R_i) передаются на соответствующие p_i .

Шаг 2. На каждом ВУ p_i на основе полученных данных осуществляется расчет собственных матриц $M_{\text{ок}(i)}$, выполняется моделирование и построение фрагментов T_i^t результирующего изображения T^t на момент времени t .

Шаг 3. На головном узле осуществляется компоновка фрагментов T_i^t в единое результирующее изображение T^t .

Изложенный вариант алгоритма позволяет ожидать менее значительную величину отклонений результатов параллельного алгоритма от результатов (по сравнению с вариантом алгоритма с собственными матрицами тенденций изменения), которые были бы получены исходным алгоритмом при традиционном последовательном исполнении. Главным образом это связано с тем, что в этом случае используемые каждым p_i матрицы тенденций изменения менее существенно отличаются от матриц, используемых традиционным последовательным алгоритмом.

В случае, если в качестве разделяемой матрицы выбрана $M_{\text{ок}}$, ее элементы могут быть найдены как $M_{\text{ок}} = \{m_{\text{ок}}[l, r] = m_{\text{ок}(i)}[l, r]/N, l, r=1, 2, \dots, M, N$ — число

доступных ВУ, M – число типов ландшафта}, обеспечивая сниженный объем ожидаемых преобразований при моделировании на каждом ВУ.

Для обеспечения еще более значительного соответствия параллельного варианта исполнения его последовательному традиционному аналогу рассмотрим еще одну разновидность алгоритма с разделяемыми матрицами тенденций изменения – с *синхронизацией матриц тенденций изменения*. Такая синхронизация может выполняться через некоторый определенный временной интервал (Δt_{syn}), задаваемый либо временными параметрами моделирования, либо некоторой относительной долей выполненных преобразований (ΔS). Учитывая относительный характер величины ΔS , ее практическое использование при определении интервала синхронизации следует признать более предпочтительным.

Параллельный алгоритм моделирования с разделяемыми матрицами тенденций изменения и их синхронизацией предполагает следующий обобщенный порядок работы:

Шаг 1. На головном узле p_0 на основе исходных изображений T^{r-2} и T^{r-1} осуществляется расчет матриц P , M_{ϕ} , $M_{\text{ож}}$ и R ; выполняется разделение базового изображения T^{r-1} на фрагменты T_i^{r-1} , формируются матрицы ранжирования $R_i \in R$. Данные $(P, M_{\text{ож}}, R, T^r)$ передаются на соответствующие p_i .

Шаг 2. На каждом p_i на основе полученных данных осуществляется моделирование изменений ландшафтного покрова. Через заданные доли преобразования ΔS производим процедуру синхронизации:

- передаем матрицу $M_{\text{ож}(i)}$ с каждого p_i на p_j , $j=1, 2, \dots, N$ при $i \neq j$;
- приводим значения матрицы $M_{\text{ож}(i)}$ в соответствие с выполненными на всех p_i преобразованиях – $M_{\text{ож}(i)} = \{m_{\text{ож}(i)}[l, r] + m_{\text{ож}(j)}[l, r], l, r=1, 2, \dots, M, j=1, 2, \dots, N \text{ при } i \neq j\}$;
- определяем оставшийся объем преобразований – $M_{\text{ож}(i)}^0 = \{M_{\text{ож}}^0[l, r] - m_{\text{ож}(i)}[l, r], l, r=1, 2, \dots, M$, где $M_{\text{ож}}^0$ – матрица ожидаемых переходов $M_{\text{ож}}$, рассчитанная на шаге 1}.

Шаг 3. На головном узле осуществляется компоновка фрагментов T_i^r в единое результирующее изображение T^r .

В приведенном алгоритме матрица $M_{\text{ож}}$ без модификаций рассылается на все ВУ, а затем синхронизируется через заданные доли преобразования ΔS . При задании ΔS необходимо учитывать то, что более частая синхронизация (малое ΔS) потребует более значительных временных и вычислительных ресурсов, а менее частая не позволит поддерживать значения матрицы $M_{\text{ож}}$ адекватно отражающими объем выполненных преобразований всеми ВУ. Очевидно, этот вариант параллельного алгоритма позволит более равномерно обеспечивать загрузку различных ВУ кластера за счет обеспечения актуального состояния значений матрицы $M_{\text{ож}}$. Это позволит исключить ситуации, когда незначительное число из доступных ВУ кластера выполняет значи-

тельную долю вычислительной работы, а «незагруженные» ВУ простаивают в ожидании окончания этой обработки, что негативно влияет на общую производительность кластера. Пересылка целочисленных значений матрицы $M_{\text{ож}}$ размерности M по каналам связи вычислительного кластера не будет существенно влиять на задержки в обработке данных из-за незначительных объемов этих данных. Все эти особенности будут позитивно отражаться на общей производительности вычислительного процесса в независимости от особенностей исходных данных. Это позволяет выбрать данный вариант в качестве основного при проведении экспериментальных исследований.

Экспериментальные исследования

Для оценки предположений об уровне адекватности и точности распределенных версий алгоритма моделирования в сравнении с его традиционным последовательным аналогом распределенные версии алгоритма моделирования с разделяемыми и с собственными матрицами тенденций изменений реализованы на языке C++ с использованием компилятора g++ версии 3.4.4 и распространеного интерфейса параллельного программирования MPI (MPICH 2.0) [13–15].

Постановка задачи исследования. Определим ключевые параметры эффективности параллельных версий выбранного алгоритма моделирования:

- производительность кластеров различной конфигурации (недорогие пользовательские ПЭВМ, объединенные в локальной вычислительной сети невысокой пропускной способности; суперкомпьютер петафлопной производительности; различное число N доступных ВУ);
- ускорение S_N ($S_N = t_{\text{пол}}^{\text{пол}} / t_N^{\text{пар}}$, где $t_{\text{пол}}^{\text{пол}}$ – время работы последовательного алгоритма, $t_N^{\text{пар}}$ – время работы параллельного алгоритма на N ВУ при тех же входных данных) и эффективность E_N ($E_N = S_N / N$) параллельного моделирования на кластерах различной конфигурации [15];
- величины отклонений результатов параллельного алгоритма моделирования от результатов, которые получены исходным алгоритмом при традиционном последовательном исполнении. Используем для такой оценки каппа-индекс согласия (КИС) [16], рассчитываемый по матрице ошибок и широко применяемый для оценки точности классификации при сравнении эталонного и результирующего изображений.

Основные характеристики конфигураций вычислительных кластеров, используемых при проведении численных экспериментов, приведены в табл. 1.

При проведении экспериментов использованы два различных набора тестовых разновременных изображений, позволяющих осуществлять многоитерационное моделирование и построение по различным исходным изображениям серии прогнозных разновременных изображений. Так, в качестве одного набора использованы данные, поставляе-

мые в качестве примерных данных с растровой геоинформационной системой *Idrisi Kilimanjaro*, – файлы *Landuse71.rst* (изображение T^2), *Landuse85.rst* (изображение T^1) и *Landuse91.rst* (изображение T^0).

Таблица 1. Характеристики вычислительных кластеров

Номер конфигурации	Число ВУ	Сеть	Характеристики ВУ		
			Процессор	ОП	ОС
1	1	–	Pentium IV 3 ГГц	4 Гбайт	MS Windows7
2	2–11	FastEthernet, 100 Мбит/с	Pentium IV 3 ГГц	4 Гбайт	MS Windows7
3	1–24	Gigabit Ethernet, 1000 Мбит/с	2×Intel XEON 5150 (2×2 ядра)	8 Гбайт	Novell SLES 10

Эти данные (тематические растровые карты) имеют 9 ландшафтных классов и геометрические размеры – 565 столбцов и 452 строки, пространственное разрешение составляет 60 м. Другой набор разновременных данных представляет детальные (пространственное разрешение 25 м) тематические карты территории Португалии (изображение T^2) 2000 (изображение T^1) и 2006 гг. (изображение T^0), содержащие более 40 ландшафтных классов (Данные предоставлены Португальским географическим институтом в рамках реализации проекта анализа динамики территории Португалии при поддержке Португальского фонда научных исследований (PTDC/CS-GEO/101836/2008).) на 1990 г.). Для практического использования и проведения экспериментов по построению прогнозных карт при различных условиях данные подвергнуты предварительной обработке – с использованием экспертной генерализации количество ландшафтных типов сокращено до 16, а также путем уменьшения пространственного разрешения получен целый ряд разновременных наборов с различными геометрическими параметрами, но идентичной ландшафтно-классовой структурой. Основные характеристики используемого в экспериментальных исследованиях набора данных приведены в табл. 2.

Таблица 2. Характеристики тестовых разновременных данных

Номер набора (изображения T^2 , T^1 , T^0)	Размеры, пикс.	Число классов	Объем, Кбайт
1	271×581	16	154
2	531×1162	16	603
3	791×1774	16	1348
4	1168×2322	16	2422
5	1341×2906	16	3806
6	1646×3488	16	5607
7	1910×4069	16	7590
8	565×452	9	259

Для оценки общей производительности обработки данных, условного параллельного ускорения S_N и эффективности E_N на вычислительных кластерах различной конфигурации (табл. 1) использовались тестовые наборы данных 1–8 (табл. 2), иден-

тичные для последовательной версии алгоритма (табл. 1, конфиг. 1) и для параллельного алгоритма моделирования с синхронизацией матриц тенденций изменения, выбранного выше в качестве основного (табл. 1, конфиг. 2 и 3).

Процесс моделирования осуществлялся традиционно – на основе изображений T^2 и T^1 или T^2 и T^0 выполнялся расчет тенденций изменения ландшафтного покрова и производилось построение серии из 15 прогнозных изображений $T^{(i)} = \{T^i, T^{i+1}, \dots, T^{i+15}\}$.

Для получения статистически достоверных результатов каждый эксперимент повторялся тридцатикратно, а в качестве оценки использовалось среднее значение, полученное по результатам каждого из экспериментов, а доверительная вероятность при этом была принята равной 95 %. По результатам экспериментов ошибка определения среднего значения оцениваемой величины (математического ожидания) не превышала 3...4 %.

Результаты экспериментов

На рис. 1 приведены некоторые результаты оценки общей производительности (в данном случае определяемой через время t , затраченное на выполнение обработки данных и формирование результата), параллельного ускорения и параллельной эффективности для варианта параллельного алгоритма моделирования с синхронизацией матриц тенденций изменения, с использованием набора данных № 5 (табл. 2). Приведенные результаты демонстрируют увеличение производительности обработки данных с ростом числа доступных ВУ в 1,5–4,0 раза.

Следует отметить, что для подмножества тестовых данных с широким диапазоном характеристик достаточным является использование $N \approx 4...6$ ВУ. Это обусловлено тем, что дальнейший рост числа используемых ВУ не обеспечивает значительного увеличения производительности и параллельного ускорения из-за увеличивающегося влияния накладных расходов параллельной обработки данных. Поэтому, определив достаточное N , можно снизить влияние накладных расходов параллельной обработки и увеличить степень рационального использования оборудования кластера.

Важной характеристикой данных, также оказывающей влияние на производительность вычислений последовательного и параллельного вариантов исполнения алгоритма моделирования, является величина геометрического размера исходных разновременных изображений. Для оценки такого влияния проведены соответствующие эксперименты, фрагмент которых представлен на рис. 2 (наборы данных 1–8, табл. 2).

Приведенные на рис. 2 графики демонстрируют нелинейное (близкое к экспоненциальному) возрастающее увеличение затрат времени на производимые вычисления с увеличением геометрического размера исходных изображений. При этом наблюдается существенное возрастание преимущества параллельного варианта алгоритма (рис. 2, б).

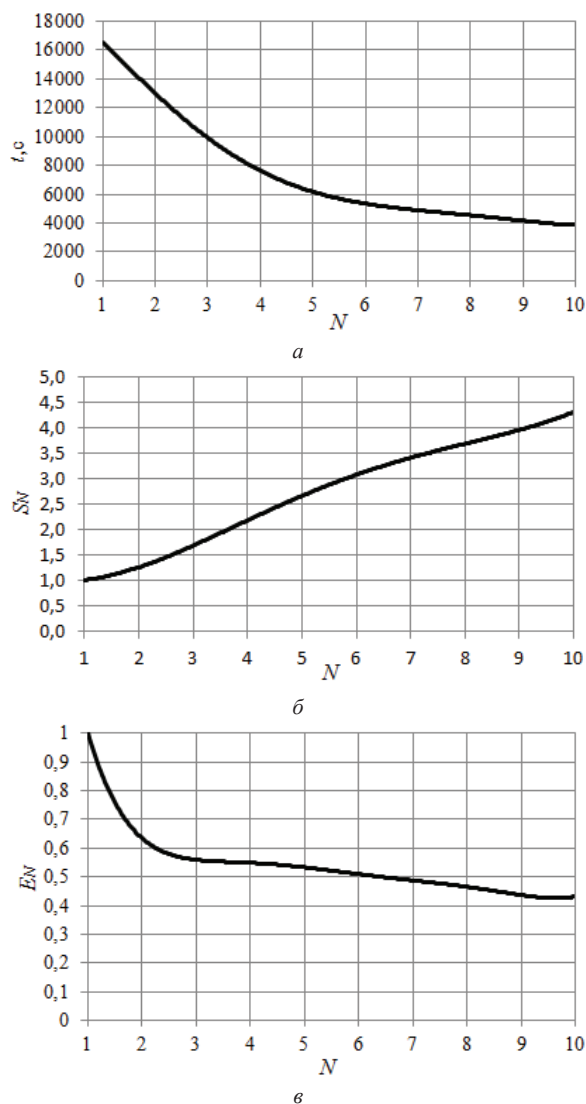


Рис. 1. Результаты оценки параллельных вариантов алгоритма моделирования: а) производительность; б) параллельное ускорение; в) параллельная эффективность

На рис. 3 приведен пример оценки адекватности параллельного варианта алгоритма моделирования его последовательному аналогу (набор данных № 5, табл. 2). Несмотря на то, что для данных с иными геометрическими и ландшафтно-классовыми характеристиками (рис. 3, а, использован набор данных № 3, табл. 2, при условии равенства наблюдаемого интервала прогнозируемому интервалу $|t_1 - t_2| = |t - t_1|$) и различных конфигураций вычислительного кластера (отличных от приведенных в табл. 1) оценки эффективности параллельного исполнения могут несколько отличаться, но в целом эти оценки являются характерными и позволяют сделать некоторые обобщенные выводы и сформулировать практические рекомендации.

В частности, результаты экспериментов (рис. 3, а) показывают, что с увеличением числа задействованных в моделировании ВУ адекватность параллельного варианта алгоритма последовательному

предсказуемо снижается из-за увеличивающегося несоответствия ландшафтно-классовой структуры обрабатываемых фрагментов и исходного изображения, не позволяющего в полной мере отразить тенденции изменения ландшафтного покрова, присутствующие на исследуемой территории в целом. Причем чем большее число ВУ задействовано в вычислениях, тем большее число фрагментов (уменьшенного геометрического размера) с более высокой степенью несоответствия исследуемой территории будет использовано при моделировании. При этом следует отметить, что алгоритм моделирования с синхронизацией матриц тенденций изменения позволяет существенно минимизировать такое отклонение. В приведенном на рис. 3, а примере максимальное отклонение при $N=8$ составило около 8 %. При использовании числа $N \approx 4-6$ ВУ, рекомендованного по результатам экспериментов как наиболее целесообразное с практической точки зрения, такое отклонение будет менее существенным.

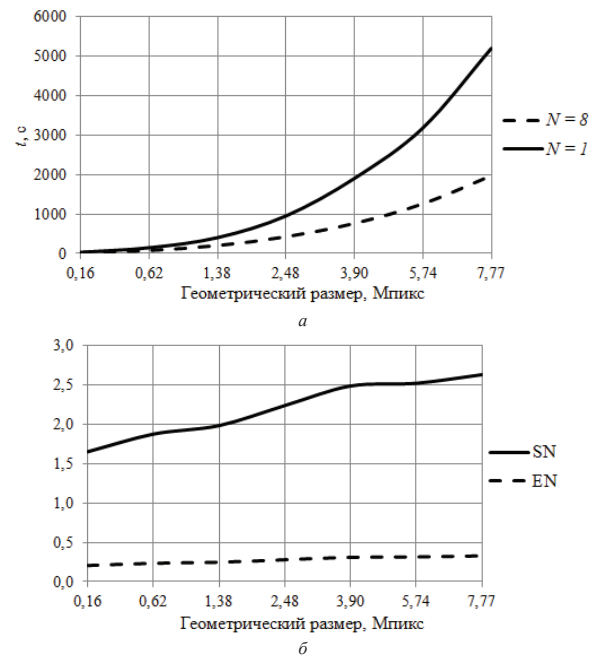


Рис. 2. Показатели эффективности параллельной обработки для данных различного геометрического размера: а) производительность; б) параллельное ускорение и эффективность

С увеличением горизонта прогнозирования адекватность получаемых результатов снижается. В примере на рис. 3, б при различных горизонтах прогнозирования ($Iter_{\text{max}} = 1, 2, \dots, 10$) степень несоответствия результатов, получаемых параллельной версией алгоритма и его последовательным аналогом, также не превысила 8 %.

Таким образом, потенциальную степень несоответствия результатов, получаемых параллельным алгоритмом и его последовательным аналогом для вычислительно сложной задачи пространственного ландшафтного прогнозирования, следует считать практически приемлемой.

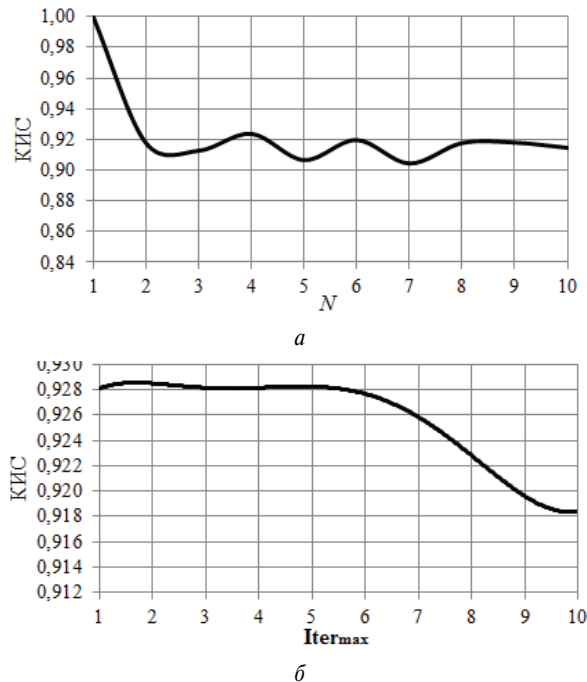


Рис. 3. Пример оценки адекватности параллельного алгоритма моделирования последовательному аналогу: а) при различном числе ВУ; б) при различном прогнозом интервале

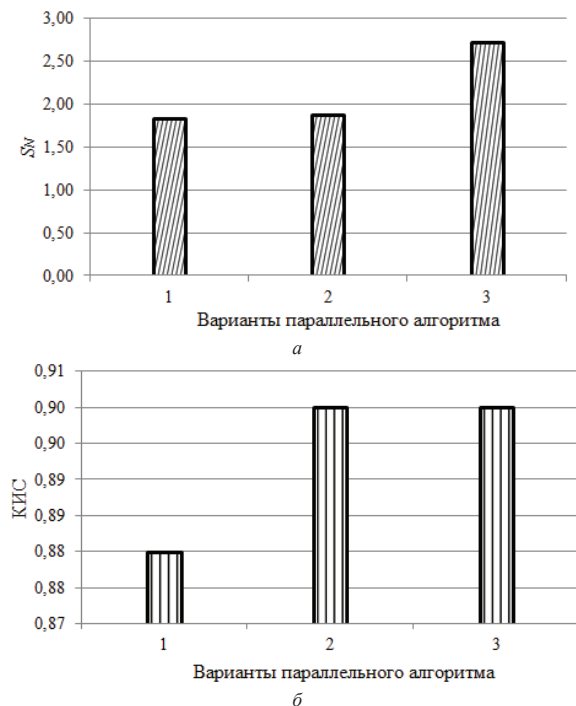


Рис. 4. Параллельное ускорение (а) и адекватность (б) различных вариантов параллельных алгоритмов последовательному исполнению алгоритма моделирования

На рис. 4 приведен фрагмент результатов сравнительного анализа, полученных при параллельном исполнении алгоритма моделирования (набор № 8, табл. 2). Эти результаты демонстрируют очевидное преимущество варианта с разделяемыми матрицами тенденций изменения и их синхронизации.

Скорость исполнения алгоритма моделирования при использовании дорогостоящего суперкомпьютерного кластера выше, чем при использовании недорогого кластера ПЭВМ. Однако параллельное ускорение, полученное на кластере ПЭВМ, будет заметно выше, чем на суперкомпьютерном кластере (рис. 5, набор № 5, табл. 2). Это связано с тем, что ВУ кластера ПЭВМ более «чувствителен» к уменьшению объема входных данных, чем ВУ суперкомпьютерного кластера.

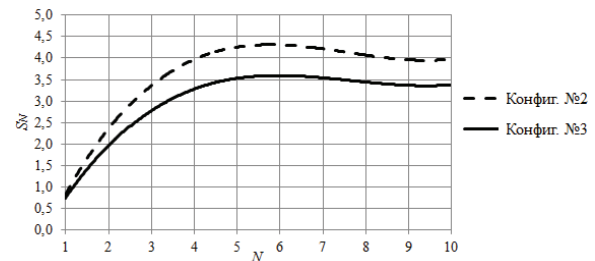


Рис. 5. Параллельное ускорение при использовании кластеров различной конфигурации

Заключение

Для существенного повышения производительности моделирования изменений ландшафтного покрова предложены варианты организации параллельного исполнения алгоритма моделирования, доступные к использованию на дорогостоящих суперкомпьютерах и недорогих кластерах ПЭВМ, отличающиеся тем, что в одном случае расчет вероятностных компонентов осуществляется на каждом ВУ независимо (без фрагментации и уменьшения геометрического размера исходных данных), а в другом предполагается моделирование с сохранением взаимного влияния вероятностных компонентов.

Для случая с сохранением взаимного влияния вероятностных компонентов предложены варианты параллельного исполнения алгоритма моделирования изменений ландшафтного покрова с собственными и с разделяемыми матрицами тенденций изменений на вычислительных узлах кластера. В качестве основного предложен и обоснован выбор варианта алгоритма с разделяемыми и синхронизируемыми матрицами тенденций изменения, характеризующегося более высокой адекватностью традиционному последовательному алгоритму моделирования.

Для оценки предположений об уровне адекватности распределенных версий основного алгоритма моделирования его традиционному последовательному аналогу проведены многочисленные эксперименты на тестовых разновременных данных с использованием дорогостоящей и недорогой конфигураций вычислительного кластера с варьируемым числом вычислительных узлов.

Результаты экспериментов позволили оценить производительность, параллельное ускорение и параллельную эффективность предложенного алгоритма, сформулировать следующие рекомендации по практическому применению параллельного ал-

горитма моделирования изменений ландшафтного покрова:

1. Несмотря на выбор в качестве основного алгоритма моделирования с синхронизацией матриц тенденций изменения, на практике возможны случаи, при которых может быть использован вариант алгоритма с собственными матрицами тенденций изменений (например, в случае низкой пропускной способности доступных каналов связи).
2. При возможности выбора доступного количества ВУ кластера следует учитывать геометрические размеры исходных тематических карт, обеспечивая в полученных после разделения фрагментах максимально возможное соответствие ландшафтно-классовой структуры фрагментов и исходного изображения.

Также показано, что производительность обработки данных с помощью дорогостоящего вычислительного кластера в 3–3,5 раза выше, а с помощью кластера из недорогих типовых ПЭВМ в локальной сети в 4,0–4,2 раза выше, чем аналогичная обработка данных, выполненная последовательным аналогом, при использовании оптимального числа ВУ (4–6). Кроме того, показано, что адекватность алгоритма моделирования с синхронизацией матриц тенденций изменения последовательному аналогу будет различной для различных исходных данных и будет снижаться при увеличении используемых вычислительных узлов и горизонта прогнозирования.

Работа выполнена при поддержке РФФИ (грант № 11-07-00027а) и Госзадания (№ 8.8113.2013).

СПИСОК ЛИТЕРАТУРЫ

1. Clarke K.C., Gaydos J. Loose-coupling a cellular automaton model and GIS: long-term urban growth prediction for San Francisco and Washington/Baltimore // *International Journal of Geographical information science*. – 1998. – V. 12. – № 7. – P. 699–714.
2. Syphard A.D., Clarke K.C., Franklin J. Using a cellular automaton model to forecast the effects of urban growth on habitat pattern in southern California // *Ecological Complexity*. – 2005. – № 2. – P. 185–203.
3. Замятин А.В. Анализ динамики ландшафтного покрова на основе данных дистанционного зондирования Земли // *Исследование Земли из космоса*. – 2006. – № 6. – С. 50–64.
4. Замятин А.В. Подход к моделированию изменений земной поверхности с использованием клеточных автоматов // *Известия Томского политехнического университета*. – 2005. – Т. 308. – № 6. – С. 164–169.
5. Наумов Л.А., Шалыто А.А. Клеточные автоматы. Реализация и эксперименты // *Мир ПЭВМ*. – 2003. – № 8. – С. 64–71.
6. Li X., Yeh A.G.O. Data mining of cellular automata's transition rules // *International Journal of Geographical Information Science*. – 2004. – № 18. – P. 723–744.
7. Cabral P., Zamyatin A. Advanced Spatial Metrics Analysis in Cellular Automata Land Cover Change Modeling // *DYNA, Sede Medellin*. – 2011. – V. 78. – № 170. – P. 42–50.
8. Plaza A.J., Chang C. *High Performance Computing in Remote Sensing*. – Chapman&Hall/CRC, 2008. – 496 p.
9. Kalluri S., JaJa J., Bader D.A., Zhang Z., Townshend J., Fallah-Adl H. High performance computing algorithms for land cover dynamics using remote sensing data // *International Journal of Remote Sensing*. – 2000. – V. 21. – № 6–7. – P. 1513–1536.
10. Ефимов С.С. Обзор методов распараллеливания алгоритмов решения некоторых задач вычислительной дискретной математики // *Математические структуры и моделирование*. – 2007. – № 17. – С. 72–93.
11. Eltgroth P.G., Bolstad J.H. Coupled Ocean / Atmosphere Modeling on High-Performance Computing Systems // *Eighth Society for Industrial & Applied Mathematics Conference on Parallel Processing for Scientific Computing* Minneapolis. – Minneapolis, 1997. – P. 1–10.
12. Ермаков С.Г. Метод устранения необходимости переключения вычислительных узлов при организации параллельной обработки информации // *Информационные технологии*. – 2007. – № 10. – С. 65–68.
13. Rokjin J.P., Daeok Y., et al. A general parallelization approach to improve computation efficiency in a global chemical transport model (GEOS-Chem) // *Geochemical Journal*. – 2010. – V. 44. – P. 323–329.
14. Sunderama V.S., Geist G.A. Heterogeneous parallel and distributed computing // *Parallel Computing*. – 1999. – № 25. – P. 1699–1721.
15. Воеводин В.В., Воеводин Вл.В. *Параллельные вычисления*. – СПб.: БХВ-Петербург, 2004. – 602 с.
16. Brennan R.L., Prediger D.J. Coefficient kappa: some uses, misuses, and alternatives // *Educational and Psychological Measurement*. – 1981. – № 41. – P. 687–699.

Поступила 21.11.2012 г.

УДК 519.673

ПРИМЕНЕНИЕ ЯЧЕЕК ОДНОРОДНОЙ СТРУКТУРЫ В ЗАДАЧЕ ОБНАРУЖЕНИЯ ШУМОВЫХ ГИДРОАКУСТИЧЕСКИХ СИГНАЛОВ НА ОСНОВЕ ИНТЕГРАЛЬНОГО ВЕЙВЛЕТ-СПЕКТРА

А.А. Хамухин

Томский политехнический университет

E-mail: aaxtpu@tpu.ru

Отмечено, что в задаче обнаружения шумовых гидроакустических сигналов на основе интегрального вейвлет-спектра значительное время занимает вычисление непрерывного вейвлет-преобразования, что приводит к запаздыванию обнаружения. Предложена модифицированная схема однородной вычислительной структуры для распараллеливания и ускорения вычисления. Представлены результаты компьютерного моделирования тестовой задачи обнаружения шумовых гидроакустических сигналов на основе интегрального вейвлет-спектра с помощью ячеек однородной структуры.

Ключевые слова:

Вычислительная ячейка, однородная вычислительная структура, параллельные вычисления, шумовые гидроакустические сигналы, вейвлет-спектр, непрерывное вейвлет-преобразование.

Key words:

Computing cell, homogeneous computing structure, parallel computation, noise hydroacoustic signals, wavelet spectrum, continuous wavelet transform.

Введение

Задача обнаружения шумовых гидроакустических сигналов, излучаемых движущимися в воде объектами (подводными лодками или надводными судами), остается актуальной на протяжении десятилетий. Это связано, с одной стороны, с непрерывной работой специалистов над снижением излучаемых объектами шумов и, с другой стороны, с развитием аппаратуры и методов обработки шумовых сигналов [1–3]. В работе [4] подробно рассматривается решение этой задачи на основе одного из новых и относительно недавно разработанных методов вычисления интегрального вейвлет-спектра шумового гидроакустического сигнала. Приводится функциональная схема устройства и результаты компьютерного моделирования, показывающие преимущество предлагаемого метода над традиционно используемым для решения этой задачи методом преобразования Фурье.

Следует отметить, что вычисление интегрального вейвлет-спектра основано на непрерывном вейвлет-преобразовании (НВП) анализируемого сигнала, которое требует больших вычислительных затрат [5]. Алгоритм задачи обнаружения шумовых гидроакустических сигналов на основе интегрального вейвлет-спектра основан на многократном повторении вычисления НВП. В результате весь расчет в задаче обнаружения движущегося в воде объекта занимает значительное время, что приводит к запаздыванию обнаружения и возможной потере его актуальности.

Для ускорения вычисления вейвлет-преобразования (ВП) разработано и продолжает разрабатываться множество методов: ВП на основе быстрого преобразования Фурье, дискретное вейвлет-преобразование, диадное ВП и др. [5]. Но все эти методы связаны с частичной потерей информации при вычислении, что также может снизить эффективность обнаружения шумовых гидроакустических сигналов. Поэтому для данной задачи важно сохранить класси-

ческое непрерывное вейвлет-преобразование и сокращать время его расчета за счет других средств.

Объект и цель исследования

Основными источниками шумовых гидроакустических сигналов являются: энергетическая установка, гребные винты, вспомогательные механизмы (насосы, генераторы и т. п.), гидродинамические эффекты, формирующиеся обтеканием жидкости корпуса объекта. Итоговый шум считают состоящим из двух компонент: широкополосный шум с непрерывным спектром частот и узкополосный шум, имеющий прерывистый (линейчатый) спектр. В задаче обнаружения вторую составляющую гидроакустического шума считают полезным сигналом на фоне аддитивной помехи (первой составляющей гидроакустического шума) [1].

Узкополосный шум состоит из отдельных тональных (синусоидальных) компонент, называемых дискретными составляющими (ДС) спектра шума. Характеристики ДС, обусловленные работой вспомогательных механизмов, обычно стабильны и не зависят от скорости объекта. Частоты и амплитуды дискретных составляющих, вызванные энергетической установкой и гребными винтами, изменяются вместе со скоростью движения объекта. В некоторых случаях дискретные составляющие, инициированные от одного источника, образуют звукоряды, т. е. наборы ДС, частоты которых кратны друг другу [4].

Эти особенности усложняют задачу обнаружения и обуславливают проведение новых исследований и применение новых методов. В частности, высокую эффективность показывает метод, основанный на непрерывном вейвлет-преобразовании шумового гидроакустического сигнала [4], главным недостатком которого являются большие вычислительные затраты.

С другой стороны, известно, что однородные вычислительные структуры (ОВС), основанные на

синхронно работающих вычислительных ячейках одинакового типа, позволяют добиваться максимального увеличения производительности при распараллеливании операций. Специфика применения ОВС состоит в том, что требуется изменение структуры ячейки в зависимости от решаемой задачи. В настоящее время разработано несколько схем ячеек для решения дифференциальных уравнений в частных производных [6–9], а также схемы для вычисления непрерывного вейвлет-преобразования [10, 11].

Цель настоящей работы – рассмотреть особенности применения ОВС в задаче обнаружения шумовых гидроакустических сигналов на основе интегрального вейвлет-спектра.

Модификация схемы однородной вычислительной структуры

Алгоритм обнаружения шумовых гидроакустических сигналов на основе интегрального вейвлет-спектра состоит из следующих этапов [4]:

1. Измерение шумового гидроакустического сигнала, который может состоять из нескольких узкополосных полезных сигналов с центральными частотами 1...20 Гц и аддитивной широкополосной помехи.
2. Оцифровка по времени t поступающего входного сигнала $S(t)$ с шагом дискретизации Δt , который должен удовлетворять требованию теоремы Котельникова–Найквиста:

$$f_d = \frac{1}{\Delta t} \geq 2f_{\max},$$

где f_d – частота дискретизации входного сигнала, $f_{\max}$ – максимальная частота, которой ограничен спектр полезной составляющей входного сигнала.

3. Непрерывное вейвлет-преобразование оцифрованного входного сигнала $S(t)$ и получение значений $W(a_j, b_k)$, где i, j, k – индексы по времени t по масштабу a , по сдвигу по времени b ; N – количество шагов по времени i ; M – количество масштабов j .
4. Вычисление интегрального вейвлет-спектра (ИВС):

$$\hat{W}(a_j) = \frac{1}{N} \sum_{k=1}^N |W(a_j, b_k)|^2 \Delta t. \quad (1)$$

5. Определение местоположений локальных максимумов ИВС, которые и являются обнаруженными полезными сигналами.

В работе [4] пункт 3 вышеизложенного алгоритма реализуется через целый ряд операций: вычисление спектра Фурье входного сигнала, перемножение полученного спектра Фурье с сопряженным базисом отмасштабированных спектров аналитических вейвлетов, вычисление обратного преобразования Фурье от результата последнего перемножения. Такое усложнение предпринято ради применения алгоритма БПФ (быстрого преобразования Фурье), ускоряющего процесс вычисления на последовательном процессоре. Но при распараллеливании вычисления НВП и реализации этого вы-

числения с помощью однородной вычислительной структуры можно избежать этого усложнения и применять прямую формулу [12]:

$$W(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} S(t) \Psi\left(\frac{t-b}{a}\right) dt$$

или для дискретизированного входного сигнала:

$$W(a_j, b_k) = \frac{1}{\sqrt{a_j}} \sum_{t=0}^N S(t_i) \Psi\left(\frac{t_i - b_k}{a_j}\right) \Delta t_i,$$

где $\Psi\left(\frac{t-b}{a}\right)$ – некоторая функция с определенными свойствами (вейвлет).

В работе [12] представлены схемы ячейки и однородной вычислительной структуры, состоящей из таких ячеек для вычисления НВП. Но для задачи обнаружения шумовых гидроакустических сигналов кроме вычисления НВП требуется вычисление интегрального вейвлет-спектра и определение максимумов этого спектра, которые соответствуют обнаруженным дискретным составляющим входного сигнала.

Анализируя возможные схемные решения этой задачи, мы пришли к выводу, что схему самой ячейки для вычисления НВП следует оставить без изменений, а модификацию внести в схему ОВС. Такой подход позволит использовать одни и те же вычислительные ячейки для решения других задач путем перенастройки и дополнения схемы всей однородной вычислительной структуры.

Значения непрерывного вейвлет-преобразования $W(a_j, b_k)$ при вычислении на ОВС получаются в матрице ячеек размером $M \times N$. Количество масштабов M значительно меньше, чем количество отсчетов N анализируемого входного сигнала. Поэтому целесообразно связать ОВС с устройством управления (УУ) параллельно по M каналам.

Можно возложить операции вычисления ИВС по формуле (1) на само УУ, но такой объем вычислений существенно увеличит общее время решения задачи обнаружения. Поэтому на M каналах между устройством управления и ОВС требуется поставить блок для параллельного вычисления интегрального вейвлет-спектра. Таким образом, ввод блока вычисления ИВС на каждом канале позволит передавать в устройство управления массив размером M вместо массива размером $M \times N$. Это возможно в рамках настоящей задачи, поскольку сами значения $W(a_j, b_k)$ после вычисления интегрального вейвлет-спектра для задачи обнаружения не нужны.

Такое решение сократит объем передаваемой для обработки в УУ информации и ускорит весь цикл расчета, поскольку в качестве УУ применяется обычный микропроцессор (с контроллером) и повышенная вычислительная нагрузка на него может снизить эффект от распараллеливания операций в ОВС [9]. Таким образом, на УУ добавляется незначительная нагрузка вычисления локальных максимумов на M числах. Модифицированная схема ОВС представлена на рис. 1.

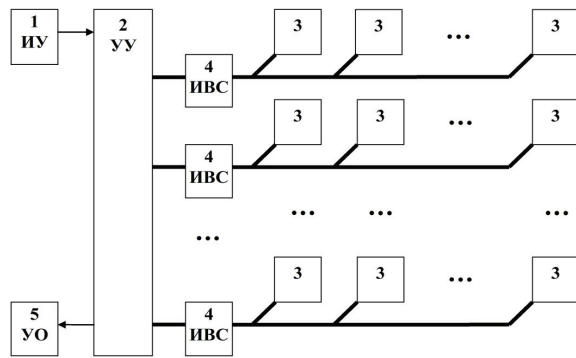


Рис. 1. Функциональная схема однородной вычислительной структуры для решения задачи обнаружения шумовых гидроакустических сигналов на основе интегрального вейвлет-спектра: 1) устройство измерения анализируемого сигнала – ИУ; 2) устройство управления – УУ; 3) ячейка однородной вычислительной структуры; 4) блок вычисления интегрального вейвлет-спектра – ИВС; 5) устройство отображения обнаруженных шумовых гидроакустических сигналов – УО

Анализируемый входной сигнал $S(i)$ измеряется в ИУ и в оцифрованном виде передается в УУ. Устройство управления распределяет этот сигнал по M каналам для передачи их в ячейки однородной вычислительной структуры 3.

При передаче данных от УУ к ячейкам 3, блоки ИВС не производят вычислений и пропускает данные в каналы доступа к ячейкам 3. При передаче данных от ячеек в УУ (а это результат вычисления НВП) блоки ИВС на каждом из M каналов, соединяющих их с линейками из N ячеек, производят операции в соответствии с формулой (1) и передают результаты в УУ. Устройство управления из полученных N чисел определяет локальные максимумы и передает их в устройство отображения обнаруженных шумовых гидроакустических сигналов.

Моделирование задачи обнаружения шумовых гидроакустических сигналов

Компьютерная модель представленной на рис. 1 схемы создана в среде Excel с программиро-

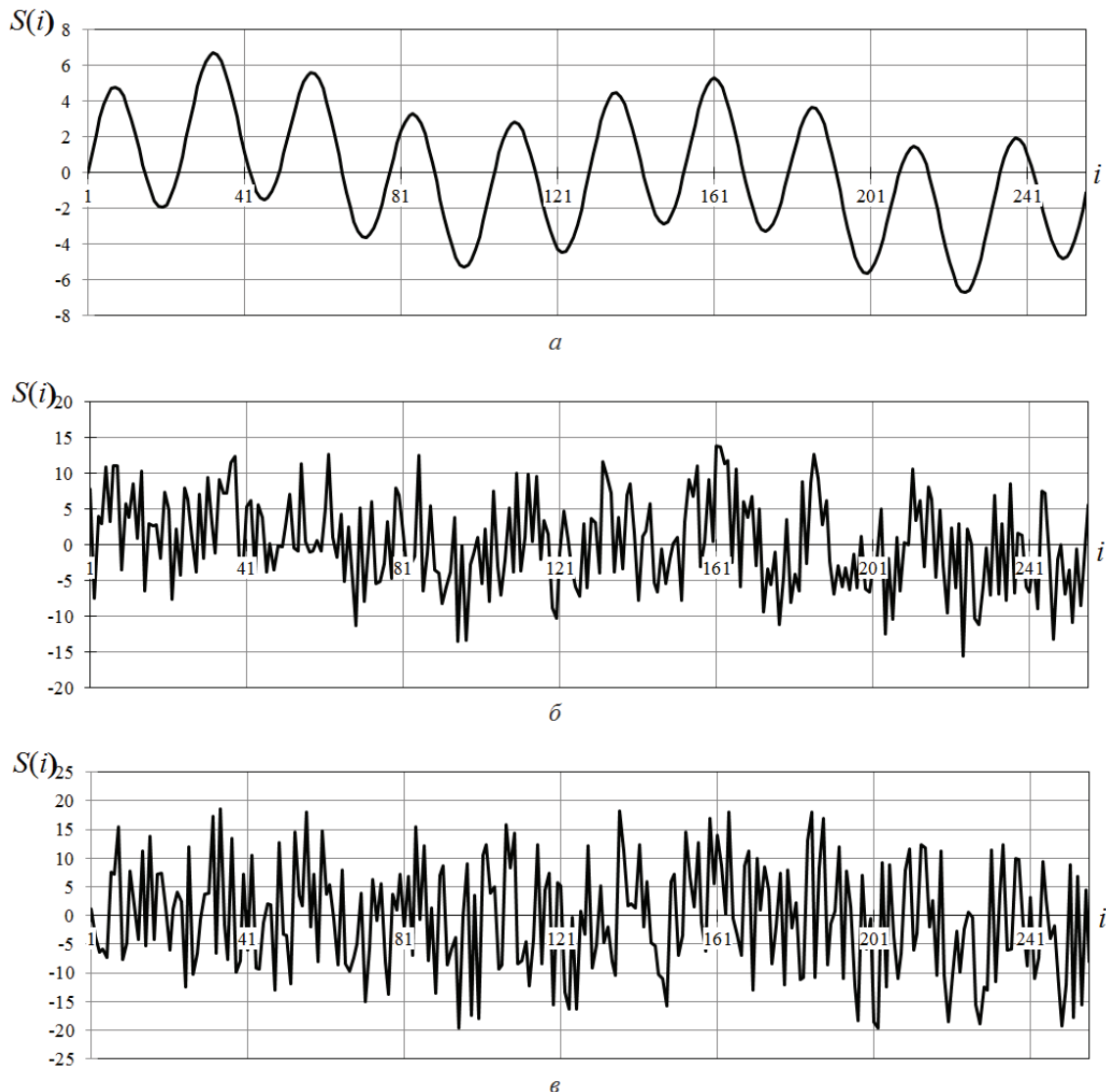


Рис. 2. Анализируемый входной сигнал: а) без помехи; с помехами по амплитуде б) 6; в) 8 дБ

ванием на языке *VBA*. Для тестирования алгоритма был выбран входной сигнал с тремя дискретными составляющими, подлежащими обнаружению. Было выбрано количество отсчетов входного сигнала $N=256$, количество масштабов $M=16$, шаг $\Delta t=1$ с.

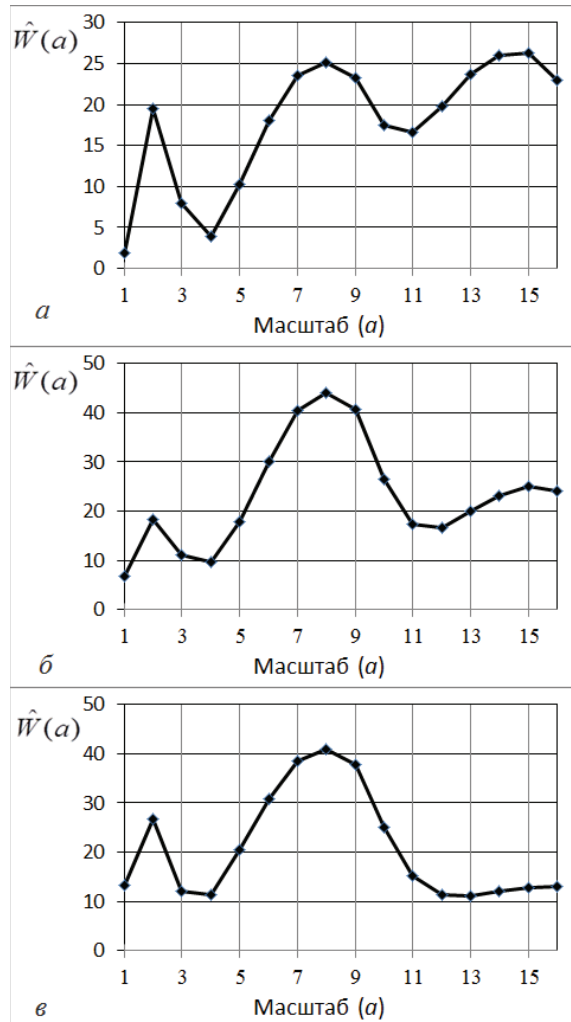


Рис. 3. Расчетный интегральный вейвлет-спектр для входных сигналов: а) без помехи; с помехами по амплитуде б) 6; в) 8 дБ

Реальная задача обнаружения узкополосных шумовых гидроакустических сигналов на основе интегрального вейвлет-спектра решается при нескольких больших количествах ДС, масштабов и отсчетов, но для проверочного тестирования работоспособности схемы мы сочли эти значения достаточными. В качестве материнского вейвлета ис-

пользовалась функция, известная под названием «мексиканская шляпа» (*MHAT*) [11].

Таким образом, моделируемая схема ОВС состояла из матрицы одинаковых вычислительных ячеек размером 16×256 и 16 блоков вычисления интегрального вейвлет-спектра. Расчеты производились на микропроцессоре *Intel Core i7* 3,40 GHz под управлением *Windows 7* (64).

Входной сигнал $S(t)$ моделировался в виде суммы трех гармоник (с частотами 1, 2 и 10 Гц) с аддитивной помехой в виде модели дискретного белого шума. Вид анализируемых входных сигналов с различным уровнем помехи представлен на рис. 2.

Вид полученного на моделируемой схеме ОВС интегрального вейвлет-спектра для представленных выше входных сигналов показан на рис. 3.

Как видно из графиков, расчетный интегральный вейвлет-спектр входного сигнала без помехи содержит три максимума, соответствующих трем гармоникам, из которых он состоит. С увеличением уровня помехи резкость максимумов несколько уменьшается, и при уровне помехи в 8 дБ третьего максимума уже практически нет, т. е. низкочастотная дискретная составляющая спектра входного сигнала не обнаруживается. Такой результат характерен в целом для метода обнаружения дискретных составляющих на основе интегрального вейвлет-спектра и не является следствием параллельного вычисления непрерывного вейвлет-преобразования по предложенной в данной работе схеме.

Выводы

1. Предложена модифицированная схема однородной вычислительной структуры для распараллеливания и ускорения вычисления непрерывного вейвлет-преобразования в задаче обнаружения шумовых гидроакустических сигналов.
2. Модификация заключается в дополнении схемы однородной вычислительной структуры блоками вычисления интегрального вейвлет-спектра, что позволяет сократить объем передаваемой в устройство управления информации и ускорить весь цикл расчета в задаче обнаружения шумовых гидроакустических сигналов.
3. Результаты моделирования на примере решения тестовой задачи обнаружения трех дискретных составляющих шумовых гидроакустических сигналов подтвердили работоспособность и эффективность предложенной схемы.

Работа выполнена в рамках государственного задания «Наука».

СПИСОК ЛИТЕРАТУРЫ

1. Бурдик В.С. Анализ гидроакустических систем. – Л.: Судостроение, 1988. – 392 с.
2. Калью В.А. Автоматический мониторинг подводного шума судов с помощью искусственной нейронной сети // Российский фонд фундаментальных исследований, грант № 11–08–00402. 2011. URL: http://www.rfbr.ru/rffi/ru/contest/n_549/o_41432 (дата обращения: 11.09.2012).
3. Калью В.А., Неворотин В.Ю., Правдин А.А. Повышение точности локализации дискретных составляющих спектра шума движущегося транспортного средства // Техническая акустика. – 2012. – № 3. URL: <http://www.ejta.org> (дата обращения: 11.09.2012).
4. Устройство обнаружения узкополосных шумовых гидроакустических сигналов на основе вычисления интегрального вейвлет-спектра: пат. 2367970 Рос. Федерация. № 2007145474/28; заявл. 28.11.07; опубл. 20.09.09, Бюл. № 26. – 27 с.

5. Яковлев А.Н. Введение в вейвлет-преобразования. — Новосибирск: Изд-во НГТУ, 2003. — 104 с.
6. Хамухин А.А. Ячеечная модель устройства для решения дифференциальных уравнений в частных производных // Известия Томского политехнического университета. — 2010. — Т. 316. — № 5. — С. 63–67.
7. Хамухин А.А. Реконфигурирование однородной вычислительной структуры с непрограммируемыми ячейками для решения дифференциальных уравнений в частных производных // Известия Томского политехнического университета. — 2010. — Т. 316. — № 5. — С. 68–72.
8. Хамухин А.А. Модификация ячейки однородной структуры для решения дифференциальных уравнений в частных производных с переменными коэффициентами // Известия Томского политехнического университета. — 2011. — Т. 318. — № 5. — С. 37–41.
9. Хамухин А.А. Адресуемая ячейка однородной вычислительной структуры для решения дифференциальных уравнений в частных производных // Известия Томского политехнического университета. — 2012. — Т. 320. — № 5. — С. 25–30.
10. Хамухин А.А. Математическая модель ячейки однородной структуры для вычисления непрерывного вейвлет-преобразования // Проблемы информатики. — 2011. — № 5. — С. 87–93.
11. Хамухин А.А. Применение ячеек однородной структуры для вычисления непрерывного вейвлет-преобразования // Известия Томского политехнического университета. — 2010. — Т. 317. — № 5. — С. 149–153.
12. Устройство для вычисления дискретизированного непрерывного вейвлет-преобразования: пат. 2437147 Рос. Федерация. № 2010127068/08; заявл. 01.07.10; опубл. 20.12.11, Бюл. № 35. — 9 с.

Поступила 21.09.2012 г.

УДК 621.312/313: 621.317.31

ИССЛЕДОВАНИЕ ТЕМПЕРАТУРНОГО ПОЛЯ КОАКСИАЛЬНОГО ШУНТА В ПРОГРАММНОМ КОМПЛЕКСЕ ANSYS

Е.В. Бедарева, Е.В. Костин

Томский политехнический университет
E-mail: bedareva@tpu.ru

Проведен анализ распределения температурного поля коаксиального шунта в программном пакете ANSYS. Установлено, что температурное поле коаксиального шунта имеет неравномерный характер из-за теплоемкости материалов и свободной конвекции. Предложено использовать систему принудительного охлаждения шунта, которая позволит уменьшить внутренние механические напряжения резистивного элемента и улучшить метрологические характеристики шунта в целом.

Ключевые слова:

Коаксиальный шунт, распределение температурного поля, температура перегрева резистивного элемента, метод конечных элементов, ANSYS.

Key words:

Coaxial shunt, temperature field distribution, superheat temperature of resistance element, finite-element method, ANSYS.

При испытаниях электроэнергетического оборудования, такого как мощные трансформаторы, силовые размыкатели, коммутаторы, защитные устройства, а также при контактной сварке и нанесении покрытий методом микродугового оксидирования применяют резистивные преобразователи тока. Предпочтения отдают коаксиальной конструкции шунтов, обладающих наиболее стабильными метрологическими характеристиками [1–3]. Коаксиальный шунт состоит из двух вложенных друг в друга цилиндров, по которым протекает ток в противоположных направлениях. Внутренний цилиндр является резистивным элементом, выполненным из материала с малым удельным сопротивлением, например манганин, а внешний — из материала с большим удельным сопротивлением, например медь.

Расчет параметров коаксиальных шунтов состоит в определении оптимальных размеров и рациональном выборе материала резистивного элемента, которые должны обеспечивать заданные технические и метрологические характеристики шунта при минимальных габаритах.

Основным фактором, определяющим размер резистивного элемента, является пропускаемый им ток и, следовательно, мощность рассеиваемого им тепла. Поэтому при выборе его габаритных размеров необходимо учитывать температуру перегрева [4]. Под температурой перегрева понимается разность между температурой поверхности резистивного элемента и температурой окружающей среды.

Температурное поле коаксиального шунта описывается сложными дифференциальными уравнениями, получение аналитического решения которых вызывает затруднения, особенно при рассмотрении трехмерного случая. Целью данной статьи является исследование температурного поля коаксиального шунта с помощью разработки модели в программном пакете ANSYS.

В настоящее время существует множество программных продуктов, позволяющих провести сложные модельные эксперименты по расчету распределения температурного поля резистивных шунтов и уменьшить трудоемкость натурных экс-

периментов. Для решения поставленной задачи была выбрана система ANSYS, реализующая метод конечных элементов (МКЭ), применение которого позволяет с высокой точностью рассчитать тепловое поле коаксиального шунта, имеющего относительно сложную геометрическую конструкцию. Основными этапами расчета в программном комплексе ANSYS являются:

- 1) построение геометрии исследуемого объекта;
- 2) ввод свойств материалов;
- 3) формирование граничных условий;
- 4) генерация сетки конечных элементов;
- 5) проведение расчета;
- 6) визуализация и анализ результатов.

Реализация этапа 1 осуществляется в рамках модуля DesignModeler или CAD-программах, этапа 2 – на платформе Ansys Workbench в рамках модуля Engineering Data, этапов 3–6 – в рамках модуля Multiple Systems [5].

Исследование температурного поля коаксиального шунта – нетривиальная задача из-за наличия внутренних источников теплоты, вызванных прохождением тока в шунте. При исследовании переноса теплоты в таких случаях важно знать интенсивность объемного выделения теплоты, так называемое джоулево тепло [3].

В программном комплексе ANSYS определение джоулева тепла проводится посредством модуля расчета электрических параметров Electric, который также позволяет учитывать теплоемкость материалов коаксиального шунта. Для получения распределения температурного поля шунта, кроме модуля Electric, необходимо использовать модуль расчета параметров переходных тепловых процессов Transient Thermal [6]. Рис. 1 иллюстрирует взаимосвязь используемых расчетно-графических модулей Workbench.

Рассмотрим подробно методику расчета температурного поля коаксиального шунта.

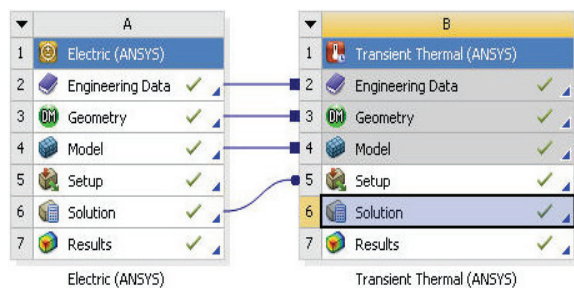


Рис. 1. Взаимосвязь используемых при моделировании расчетно-графических модулей Electric и Transient Thermal

Этап 1. Построение геометрии исследуемого объекта

Разработка конструкции коаксиального шунта возможна с помощью модуля DesignModeler или импорта готовой геометрии из CAD-программ. В рассматриваемом случае наиболее эффективным оказалось создание геометрии коаксиального шунта в CAD-программе, а именно в графическом пакете Inventor 2011. Конструкция исследуемого коаксиального шунта представлена на рис. 2.

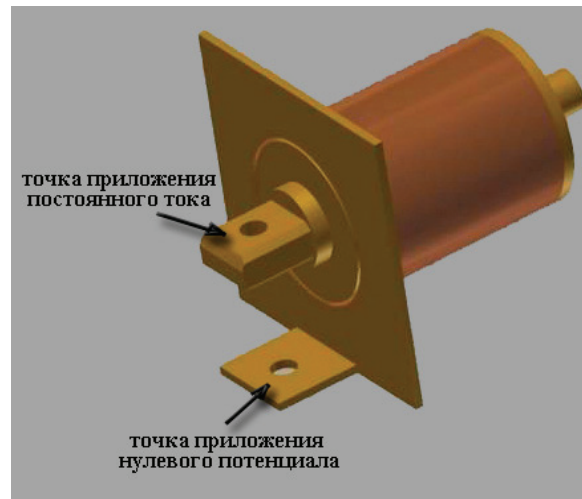


Рис. 2. Конструкция исследуемого коаксиального шунта

Этап 2. Ввод свойств материалов

Ввод свойств материала проводится с помощью модуля Engineering Data. Данный модуль содержит базу данных свойств различных материалов, в том числе теплоемкость, удельное сопротивление и т. д. Эта база может дополняться и изменяться пользователем. В процессе моделирования в базу были добавлены свойства манганина посредством следующей последовательности операций: Engineering Data→Edit→A2→ввод названия материала→выбор необходимых свойств материала→ввод значений свойств материала→+→Refresh Project→Return to Project. Необходимо отметить, что перечень доступных свойств материалов зависит от выбранного типа используемых расчетно-графических модулей.

Этап 3. Формирование граничных условий

Основным этапом в процессе получения решения любого модуля является правильное формирование граничных условий. От значений граничных условий зависит размерность расчетной области (число расчетных точек и конечных элементов).

Граничными условиями для модуля Electric являются значение постоянного тока 2 кА, приложенного к токовому выводу, и нулевое значение потенциала на выводе заземления, как представлено на рис. 2.

Первым граничным условием для модуля Transient Thermal является решение, полученное в модуле Electric, а именно распределение джоулева тепла, которое импортируется в этот модуль следующей последовательностью действий: Electric→Solution→Transfer Data to New→Transient Thermal.

Вторым граничным условием при расчете температурного поля является часть поверхности шунта, подвергаемая естественному охлаждению, которое осуществляется за счет свободного перемещения воздуха в окружающей среде [6]. При естественном охлаждении тепло от резистивного элемента коаксиального шунта передается в окружающую среду в основном конвекцией и частично за счет тепло-

проводности внешней медной трубки, токоподводов, крепежных деталей. Таким образом, в качестве второго граничного условия выбрана вся внешняя поверхность моделируемого шунта (рис. 3).

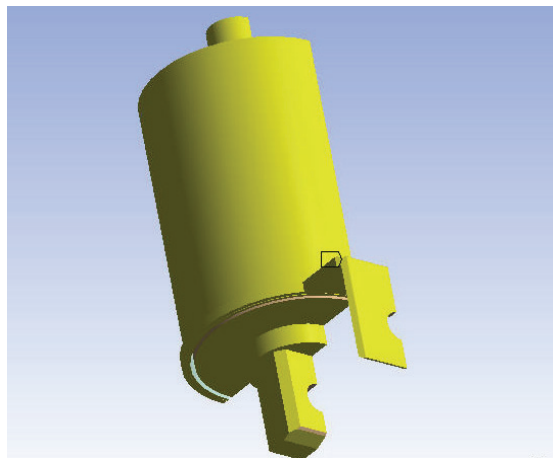


Рис. 3. Внешняя поверхность шунта при естественной конвекции

Этап 4. Генерация сетки конечных элементов

При разбиении расчетной области на конечные элементы в процессе исследования температурного поля коаксиального шунта использовалась стандартная сетка, для которой был задан размер конечных элементов 0,01. При этом число расчетных узлов (Nodes) составило 111751, число элементов (Elements) – 19163, что позволило достичь приемлемого качества решения. Сгенерированная сетка модели коаксиального шунта представлена на рис. 4.

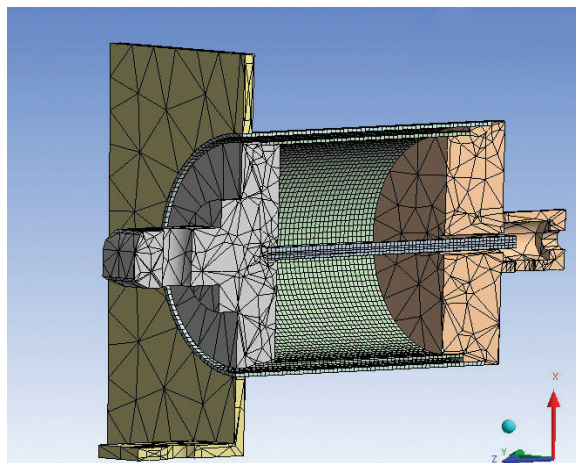


Рис. 4. Сгенерированная сетка коаксиального шунта

Этап 5. Проведение расчета

Из множества параметров, предлагаемых программным комплексом ANSYS для определения температурного поля коаксиального шунта, были выбраны следующие:

- Джоулево тепло (в модуле Electric Joule Heat);
- Температура (в модуле Transient Thermal Temperature).

Определение джоулева тепла и распределения температурного поля осуществлялось для всей поверхности коаксиального шунта.

Этап 6. Визуализация и анализ результатов

Визуализация результата расчета джоулева тепла для всей конструкции коаксиального шунта в модуле Electric представлена на рис. 5. Как отмечалось выше, полученное решение является граничным условием для модуля Transient Thermal. Как видно из рис. 5, при подаче постоянного тока 2 кА в центральной (светлой) области А джоулево тепло распределяется равномерно по всему резистивному элементу вследствие предположения, что ток протекает по телу резистивного элемента равномерно с постоянной плотностью.

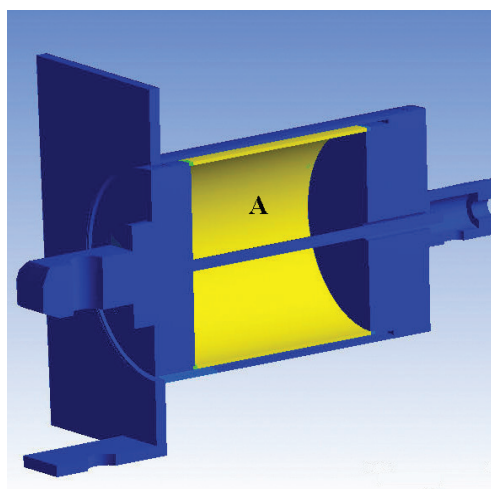


Рис. 5. Джоулево тепло коаксиального шунта

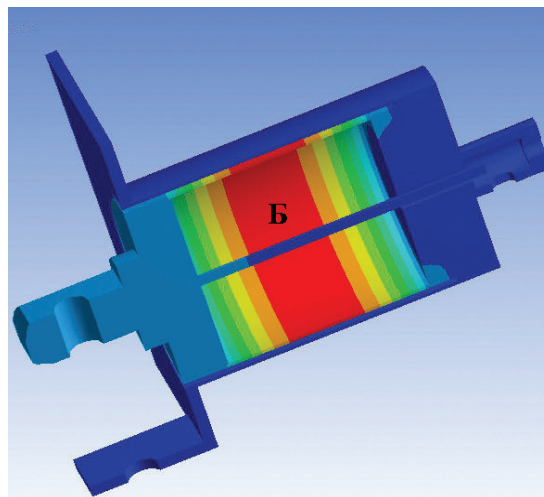


Рис. 6. Распределение температурного поля коаксиального шунта

Равномерно распределенное джоулево тепло передается в модуль Transient Thermal, в котором, как отмечалось выше, проводится определение температурного поля коаксиального шунта. Полученное трехмерное распределение температурного

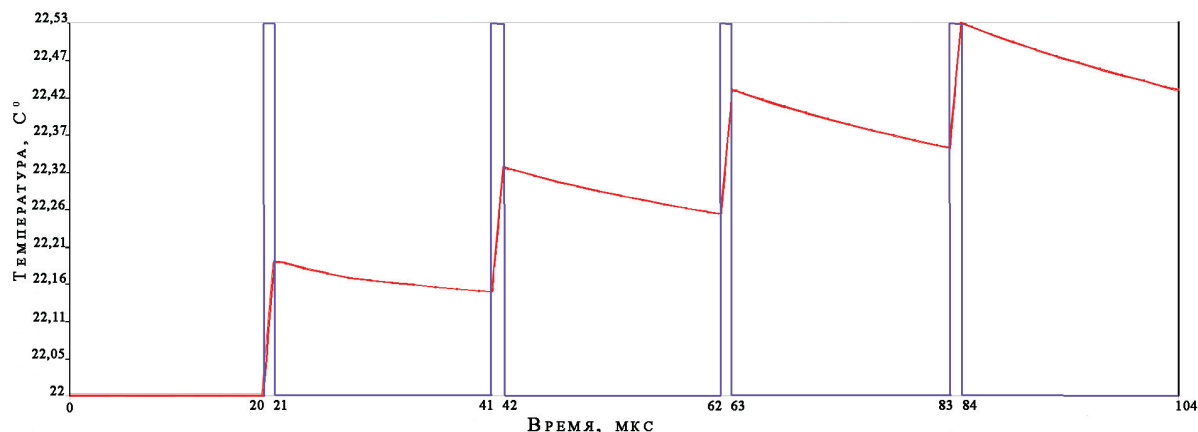


Рис. 7. Диаграмма изменения температуры при импульсном токе

поля коаксиального шунта представлено на рис. 6. Температура имеет максимальное значение в центре резистивной трубки в области Б на рис. 6. Минимальные температуры в основном сосредоточены на внешней медной трубке и торцах коаксиального шунта. Такое распределение температурного поля обусловлено естественной конвекцией и свойствами используемых материалов в коаксиальном шунте.

По описанной выше методике расчета были проведены исследования температурного поля коаксиального шунта при воздействии **импульсного тока длительностью 1 мкс** с периодом следования 21 мкс. В итоге получена зависимость изменения температуры от времени, представленная на рис. 7.

Если ток через шунт является **периодическим импульсным**, благоприятный режим для шунта, с точки зрения охлаждения, обеспечивается при периоде следования импульсов, равном времени охлаждения шунта до его первоначальной температуры. Если же шунт не успевает охладиться до первоначальной температуры, как в нашем случае, то для каждого следующего импульса начальной температурой является температура, до которой шунт охладится после окончания действия предыдущего импульса. Такой кумулятивный процесс происходит до тех пор, пока не наступит квазистационарное состояние температуры, когда повышение температуры за время действия импульса становится

равным температуре, полученной за период следования [7]. При периодическом импульсном сигнале с указанными выше параметрами квазистационарное состояние температуры не было получено из-за ограниченных возможностей используемого персонального компьютера.

Заключение

Результаты проведенного моделирования показывают, что при подаче постоянного или переменного электрического тока равномерное выделение теплоты приводит к неравномерному нагреву резистивного элемента коаксиального шунта. Максимальная температура нагрева имеет место в центре резистивной трубки и уменьшается к краям, а особенно к торцам. Это обусловлено наличием естественной конвекции и теплоемкости используемых материалов. Неравномерный нагрев резистивного элемента шунта нежелателен, т. к. неравномерность температурного поля вызывает внутренние механические напряжения в резистивном элементе при изменении значения тока в шунте, приводя к нелинейному характеру зависимости комплексного коэффициента преобразования шунта от температуры.

Таким образом, для снижения температуры перегрева резистивного элемента при постоянном или переменном токах следует использовать системы принудительного охлаждения.

СПИСОК ЛИТЕРАТУРЫ

1. Muravyov S.V., Borikov V.N., Natalina N.M. A Computer System: Measurement of Welding Surge Current // Measurement and Control. – 2009. – V. 42 – № 2. – P. 44–47.
2. Бороков В.Н., Баранов П.Ф., Мамаев А.И. Программно-аппаратный комплекс для исследования микроплазменных процессов в растворах // Приборы. – 2011. – № 12. – С. 53–58.
3. Kawamura T., Haginomi E., Goda Y., Nakamoto T. Recent Developments on High Current Measurement Using Current Shunt // Transactions on electrical and electronic engineering. – 2007. – V. 2. – № 5. – P. 516–522.
4. Векслер М.С., Теплинский А.М. Шунты переменного тока. – Л.: Энергоатомиздат, 1987. – 120 с.
5. Буль К.А. ANSYS в примерах и задачах. – М.: Компьютер-Пресс, 2002. – 223 с.
6. Залесский Л.М., Кукеков Г.А. Тепловые расчеты электрических аппаратов. – Л.: Энергия, 1967. – 379 с.
7. Бедарева Е.В. Исследование распределения температурного поля коаксиального шунта методом конечных элементов // Молодежь и современные информационные технологии: Труды IX Междунар. научно-практ. конф. молодых ученых. – Томск, 2011. – С. 280–281.

Поступила 13.02.2013 г.

УДК 533.9.08

ПРИБОРНЫЙ КОМПЛЕКС ДЛЯ СБОРА И РЕГИСТРАЦИИ РЕЗУЛЬТАТОВ ИЗМЕРЕНИЙ МАГНИТНЫХ ПАРАМЕТРОВ ЭЛЕКТРОФИЗИЧЕСКОЙ УСТАНОВКИ

С.В. Меркулов, Ю.Н. Голобоков, К.И. Байструков, В.М. Павлов,
А.Г. Качкин, А.А. Мезенцев, А.М. Ли, В.С. Хохряков

Томский политехнический университет
E-mail: stepan@tpu.ru

Предложена структура комплекса технических средств для сбора и регистрации результатов измерения магнитных параметров установки КТМ. Разработаны алгоритмы и протоколы передачи результатов измерений от измерительных преобразователей в блоки регистрации данных. На базе модулей с программируемой логикой реализованы устройства приёма данных от измерительных преобразователей. Разработаны алгоритмы сбора, регистрации и первичной обработки результатов измерений электромагнитных параметров.

Ключевые слова:

Токамак, магнитная диагностика, комплекс сбора и регистрации данных, структура измерительного канала, протокол последовательной передачи данных, ввод-вывод цифровых сигналов, алгоритм сбора данных.

Key words:

Tokamak, magnetic diagnostics, data acquisition and logging system, measuring channel structure, serial data transfer protocol, digital I/O, data acquisition algorithm.

Введение

Измерение магнитных параметров в электрофизических установках исследовательского и промышленного назначения является весьма важной задачей для реализации функций управления и контроля процессом в реальном масштабе времени. Кроме этого, результаты измерений, накопленных за разряд, необходимо сохранять для последующей обработки и анализа. В течение каждого пуска установки (эксперимента) с измерительных преобразователей электромагнитной диагностики снимается значительный объем информации, составляющий несколько десятков мегабайт на канал. Использование в каждом измерительном канале для регистрации этой информации внутренней памяти приводит к усложнению и удорожанию системы сбора и регистрации данных. В данной статье рассматривается альтернативный подход, при котором весь поток измерительной информации, сформированный аналого-цифровыми преобразователями, установленными в схемах нормирования сигналов, передается по высокоскоростным каналам последовательной передачи на верхний уровень системы, где регистрируется на магнитных носителях и используется, в том числе для управления параметрами процесса в реальном масштабе времени. Рассматриваемое в работе решение реализовано в подсистеме сбора данных с датчиков электромагнитной диагностики токамака КТМ [1]. В дальнейшем изложении для обозначения набора аппаратных и программных средств, разработанных в составе подсистемы сбора данных, будет использоваться термин «приборный комплекс».

Структура приборного комплекса

Структура приборного комплекса показана на рис. 1. Измерение сигналов с датчиков электромагнитной диагностики производится модулями

первичной обработки сигналов (МПОС), входящими в состав блоков БПОС1 и БПОС2. Датчики электромагнитной диагностики представляют собой индукционные катушки (локальные зонды), разомкнутые витки (распределенные петли), а также пояса Роговского. Выходные сигналы с датчиков, с целью уменьшения влияния помех, передаются в систему измерения по витым парам, помещенным в металлическую оплетку. Тем не менее, желательно сократить общую длину используемых в системе проводных линий. Для этого техническая реализация БПОС выполнена в виде экранированных блоков, что позволяет разместить их на расстоянии 1 метра от установки и передавать сигналы в цифровом виде по волоконно-оптическим линиям связи, на которые переменное магнитное поле не оказывает никакого влияния.

Все МПОС выполняют измерение одновременно. Каждый модуль имеет в своем составе четыре параллельно работающих схемы нормировки и аналого-цифрового преобразования сигналов, что позволяет вести одномоментную оцифровку сигналов с четырех датчиков. Синхронность работы всех схем внутри блока БПОС обеспечивает модуль локальной синхронизации ЛМС-3v, также входящий в состав блока БПОС. После каждого временного отсчета модули МПОС формируют из результатов измерения информационные пакеты и передают их последовательно бит за битом в блоки регистрации данных БРД1 и БРД2 по волоконно-оптическим линиям связи.

Оптические сигналы, представляющие собой последовательности битов информационного блока, перед вводом в БРД преобразуются в электрическую форму. Эту функцию выполняют блоки опто-электрического преобразования (ОЭП). Дополнительно ОЭП выделяет из потока данных сигнал несущей частоты, который используется для

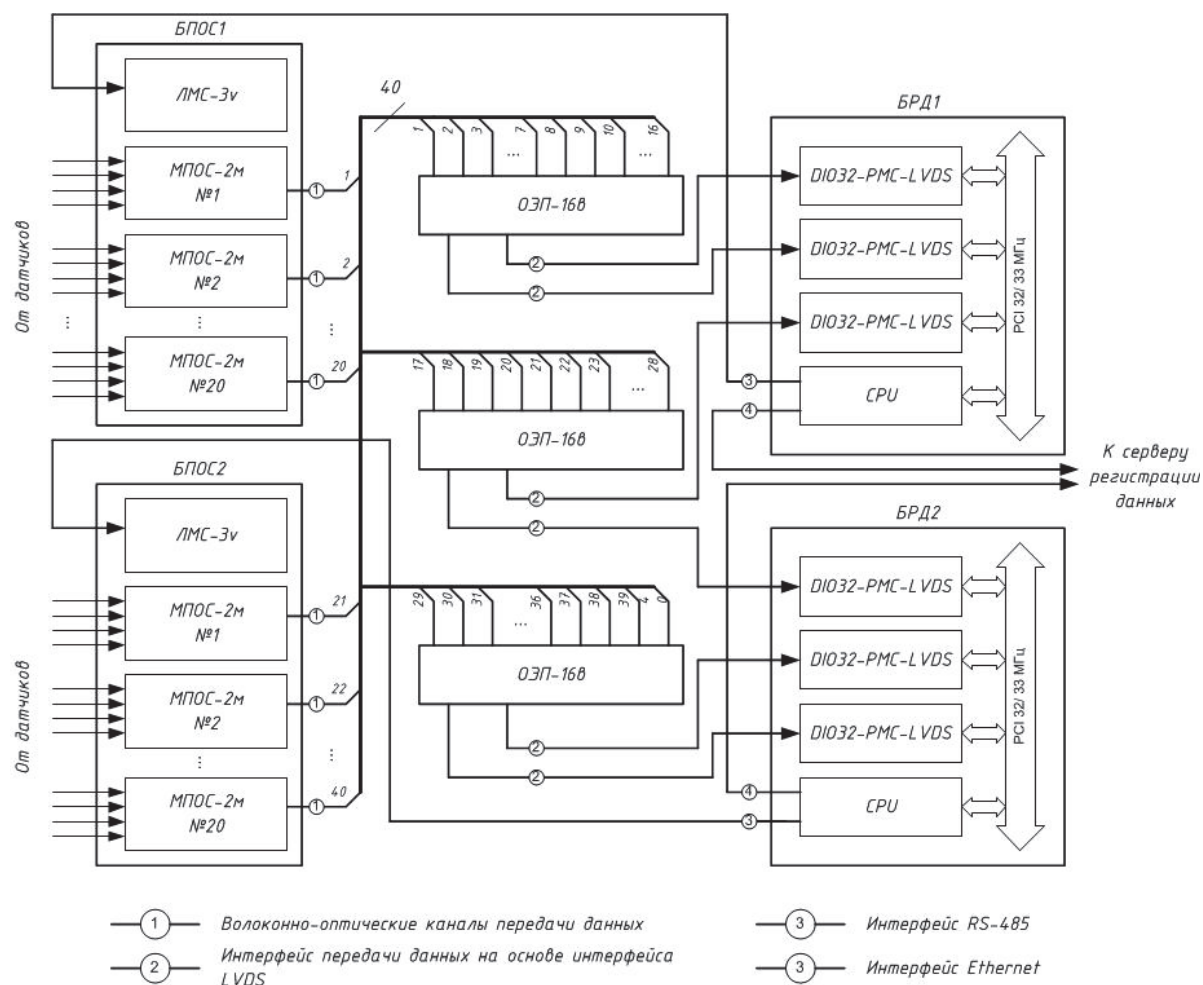


Рис. 1. Структура приборного комплекса

тактирования приемников цифровых данных в блоке БРД. Это значительно упрощает схему приемника и повышает надежность работы измерительных каналов приборного комплекса. Также в целях повышения помехоустойчивости и надежности комплекса в целом сигналы на выходе ОЭП представлены в формате LVDS.

Прием и первичную регистрацию информационных пакетов в буферной памяти осуществляют модули ввода цифровых сигналов DIO32-PMC-LVDS, входящие в состав блоков регистрации данных. Каждый такой модуль может принимать информацию от восьми МПОС при максимальной частоте оцифровки сигналов датчиков до 350 тысяч отсчетов в секунду. При приеме DIO32-PMC-LVDS выполняет проверку пакетов на наличие искажений, извлекает из них результаты измерений и записывает эти данные в свою внутреннюю буферную память. Из буферной памяти через системную шину PCI информация считывается процессорным модулем (CPU), который подвергает её различным алгоритмам обработки и сохраняет до окончания разряда на магнитном диске блока регистрации данных. Накопленные за разряд данные по окончании эксперимента считываются

из БРД сервером регистрации и хранения данных электрофизической установки по сети Ethernet.

Кроме хранения экспериментальных данных сервер регистрации и хранения данных выполняет функции конфигурирования приборного комплекса. Конфигурационные данные, хранящиеся на сервере, рассылаются в БРД по сети Ethernet. В свою очередь каждый из блоков регистрации данных, используя интерфейс RS-485, выполняет загрузку в МПОС и ЛМС параметров настройки, таких как коэффициенты усиления и деления входных сигналов, частоты среза входного фильтра, частоты тактирования схем нормировки и АЦП и другие.

Структура и формат информационных пакетов

Как упоминалось выше, результат каждого измерения модули первичной обработки сигналов передают в БРД в виде информационных пакетов. Структура пакета представлена на рис. 2.

Информационный пакет начинается с заголовка START, позволяющего обнаружить пакет в потоке данных. Далее следуют поля DATA1–DATA4, содержащие результаты очередных измерений каждого из четырех измерительных каналов МПОС.

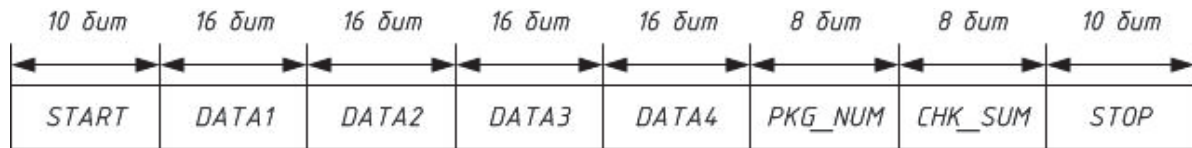


Рис. 2. Структура информационного пакета

Эти результаты представлены в виде 16-ти разрядного двоичного кода.

Поле PKG_NUM содержит номер пакета, лежащий в диапазоне от 0_{10} до 255_{10} , значение которого изменяется на единицу у очередного пакета относительно предыдущего. При достижении максимального значения номер пакета устанавливается в ноль, и изменение продолжается в том же порядке. Использование поля PKG_NUM позволяет вести контроль «выпавших пакетов», т. е. пакетов, при передаче и приеме которых произошел сбой. Данный контроль необходим, поскольку в целях повышения пропускной способности приборного комплекса результаты измерений передаются на верхний уровень в симплексном режиме без буферизации и без возможности повторной передачи пакета. Таким образом, зная номер текущего и предыдущего пакетов, программное обеспечение на приемной стороне может определить не принятый или пропущенный пакет, выявляя таким образом в результатах измерений «промахи» и обеспечивая возможность дальнейшей корректной обработки данных.

Поле CHK_SUM содержит контрольную сумму, необходимую для выявления искажений в данных, которые возникли при передаче и (или) приеме пакета. Значение CHK_SUM рассчитывается как сумма по модулю 2 содержимого полей DATA1, DATA2, DATA3, DATA4 и PKG_NUM.

Завершает пакет поле STOP. По значению этого поля на приемной стороне определяется конец пакета.

Необходимо отметить также особенности работы блока ОЭП в процессе передачи и приема информационных пакетов. Как упоминалось ранее, ОЭП на основе потока данных формирует сигналы синхронизации, стробирующие каждый бит информационного пакета. Но с целью повышения надежности работы ОЭП длинные последователь-

ности нулей и единиц из последовательного потока данных необходимо исключить. Для этого к данным, передаваемым модулями первичной обработки сигналов, применен метод кодирования 4B5B, используемый также в таких сетевых стандартах, как FDDI и Fast Ethernet. Кодированию по методу 4B5B подлежат все поля информационного пакета за исключением полей START и STOP, которые изначально представляют собой фиксированные битовые комбинации, построенные при условии обязательного чередования нулей и единиц.

Блок регистрации данных

Каждый из двух блоков регистрации данных выполняет прием, первичную регистрацию и обработку результатов измерений и представляет собой промышленный компьютер, работающий под управлением операционной системы LINUX. В составе БРД находится ЭВМ и подключенные к ней по магистрали PCI модули DIO32-PMC-LVDS.

Интерес представляют модули цифрового ввода сигналов, реализующие функции приемника информационных пакетов. БРД содержит в своем составе по три таких модуля. Каждый из модулей одновременно способен принимать данные от 8-ми источников цифрового сигнала. Основу модуля составляет микросхема программируемой логики — ПЛИС. На рис. 3 показана структурная схема реализованного на базе ПЛИС приемника информационных пакетов, обеспечивающая, сбор и первичную регистрацию данных в буферной памяти от 8-ми МПОС.

В приведенной схеме отдельно выделяется восемь приемных каналов, работу которых рассмотрим более подробно. От блока ОЭП на вход CNL.DATA1 канала поступает поток последовательных данных совместно с сигналами синхронизации. По переднему фронту синхросигнала CLOCK1 биты потока данных задвигаются в 120-ти

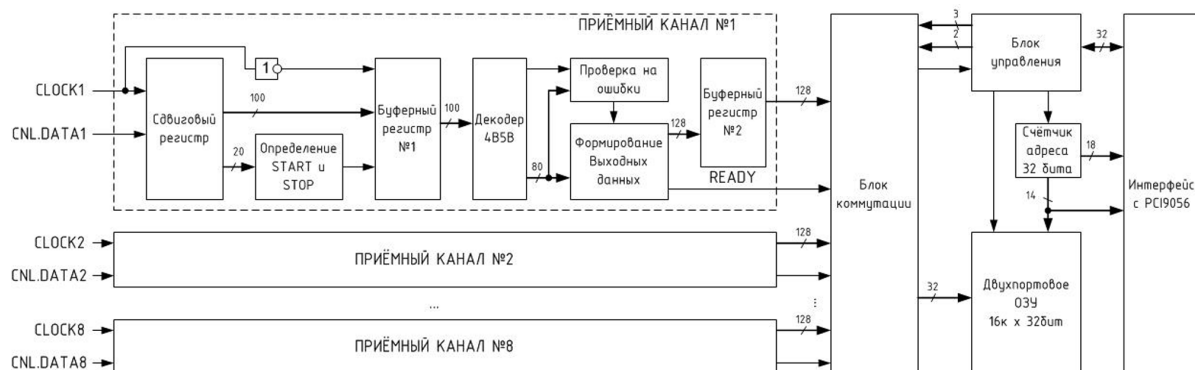


Рис. 3. Структурная схема приёмника информационных пакетов

разрядный «Сдвиговой регистр». Одновременно с этим на каждом такте CLOCK1 значения первых и последних 10-ти разрядов регистра проверяются на их соответствие фиксированным значениям полей START и STOP информационного пакета. В случае совпадения с выхода «Сдвигового регистра» содержимое оставшихся ста бит информационного пакета заносится в «Буферный регистр № 1».

Из «Буферного регистра № 1» пакет передается в декодер, где принимает свой исходный вид. Затем его содержимое пересылается в блок проверки ошибок, где путем расчета контрольной суммы проверяется на наличие искажений. При этом кроме контрольной суммы учитывается присутствие или отсутствие выявленных при декодировании «запрещенных» кодом 4B5B комбинаций. «Сбойный» информационный пакет далее не обрабатывается и исключается из потока. При отсутствии ошибок блок формирования выходных данных устанавливает сигнал готовности READY и на основе содержимого пакета составляет четыре 32-х разрядных выходных слова, которые сохраняются в буферном регистре № 2. Формат этих слов представлен на рис. 4.

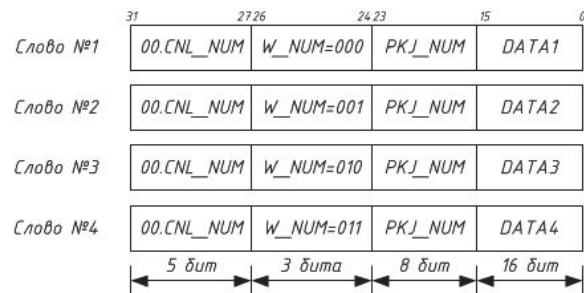


Рис. 4. Структура выходных слов

Поле CNL_NUM содержит информацию о номере приёмного канала модуля DIO32-PMC-LVDS (лежит в диапазоне от 1 до 8). Трёхразрядное поле W_NUM указывает порядковый номер выходного слова (от 000₂ до 011₂). Это значение эквивалентно номеру измерительного канала в МПОС, что необходимо для идентификации данных после передачи их в ЭВМ. Поле PKJ_NUM содержит номер полученного информационного пакета, а DATA1–DATA4 соответственно передаваемые в пакете данные.

Выходные данные приёмных каналов поступают на вход блока коммутации, где при помощи трех битового адреса сканируются сигналы READY приёмных каналов. При обнаружении установленного сигнала готовности канала блок управления, используя двухразрядный адрес, последовательно записывает выходные слова этого канала в двухпортовое ОЗУ.

Модули DIO32-PMC-LVDS связываются с ЭВМ посредством шины PCI. Для реализации этой возможности в модулях DIO32-PMC-LVDS используется специализированная микросхема PC19056.

Алгоритм сбора и обработки данных

Архитектура микропрограммы приемника данных DIO32-PMC-LVDS предполагает помещение получаемых информационных пакетов во входной буфер приемника в порядке их прихода. При такой схеме возможен приход пакетов одного временного отсчета со всех МПОС в произвольном порядке, следовательно, необходимо после ввода информационных пакетов из буфера упорядочить их в соответствии с номером входа DIO32-PMC-LVDS, номером слова данных и номером временного отсчета, пересылаемым вместе с результатами измерений. Эта информация позволяет установить исходную последовательность принятых данных, а также ввести механизм защиты в случае потери пакета при передаче либо в случае переполнения входного буфера DIO32-PMC-LVDS. Такая организация данных может быть особенно полезна для систем, которые используются в контурах управления в реальном масштабе времени. Для восстановления последовательности данных применяется алгоритм, показанный на рис. 5.

Передача информационных пакетов из буфера приемника DIO32-PMC-LVDS осуществляется при помощи механизма DMA Master. Ввод данных из буфера модуля в память ЭВМ происходит при наличии достаточного количества информационных пакетов в буфере модуля. При определении необходимого минимального количества данных для ввода необходимо учитывать, что механизм DMA работает наиболее эффективно при использовании блоков информации как можно большего размера, так как при передаче каждого блока возникают накладные расходы на программирование контроллера DMA. С другой стороны, буфер приемника DIO32-PMC-LVDS имеет ограниченный размер, в каждом БРД используется 3 модуля DIO32-PMC-LVDS и могут возникать задержки в работе программы CPU, связанные с передачей управления другой задаче планировщиком операционной системы. В серии экспериментов, был определен минимальный размер блока, при котором эффективно выполняется транзакция DMA, который составляет 256×32 бит. Для определения количества данных, находящихся в буфере, используется счетчик адреса.

При наличии достаточного количества информационных пакетов в буфере приемника фиксируется метка времени начала сбора данных. Метки времени в дальнейшем используются для определения правильности частоты аналого-цифрового преобразования входного сигнала, а также для восстановления последовательности данных. После получения метки времени выполняется настройка DMA контроллера модуля DIO32-PMC-LVDS на передачу собранных информационных пакетов в режиме DMA Master. Об окончании процесса передачи модуль извещает при помощи прерывания. В соответствии с принципом работы механизма DMA для ввода данных требуется память с прямой

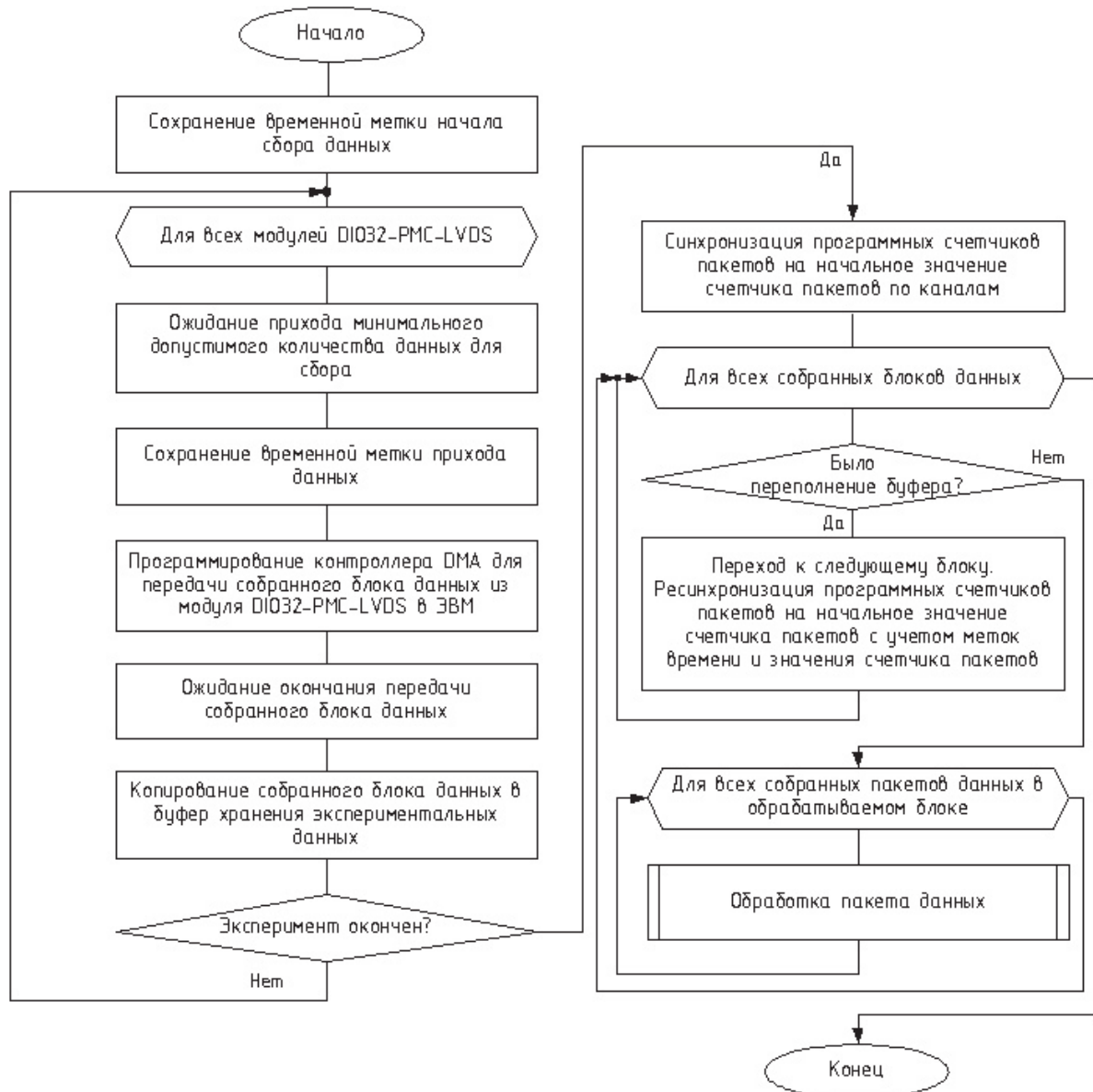


Рис. 5. Блок-схема алгоритма сбора данных

физической адресацией, размер которой, как правило, ограничен. Поэтому пакеты, введенные из модуля, в дальнейшем копируются из области памяти с прямой физической адресацией в буфер хранения, являющийся обычной областью памяти с виртуальной адресацией. Указанный алгоритм выполняется до окончания эксперимента для каждого из модулей DIO32-PMC-LVDS последовательно и многократно.

После окончания эксперимента выполняется обработка и сортировка собранных экспериментальных данных. Для этого выполняется последовательная обработка информационных пакетов данных по алгоритму, приведенному на рис. 6.

Для осуществления сортировки данных в памяти выделяется массив для каждого из каналов сбора данных. Каждый канал сбора данных ставится в соответствие измерительному каналу модуля МПОС-2м.

Первичная обработка блока данных заключается в подсчете количества отсчетов для каждого канала в собранном блоке и проверке соответствия этого количества расчетному теоретическому количеству. Расчетное количество вычисляется исходя из заданной частоты оцифровки f и разницы между метками времени сбора предыдущего t_{n-1} и текущего t_n блока данных по следующей формуле:

$$N_{\text{отсчетов}} = (t_n - t_{n-1})f.$$

В случае если количество собранных данных сходится с расчетным либо незначительно от него отличается, выполняется следующий этап обработки. В случае если количество отличается значительно, то каналу выставляется признак недостоверности. Это означает, что при приходе значения по данному каналу необходима ресинхронизация последовательности данных. В случае если при сборе обрабатываемого блока данных произошло

переполнение буфера, то значения данных по всем каналам в данном блоке считаются недостоверными и также выполняется ресинхронизация.

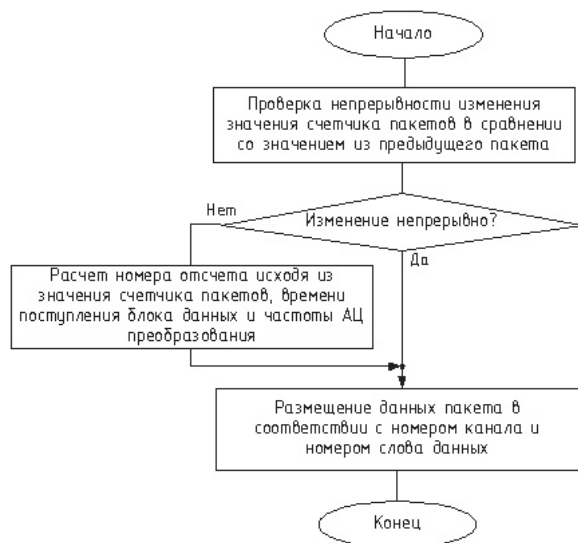


Рис. 6. Блок-схема обработки пакета данных

Далее выполняется второй проход по блоку данных, в котором каждое слово данных выбирается из памяти, из него извлекается номер слова данных, номер входа DIO32-PMC-LVDS и значение счетчика пакетов. Для указанного в пакете номера канала и слова данных определяется номер канала сбора данных по формуле:

$$n_{\text{канала сбора}} = ((n_{\text{модуля DIO}} - 1) \cdot 8 + (n_{\text{входа модуля DIO}} - 1)) \cdot 4 + n_{\text{слова}}$$

После этого выполняется сравнение значения номера обрабатываемого пакета для вычисленного номера канала сбора данных и предыдущего значения счетчика для этого же канала. Номер счетчика должен монотонно инкрементироваться, а при переполнении должен начинать счетную последовательность с нуля. Если данное условие не выполняется, то инициируется процесс ресинхронизации. Процесс ресинхронизации заключается в расчете места во временной последовательности для пакета информации, пришедшего после какой-либо ошибки. Для этого рассчитывается разность между текущим и предыдущим значениями счетчика пакетов. Исходя из закона изменения счетчика, в этом случае мы можем считать, что в последовательности было пропущено M отсчетов данных, при этом M вычисляется по формуле:

$$M = (A_n - A_{n-1}) + 256K,$$

где A_n — текущее значение счетчика пакетов; A_{n-1} — предыдущее значение счетчика пакетов; K — количество пропущенных циклов счетчика, которое может принимать значение от 0 до $(N_{\text{общ}} - N_{\text{обраб}}) / 256$; где $N_{\text{общ}}$ — общее количество отсчетов, собранное за эксперимент по данному каналу, а $N_{\text{обраб}}$ — количество уже обработанных отсчетов по данному каналу.

Для того чтобы определить количество пропущенных циклов K , также используются метки времени, значение счетчика пакетов и частота оцифровки. Вычисление производится по формуле:

$$K = ((t_n - t_{n-1})f) / 256.$$

При этом если $A_n > A_{n-1}$, то значение K округляется в меньшую сторону, а если это условие не выполняется, то в большую. Зная количество пропущенных отсчетов M , мы можем точно определить номер пришедшего отсчета. Для всех пропущенных отсчетов в буфере обработанных данных выставляется признак недостоверности. По окончании работы алгоритма, обработанные данные помещаются в файл.

Заключение

Описанный в статье приборный комплекс сбора и регистрации данных успешно прошел испытание на электрофизической установке — токамаке КТМ, расположенной в г. Курчатова, Республика Казахстан, в составе Системы автоматизации экспериментов на токамаке, и используется в настоящее время для измерения и регистрации сигналов с датчиков электромагнитной диагностики. Используемые в работе структурные и аппаратно-технические решения позволяют обеспечить помехоустойчивость измерительных каналов комплекса в условиях сложной электромагнитной обстановки, которая имеет место в реакторном зале. Предлагаемый в работе алгоритм первичной обработки информации позволяет осуществлять точную привязку полученных данных ко времени оцифровки без необходимости передачи временных меток от источников измерительной информации, что дает возможность увеличения пропускной способности системы в целом. Комплекс обеспечивает одновременное 16-ти разрядное измерение и регистрацию сигналов от 160-ти датчиков с частотой дискретизации до 350 кГц.

Исследование выполнено при поддержке Министерства образования и науки Российской Федерации, соглашение 14.V37.21.0457 «Разработка высокопроизводительного модульного приборного комплекса для автоматизированных систем экспериментальных исследований и управления электрофизическими установками ядерной энергетики».

СПИСОК ЛИТЕРАТУРЫ

1. Обходский А.В., Байструков К.И., Павлов В.М., Меркулов С.В., Голобоков Ю.Н. Система измерения электромагнитных параметров для электрофизической установки токамак

КТМ // Приборы и техника эксперимента. — 2008. — № 6. — С. 23–28.

Поступила 20.09.2012 г.

УДК 681.5

РАЗРАБОТКА СИСТЕМЫ УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИМ ПРОЦЕССОМ ПЕРЕРАБОТКИ НАТРИЕВОГО ТЕПЛОНОСИТЕЛЯ

П.Ю. Кривцов, В.М. Павлов*

Национальный ядерный центр Республики Казахстан, г. Курчатов

*Томский политехнический университет

E-mail: Krivtsov@nnc.kz

Дан общий алгоритм управления установкой по переработке натрия, определены основные функции системы и контуры регулирования технологических параметров в зависимости от текущего режима работы установки. На примере системы подачи натрия показан процесс разработки системы управления. Дано описание состава используемого программного обеспечения, приведена структура технологических экранов интерфейса оператора.

Ключевые слова:

Натриевый теплоноситель, установка по переработке натрия, система управления.

Key words:

Sodium coolant, sodium process facility, control system.

Введение

В настоящее время для решения комплекса задач, связанных с управлением сложными технологическими процессами, используются различные программные и технические средства, надежность и качество которых позволяют обеспечить требуемую безопасность объекта управления [1, 2]. Внедрение автоматизированных систем управления технологическими процессами (АСУ ТП) предусматривает расширенную функциональность системы управления, в котором помимо традиционных функций (системы измерения, дистанционного управления арматурой и исполнительными механизмами, технологическими защитами, автоматическим регулированием) решаются задачи автоматизированного логического функционально-группового управления технологическим объектом и информационно-вычислительные функции, оказывающие помощь обслуживающему персоналу в управлении технологическим процессом. Так, представленная в работе АСУ ТП установки по переработке натриевого теплоносителя (УПН) реактора БН-350, расположенного в г. Актау, позволяет обеспечить решение поставленных задач управления взрыво- и пожароопасным технологическим процессом.

Математическое описание химического процесса, технологии переработки натрия, методы реализации математической модели процесса и синтеза имитационного комплекса УПН представлены в работах [3, 4]. Завершающим этапом работ по созданию системы управления технологическим процессом на УПН является практическая реализация теоретических изысканий, представленных в указанных публикациях.

Объект автоматизации

Принцип работы установки по переработке натрия заключается в осуществлении реакции натрия с водой в концентрированном растворе гидроксида в химическом реакторе [5]. Для получе-

ния раствора заданной концентрации необходимо осуществлять подачу натрия и воды в строго определенных количествах.

При взаимодействии натрия с водой выделяется большой объем водорода и высвобождается значительное количество тепловой энергии. Избыток тепла отводится за счет испарения воды, таким образом, температура в химическом реакторе поддерживается на уровне температуры кипения раствора при заданной концентрации. Уравнение реакции записывается следующим образом:

	Na	+	H ₂ O	=	NaOH	+	1/2 H ₂	
ν	1		1		1		1/2	моль
M_r	22,99		18,02		40,00		2,02	г/моль
m	22,99		18,02		40,00		1,01	г
H	0		285,83		426		0	кДж

где ν — количество молей вещества; M_r — молярная масса; m — масса; H — энтальпия образования.

Под объектом автоматизации понимается технологическая часть УПН, к которой относятся элементы и конструкции основного и вспомогательного технологического оборудования установки, непосредственно или косвенно задействованного в реализации процесса переработки натрия. В соответствии с технологией переработки натрия, структура технологической части УПН как объекта автоматизации имеет вид, представленный на рис. 1.

Разделение УПН на технологические системы (подсистемы) по функциональному признаку носит относительный и условный характер и выполнено с целью упрощения описания функций автоматизации и контроля технологического процесса переработки натрия.

Основными технологическими параметрами, подлежащими измерению и управлению, являются:

- температура элементов и конструкций УПН;
- давление в емкостях и трубопроводах технологических систем УПН;

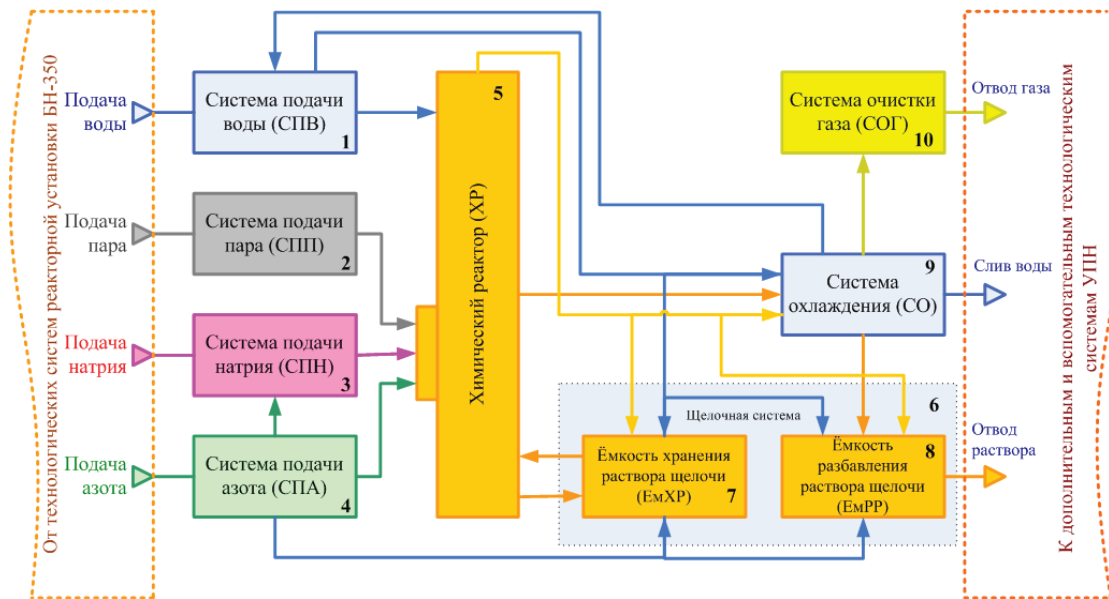


Рис. 1. Структура технологической части УПН

- расход натрия/щелочи/воды;
- уровень натрия/щелочи/воды в емкости;
- концентрация щелочи.

Основным назначением АСУ ТП УПН является реализация режимов работы объекта автоматизации в соответствии с выбранной технологией переработки. На этапе анализа объекта управления и исследования технологии переработки определен общий алгоритм управления технологическим процессом, представленный на рис. 2.

Разработка системы управления

Рассмотрим процесс разработки системы управления на примере системы подачи натрия (СПН), функционально-технологическая схема которой представлена на рис. 3. Для СПН можно выделить следующие контуры управления:

- потоком натрия (СРБ1, СРБ2, форсунка 1, форсунка 2);
- температурой конструкций СПН (СРБ1, СРБ2, трубопровод).

В различных технологических режимах работы УПН в составе АСУ ТП можно выделить два локальных контура управления потоком натрия. Первый обеспечивает управление заполнением суточных расходных баков (СРБ) натрием от БН-350, второй регулирует количество натрия, подаваемого в химический реактор (ХР).

Управление подачей натрия в СРБ осуществляется запорными электроприводными вентилями SVE-Na01 и SVE-Na02, контроль расхода выполняется по показаниям расходомера FE-Na01.

Из суточных расходных баков натрия подается к форсункам химического реактора. Для управления подачей натрия в форсунки используется ряд запорных вентилях и регулирующие вентили CV-Na01 и CV-Na02. Мгновенный расход натрия оценивается по показаниям расходомеров FE-Na02 и FE-Na03,

величина потока определяется проходным сечением регулирующего вентиля и давлением в СРБ.

На всех этапах работы установки системе управления необходимо обеспечить разогрев и поддержание рабочей температуры технологического оборудования, входящего в состав СПН. В СПН можно выделить два объекта нагрева: трубопровод и СРБ, при этом технологический регламент УПН предъявляет к системе управления нагревом следующие требования:

- разогрев трубопроводов и оборудования с натрием вести последовательным включением тепловых зон от свободного объема;
- разогрев основных трубопроводов, СРБ вести со скоростью $10\text{ }^{\circ}\text{C/ч}$ с выдержкой 4 ч через каждые $50\text{ }^{\circ}\text{C}$;
- допустимый перепад температур между соседними тепловыми зонами трубопроводов, заполненных натрием, должен быть не более $50\text{ }^{\circ}\text{C}$.
- разогрев СРБ вести от газовой «подушки» со скоростью $10\text{ }^{\circ}\text{C/ч}$ с выдержкой 4 ч через каждые $50\text{ }^{\circ}\text{C}$.

Конструктивно трубопроводы СПН представляют собой обогреваемые, теплоизолированные магистрали, выполненные из труб с диаметром условного прохода 25 мм и толщиной стенки 2,5 мм. Для обеспечения равномерного разогрева трубопровод СПН разделен на 20 тепловых зон, каждая из которых оборудована кабельными нагревателями, тиристорным регулятором мощности и датчиком температуры (термосопротивлением).

Суточный расходный бак представляет собой цилиндрическую герметичную емкость, оборудованную тридцатью восемью пластинчатыми нагревателями, расположенными на поверхности обечайки. Нагреватели СРБ сгруппированы в шести тепловых зонах с независимыми контурами управления.

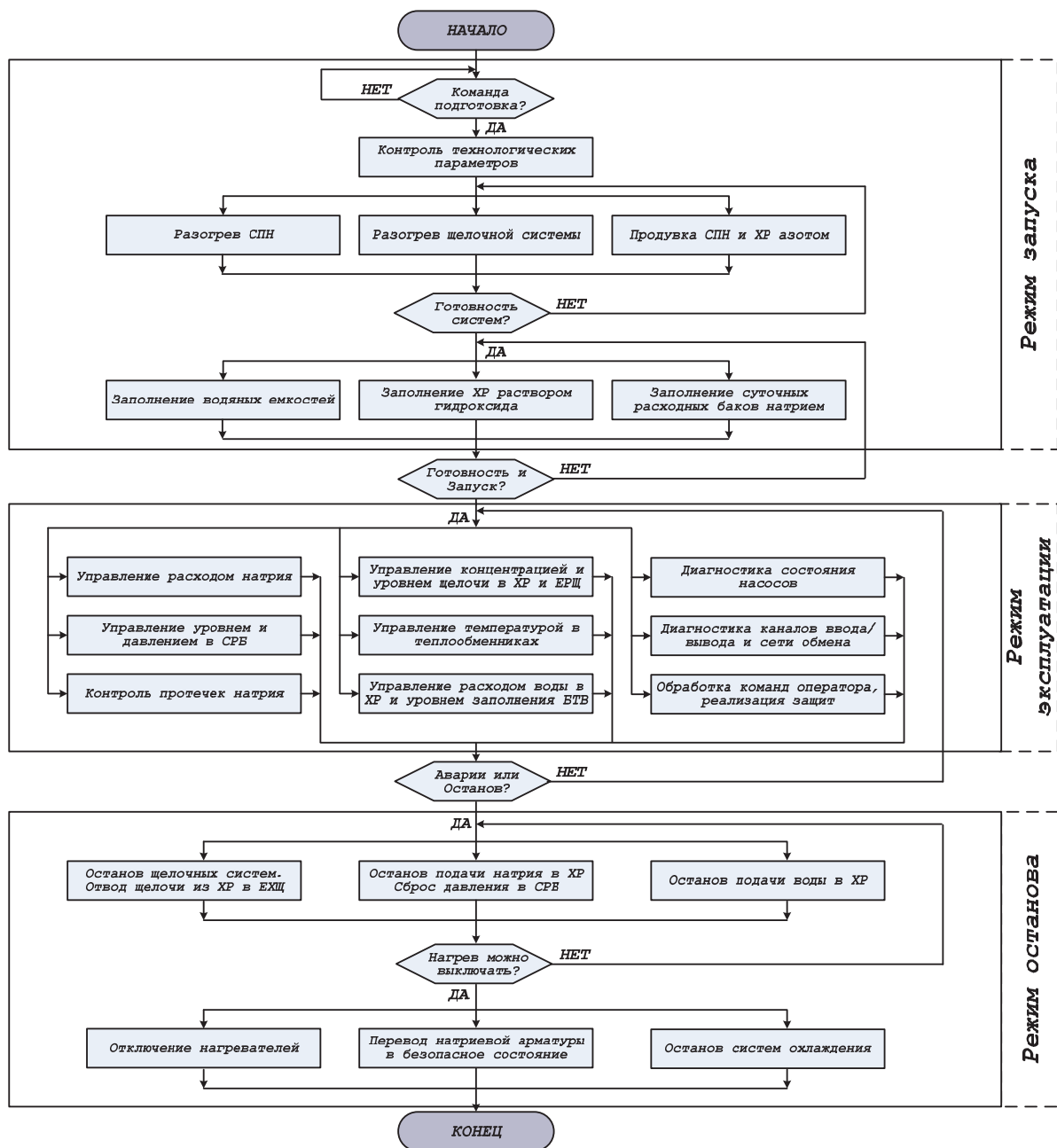


Рис. 2. Общий алгоритм управления технологическим процессом

Структурная схема контура регулирования температуры в отдельной тепловой зоне СПН приведена на рис. 4.

Она состоит из сумматора 1; регулятора температуры (РТ) 2; регулятора напряжения/мощности (РМ) 3, обеспечивающего согласование регулятора температуры и нагревателя; нагревательного элемента/нагревателя (НЭ) 4; датчика температуры (ДТ) 5 и объекта нагрева 6, под которым понимается любая часть или элемент технологического оборудования, где поддерживается требуемый температурный режим.

На основе анализа функционально-технологической схемы СПН, с учетом особенностей испол-

нительных элементов и средств измерений, определено, что информационная емкость системы управления должна обеспечивать обработку пятидесяти девяти аналоговых сигналов (датчики температуры, расхода, положения регулирующих вентилей) и тридцати трех дискретных сигналов (датчики уровня, протечек, положения запорных вентилей). При этом необходимо обеспечить формирование тридцати четырех аналоговых сигналов управления (регуляторы мощности и регулирующие вентили) и четырнадцать дискретных сигналов управления запорной арматурой.

Необходимо отметить, что на этапе выбора средств измерений и исполнительных элементов

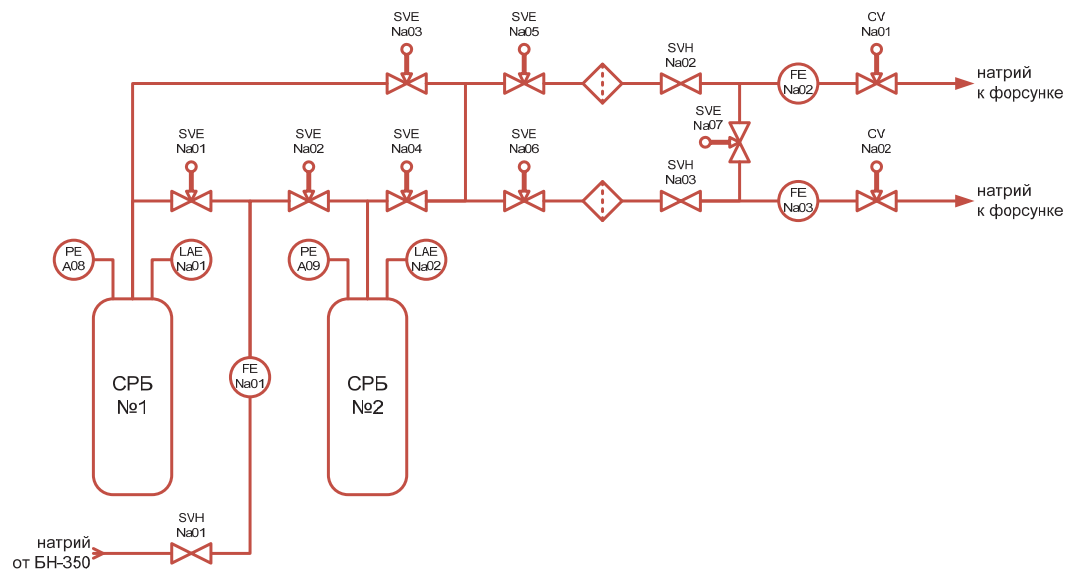


Рис. 3. Функциональная технологическая схема системы подачи натрия: CV – вентиль регулирующий, SVE – вентиль запорный с электроприводом, SVH – вентиль запорный ручной, FE – расходомер, PE – датчик давления, LE – дискретный датчик уровня

было принято решение использовать для аналоговых сигналов интерфейс «токовая петля» 4–20 мА, а для дискретных сигналов – 0...24 В.

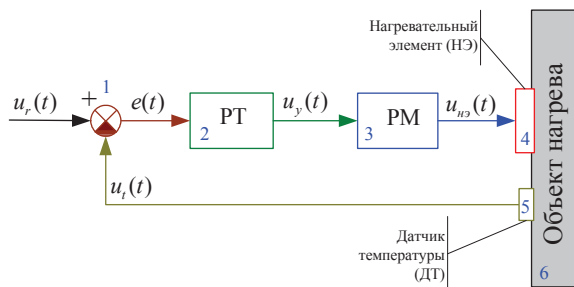


Рис. 4. Структурная схема контура регулирования температуры: $u_r(t)$ – значение сигнала уставки (задания), $e(t)$ – значение ошибки (величина рассогласования), $u_y(t)$ – сигнал управления, $u_{из}(t)$ – напряжение на нагревательном элементе, $u_f(t)$ – сигнал с датчика температуры

Сводные данные по средствам измерения и исполнительным элементам управления УПН представлены в табл. 1.

Таблица. Средства измерения и исполнительные элементы УПН

Подсистема	Средства измерения						Исполнительные элементы					
	TE	PE	FE	LAE	LE	NE	SVE	SVH	CV	CP	PCR	
Натрий	54	–	3	–	2	7	7	3	2	–	32	
Щелочь	24	4	2	–	2	3	10	7	3	4	19	
Вода	1	6	7	3	–	–	4	12	7	4	–	
Азот	4	21	9	–	–	–	4	–	5	–	4	
Отход. газ	8	–	1	–	–	–	2	–	–	–	6	
Пар	2	–	–	–	–	–	–	2	–	–	2	

TE – датчик температуры, LAE – аналоговый датчик уровня, NE – датчик протечек, CP – насос, PCR – тиристорный регулятор мощности.

Таким образом, общая информационная емкость контроллера сбора данных и управления должна обеспечивать ввод ста шестидесяти шести аналоговых и ста тридцати семи дискретных сигналов и формирование восьмидесяти аналоговых и семи-десяти дискретных сигналов управления.

По результатам анализа характеристик технических средств, представленных на казахстанском рынке средств автоматизации, а также изучения технико-коммерческих предложений поставщиков оборудования было определено, что большинство предлагаемых решений удовлетворяют основным требованиям, позволяя создать относительно равные по потребительским свойствам АСУ ТП. Наиболее рациональным вариантом решения поставленной задачи, ввиду лучшего соотношения цена/качество, явилась платформа ControlLogix производства компании Allen-Bradley.

Программный комплекс системы управления

Для решения комплекса поставленных задач была определена структурная схема прикладного программного обеспечения АСУ ТП УПН, представленная на рис. 5.

В состав комплекса программных средств системы управления УПН входят:

- RSView32 – SCADA (Supervisory Control And Data Acquisition) система, выполняющая функции представления информации о ходе технологического процесса, организации диалога с оператором установки, звуковой и световой сигнализации, первичной регистрации данных;
- RSLogix 5000 – программный пакет, в котором осуществляется программирование, отладка и редактирование программ контроллера сбора данных и управления;

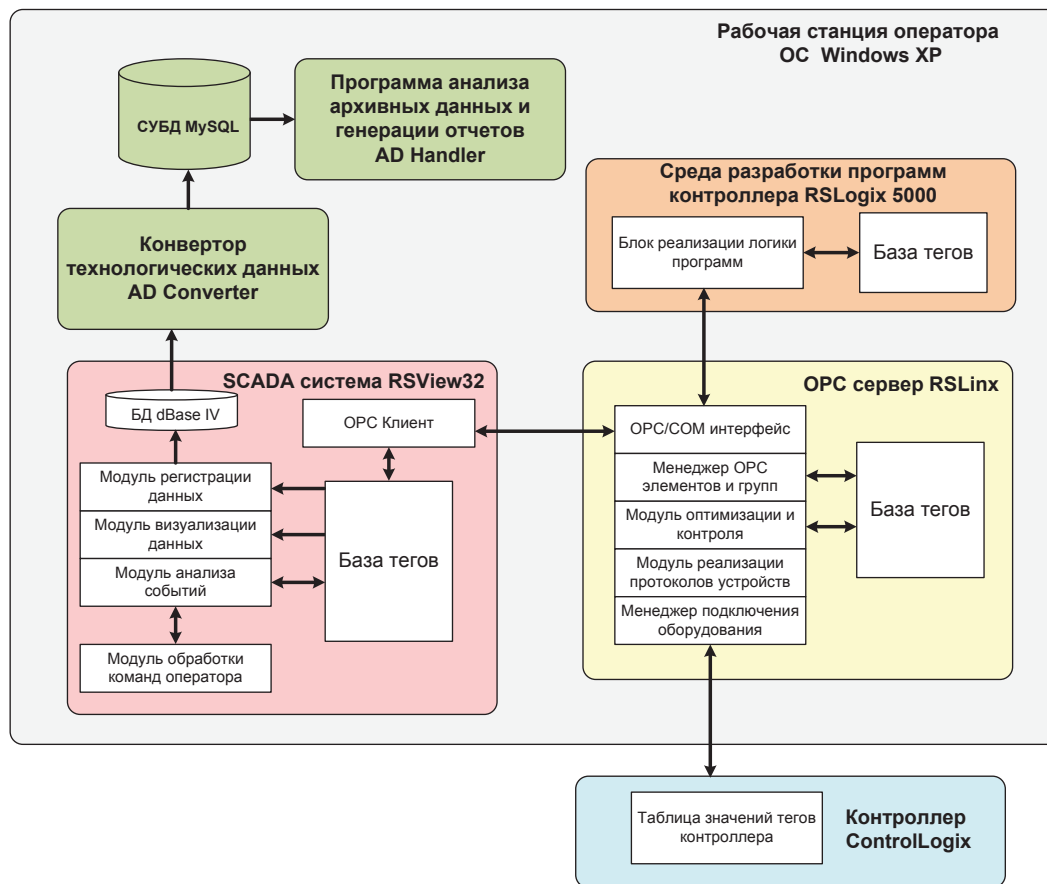


Рис. 5. Структурная схема программного обеспечения АСУ ТП УПН

- RSLinx – OPC (OLE for Process Control) сервер, выполняющий функции организации взаимодействия программных компонентов по сети ControlNet;
- AD Converter – прикладная программа, выполняющая преобразование первичных данных из формата dBase IV в СУБД MySQL для долговременного хранения;
- MySQL – СУБД (Система Управления Базами Данных), выполняющая функции основного хранилища истории изменения технологических параметров, тревожных событий и сообщений системы за все время работы УПН;
- AD Handler – прикладная программа, обеспечивающая доступ к архивным данным из MySQL, позволяющая выполнять просмотр и экспорт данных за требуемый временной интервал в текстовые файлы и файлы электронных таблиц Excel.

Графический интерфейс системы управления

Графический интерфейс сводится к системе видеокадров, текстовых и звуковых сообщений, снабженных необходимыми подсказками и помощью, при организации диалога оперативного персонала установки с программно-техническими средствами АСУ ТП.

Соблюдены следующие принципы компоновки мнемосхем:

- мнемосхемы упорядочиваются по слоям;
- слои отличаются друг от друга степенью детализации информации;
- чем выше слой, тем выше уровень обобщения представления информации на мнемосхемах этого слоя.

На рис. 6 приведена структура технологических экранов УПН.

Мнемосхемы разделены на следующие группы:

- технологических линий. Позволяют обеспечить контроль за технологическим процессом в реальном времени;
- управления. Реализуют взаимодействие оператора с исполнительным технологическим оборудованием, при этом обеспечивая отображение состояния контуров управления;
- просмотра архивов. Предоставляют средства просмотра архивных данных для событий системы и тревожных сообщений, позволяют получить отчет в электронном виде и твердой копии;
- контроля за оборудованием. Обеспечивают наглядное представление информации о неисправностях модулей ввода/вывода АСУ ТП УПН и оборудования электроснабжения установки.

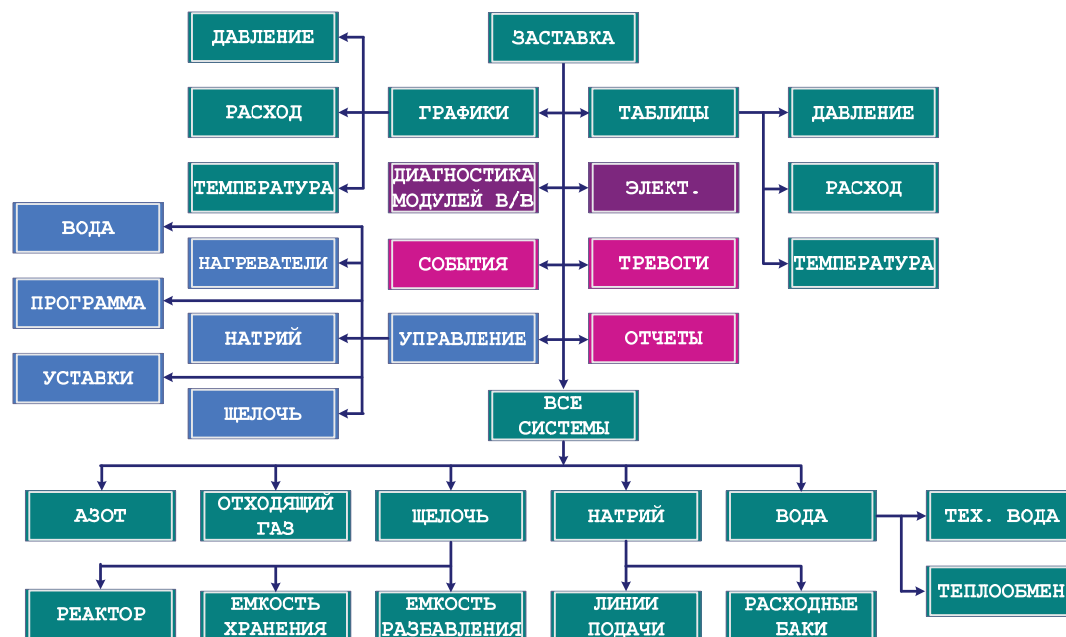


Рис. 6. Структура экранов проекта

Заключение

АСУ ТП УПН представляет собой комплекс технических и программных средств, обеспечивающих реализацию заданного набора функций, направленных на достижение конечного результата: обеспечение качественного и безопасного управления технологическим процессом переработки натрия. Разработка системы выполнена на основе анализа общего алгоритма управления и функционально-технологических схем различных подсистем УПН.

Окончательным подтверждением результатов проектирования любой разработки, и в частности АСУ ТП УПН, являются натурные испытания, ко-

торые дают ответ о качестве разработки и целесообразности ее дальнейшего использования. Так, пусконаладочные работы АСУ ТП УПН на объекте подтвердили соответствие характеристик разработанной конфигурации программно-технических средств системы управления как функциональным и техническим требованиям, так и требованиям эргономики и технической эстетики.

Исследование выполнено при поддержке Министерства образования и науки Российской Федерации, соглашение 14.В37.21.0457 «Разработка высокопроизводительного модульного приборного комплекса для автоматизированных систем экспериментальных исследований и управления электрофизическими установками ядерной энергетики».

СПИСОК ЛИТЕРАТУРЫ

1. Дорф Р. Современные системы управления. – М.: Лаборатория базовых знаний, 2002. – 832 с.
2. Абдулов Н.Д. Теория и методы проектирования оптимальных регуляторов. – Л.: Энергоатомиздат, 1985. – 324 с.
3. Кривцов П.Ю., Павлов В.М. Разработка модели химического реактора установки по переработки натриевого теплоносителя // Известия Томского политехнического университета. – 2011. – Т. 318. – № 3. – С. 41–46.

4. Кривцов П.Ю., Павлов В.М. Использование имитационного комплекса для отладки системы управления процессом переработки натрия // Известия Томского политехнического университета. – 2012. – Т. 320. – № 5. – С. 105–110.
5. Garcia H.E. Modeling and Control of a Sodium Conversion Process Applied to Nuclear Decommissioning Activities // Nuclear Technology. – 1998. – V. 123. – № 8. – P. 54–60.

Поступила 23.11.2013 г.

УДК 62–533.65

ПРОГРАММНЫЙ МОДУЛЬ ПРЕДУПРЕДИТЕЛЬНОЙ СИГНАЛИЗАЦИИ, АВАРИЙНОЙ ЗАЩИТЫ И ПОМОЩИ ОПЕРАТОРУ ВАКУУМНОЙ СИСТЕМЫ ТОКАМАКА КТМ

А.Г. Коровиков, В.М. Павлов*, Д.А. Ольховик

Институт атомной энергии НЯЦ Республики Казахстан, г. Курчатов

*Томский политехнический университет

E-mail: Korovikov@nnc.kz

Исследована система форвакуумной откачки вакуумной камеры токамака КТМ на модели, реализованной с помощью сетей Петри. Определены качественные характеристики модели. На основе данных модели запрограммирован модуль предупредительной сигнализации, аварийной защиты и помощи оператору.

Ключевые слова:

Система управления, математическая модель, сеть Петри, вакуумирование.

Key words:

Control system, mathematical model, Petri nets, vacuum pumping.

Введение

В настоящее время в г. Курчатов, Республика Казахстан, ведутся работы по реализации проекта строительства термоядерной установки «Казахстанский материаловедческий токамак КТМ».

Эффективность проведения экспериментальных исследований на термоядерных установках типа токамак определяется количеством и составом остаточного газа в объеме и состоянием поверхности разрядной камеры, которая является источником поступления в плазму различных примесей: паров воды, кислорода, углерода и т. п. Поэтому на этапе подготовки установки к эксперименту должна быть обеспечена откачка газов из рабочей камеры до 10^{-8} Торр.

Вакуумная система токамака КТМ относится к дискретно-событийным системам и состоит из двух подсистем:

- форвакуумная откачка обеспечивает получение в вакуумной камере давления 10^{-3} Торр, необходимого для последующего включения высоковакуумных средств откачки — турбомолекулярных насосов дежурной откачки;
- дежурная откачка обеспечивает получение в вакуумной камере давления 10^{-8} Торр.

При вакуумировании большой удельный вес имеют технологические процессы дискретного (переключательного) типа, характеризующиеся большим количеством логических операций, необходимых для управления вакуумной системой. Управление процессами переключательного типа осуществляется в соответствии с некоторой стратегией переключения, получаемой в виде алгоритма управления. Для составления формализованного описания данного производственного процесса переключательного типа с последующей реализацией по нему логического управления широко используется аппарат дискретной математики [1–3].

Алгоритм работы форвакуумной системы

Форвакуумная система выполнена на базе трех пластинчато-роторных насосов 2НВР-250Д с азотными ловушками (NL1.1–NL1.3) (рис. 1). Предель-

ное остаточное давление в системе измеряется приборами Varian: шестью низковакуумными термомпарными датчиками Convectorr (PT1.1–PT1.6) и одним вакуумметром Multi-Gauge. В начале процесса вакуумирования производится включение насосов 2НВР-250Д, параллельно производится включение вакуумметра и закрытие клапанов-напекателей соответствующего насоса (VF1.1–VF1.3). При достижении вакуума 10^{-3} Торр производится открытие задвижек 23ВЭ-100 (VTM1.2–VTM1.4), при достижении вакуума 10^{-3} Торр в объеме до клапана КУЭТ-100 (VM1.1) клапан плавно открывается, а при достижении вакуума 10^{-3} Торр в объеме до задвижки Vat (VTP1.1) задвижка открывается.

При плановой или аварийной остановке любого из насосов необходимо последовательно закрыть его задвижку, открыть (подать питание) его клапан-напекатель и отключить соответствующий насос. При плановой или аварийной остановке последнего из ранее работавших насосов необходимо закрыть все клапаны и задвижки системы, открыть все клапаны-напекатели и отключить насосы.

Моделирование

В качестве граф-модели дискретного процесса применена сеть Петри, позиции которой сопоставлены с операциями, а переходы — с условиями смены операций. Применение такой модели позволяет решать ряд задач [4, 5]:

- определить классы дискретных процессов (циклических, конвейерных и т. д.), их свойства, способы отображения, провести анализ и синтез через формальные понятия и методы сетей Петри;
- свести задачу проверки корректности дискретного процесса к исследованию сетей Петри;

Модель функционирования форвакуумной системы представлена сетью Петри (рис. 2), а ее содержательный смысл приведен в табл. 1 и 2. Сеть содержит 36 позиций b_1 – b_{36} и 23 перехода d_1 – d_{23} . Начальная маркировка соответствует состоянию, при котором все насосы системы выключены, клапаны и задвижки закрыты, открыты клапаны-напекатели. Записывая

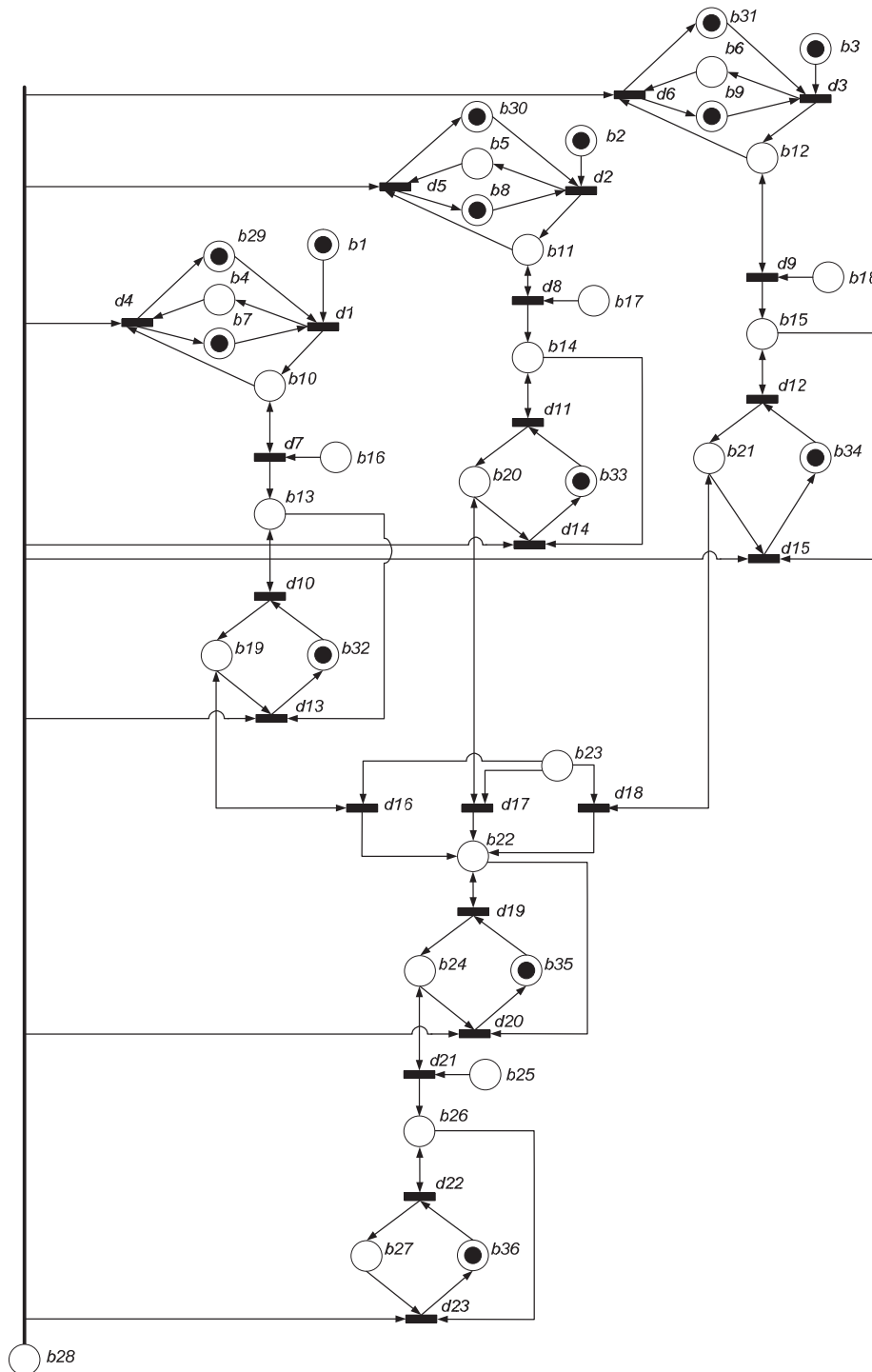


Рис. 2. Модель функционирования форвакуумной системы в виде сети Петри

- живой, имеется возможность срабатывания любого перехода при функционировании моделируемого объекта, и отсутствуют тупики;
- достижимой, имеется возможность перехода сети из одного заданного состояния (характеризуемого распределением меток) в другое. Дерево представляет все возможные последовательности запусков переходов. Всякий путь в дереве, начинающийся в корне, соответствует допустимой последовательности переходов.

Разработка программного модуля

Алгоритмы работы систем форвакуумной и дежурной откачки требуют четкого и согласованного управления исполнительными агрегатами систем (клапаны, задвижки, насосы) как во время регламентных работ, так и во время аварийных ситуаций или процедуры останова работы систем. Процесс вакуумирования камеры КТМ занимает длительное время (несколько суток), и постоянное участие оператора в управлении процессом нерационально.

ДЕЖУРНАЯ ОТКАЧКА

ЛОКАЛЬНЫЙ ПУЛЬТ

12:08:34

ГЛАВНЫЙ
ИНФОРМАЦИЯ
ДИАГНОСТИКА
ВСЕ СИСТЕМЫ
НАГРЕВ
КОНТРОЛЬ ЭМС
ТЕМПЕРАТУРА КАМЕРЫ
УУШП КИП
ФОРВАКУУМ
ДЕЖ. ОТКАЧКА
УУПО КИП
ЦЕНТР. ЗАЛ
ПОДГОТОВКА
СОБЫТИЯ
Ис источник
ТРМ202
Агрегаты
Уставки КИП
Вакууметры

От диверторного патрубка, 16 сектор

От экваториального патрубка, 12 сектор

От экваториального патрубка, 20 сектор

Дополнительные сообщения оператору

Сообщения отчета тревог

VP2.1 заблокирована

Время	Категория	Имя	Сообщение	Время квитирования
26.11.2012 09:34	РТ2.1		Превышено давление. Вкл. блокировка VP2.1	

91

но. Кроме того, ошибочные действия оператора или программного модуля автоматики могут привести к выходу из строя дорогостоящего оборудования.

С целью уменьшения человеческого фактора в управлении данными системами и, как следствие, увеличения их надежности разработан программный модуль предупредительной сигнализации, аварийной защиты и помощи оператору СУТП токамака КТМ, алгоритм работы которого основан на логике управления отработанной модели. Модуль выполняет следующие функции:

- переключение ручного и автоматического управления системами;
- предупредительное оповещение оператора;
- аварийная защита систем;
- динамическая настройка параметров каналов контроля;
- регистрация действий оператора и состояния агрегатов в лог-файл.

В качестве программного продукта реализации модуля используется SCADA-система TraceMode 6 и встроенный язык программирования ST стандарта МЭК 6-1131/3.

На рис. 4 показан экран оператора системы, на котором отображается состояние технологической схемы форвакуумной откачки.

Датчик	Расположение	Верхняя АЗ	Верхняя ПС	Значение	Нижняя ПС	Нижняя АЗ	Контроль установки
PT2.1	Куб ТТ2.1	18,0	16,0	2,0	1,0	0,3	ВКЛ
PT2.2	Куб ТТ2.2	15,0	13,5	4,8	1,0	0,3	ВКЛ
PT2.3	Куб ТТ2.3	15,0	13,5	3,5	1,0	0,3	ВКЛ
PT2.4	Проставок ТТ2.4	13,0	12,0	2,1	1,0	0,3	ВКЛ
PM2.1	Куб 2.1	18,0	16,0	5,2	2,0	1,0	ВКЛ
PM2.2	Куб 2.2	18,0	16,0	6,1	2,0	1,0	ВКЛ
PM2.3	Куб ТТ2.3	16,0	14,5	4,7	1,0	0,3	ВКЛ
PM2.4	Проставок ТТ2.4	14,0	12,5	2,8	1,0	0,3	ВКЛ
PT1.1	Крестовина ТТ1.3	14,0	12,5	4,4	1,0	0,3	ВКЛ
PT1.2	Крестовина ТТ1.5	14,0	12,5	4,6	1,0	0,3	ВКЛ
PT1.3	Крестовина ТТ1.6	14,0	12,5	4,3	1,0	0,3	ВКЛ

Рис. 5. Экран «Уставки КИП»

Настройка значений верхних и нижних предупредительных и аварийных границ датчиков систем осуществляется оператором системы с помощью экрана «Уставки КИП» (рис. 5). В соответствии с заданными критическими параметрами будут выводиться сообщения отчёта тревог и записываться сообщения в лог-файл. Сообщения оператору определяются в словарях на этапе наладки системы. Для удобства оператора состояние всех агрегатов вакуумных систем отображается на отдельном экране «Состояние агрегатов» (рис. 6).

СПИСОК ЛИТЕРАТУРЫ

1. Васильев В.В., Кузьмук В.В. Сети Петри, параллельные алгоритмы и модели мультипроцессорных систем. — Киев: Наукова думка, 1990. — 216 с.
2. Захаров В.Н., Поспелов Д.А., Хазацкий В.Е. Системы управления. Задание. Проектирование. Реализация. — М.: Энергия, 1977. — 424 с.
3. Лескин А.А., Мальцев П.А., Спиридонов А.М. Сети Петри в моделировании и управлении. — Л.: Наука, 1989. — 133 с.

Турбомолекулярные насосы подсистемы дежурной откачки работают под управлением собственных контроллеров. Программный модуль производит опрос контроллеров и производит анализ показаний датчиков (напряжение и ток на якоре, потребляемая мощность, частота вращения, температура на фланце и роторе). При выходе этих значений за допустимые пределы система управления производит отключение соответствующего турбомолекулярного насоса, выдавая при этом предупредительное или аварийное сообщение оператору.

Система форвакуумной откачки				Система дежурной откачки			
Название агрегата	Состояние агрегата			Название агрегата	Состояние агрегата		
Насос NL1.1	Отключен			Насос NR1.1	Отключен		
Насос NL1.2	Отключается			Насос NR1.2	Отключен		
Насос NL1.3	Включен			Клапан VE2.1	Закрыт		
Клапан VE1.1	Закрыт			Клапан VE2.2	Закрыт		
Клапан VE1.2	Закрыт			Задвижка VP2.1	Открыта		
Клапан VE1.3	Закрыт			Задвижка VP2.2	Открывается		
Клапан VM1.1	Открыт						
Задвижка VM1.2	Открыта						
Задвижка VM1.3	Закрыта						
Задвижка VM1.4	Закрыта						
Задвижка VM1.5	Закрыта						
Задвижка VM1.6	Закрыта						
Задвижка VTP1.1	Закрыта						

Рис. 6. Экран «Состояние агрегатов»

Выводы

На основе теории сетей Петри разработана модель функционирования технологических агрегатов, задействованных в процессе форвакуумной откачки вакуумной камеры токамака КТМ. Результаты анализа систем с помощью моделей послужили основой для разработки программного модуля предупредительной сигнализации, аварийной защиты и помощи оператору СУТП. Модуль позволил управлять длительными процессами вакуумирования камеры КТМ в автоматическом режиме и объединить в одной системе оборудование различных производителей. Таким образом, с применением программного контроля действий оператора и обеспечения предупредительной сигнализацией событий решена такая важная задача процесса подготовки вакуумной камеры к экспериментам, как обеспечение заданного уровня вакуума в камере с максимальной защитой дорогостоящего оборудования от выхода из строя при нештатных ситуациях и надежностью.

Исследование выполнено при поддержке Министерства образования и науки Российской Федерации, соглашение 14.B37.21.0457 «Разработка высокопроизводительного модульного приборного комплекса для автоматизированных систем экспериментальных исследований и управления электрофизическими установками ядерной энергетики».

4. Юдицкий С.А., Магергут В.З. Логическое управление дискретными процессами. Модели. Анализ. Синтез. — М.: Машиностроение, 1987. — 176 с.
5. Юдицкий С.А., Тагаевская А.А., Ефремова Т.К. Проектирование дискретных систем автоматики. — М.: Машиностроение, 1980. — 232 с.
6. Питерсон Дж. Теория сетей Петри и моделирование систем / пер. с англ. — М.: Мир. 1984. — 264 с.

Поступила 01.12.2012 г.

УДК 681.5.015+004.428

ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ МОМЕНТА СОПРОТИВЛЕНИЯ ПРОИЗВОДСТВЕННЫХ МЕХАНИЗМОВ

А.А. Филипас*, С.Н. Миколаенко*, С.Н. Кладиев

*Северский технологический институт НИЯУ «МИФИ»

Томский политехнический университет

E-mail: alf@ssti.ru

Предложен программный комплекс для задания характеристик момента сопротивления производственных механизмов. Комплекс разработан в среде визуального программирования MatLab Simulink и сохраняет все особенности базового программного продукта. Модели типовых производственных механизмов представлены в виде разработанной библиотеки элементов, которые получены из стандартных программных блоков. Интерфейсы блоков построены с учетом особенностей типовых производственных механизмов как нагрузок и позволяют учитывать количественные значения, определяющие конкретные параметры производственного механизма.

Ключевые слова:

Моделирование, момент сопротивления, библиотека подпрограммы, статические и динамические характеристики, стохастическая модель.

Key words:

Simulation, load torque, library routines, static and dynamic characteristics, stochastic model.

Основной целью при автоматизации производственных механизмов является синтез систем автоматического управления технологическими процессами на основе исполнительных электроприводов. При этом производственные механизмы, как нагрузки исполнительных регулируемых электроприводов, определяют методики выбора силового оборудования, особенности расчета статических и динамических характеристик, а также структуру и параметры системы управления [1–3]. Существенное упрощение этих задач возможно путём разработки специализированного программного обеспечения в системах визуального моделирования, таких как MatLab Simulink, при этом сохраняются все особенности базового программного продукта [4].

Предлагаемый блочно-модульный программный комплекс содержит и типовые модули для составления имитационной модели моментов сопротивления производственных механизмов. Модели типовых нагрузок производственных механизмов оформлены в виде библиотеки стандартизованных программных элементов, построенных методом имитационного структурного моделирования. Модули библиотеки нагрузки производственных механизмов разработаны на основе математического описания и представлены как структурные схемы, состоящие (в основном) из типовых блоков MatLab Simulink с общим интерфейсом, который ориентирован на конкретную типовую нагрузку. Интерфейсы блоков позволяют учитывать количественные значения, определяющие конкретные параметры производственного механизма.

В основу предлагаемого программного комплекса положен принцип декомпозиции, т. е. деления сложных систем на многоуровневые конструкции из взаимосвязанных элементов, объединяемые в подсистемы разных уровней. Применение этого

принципа позволило разработать модели нагрузок, которые имеют сложный нелинейный характер, в том числе и случайно зависящий от многих факторов.

С целью систематизации программного комплекса предлагается использование стандартных классификаций моделируемых моментов сопротивления, причем моменты могут быть представлены функциями с эндогенными или экзогенными переменными, определяемыми в том числе и временем. Моменты могут быть представлены статическими, динамическими и стохастическими блоками.

В качестве иллюстрации возможностей предлагаемого программного комплекса приведем несколько примеров блоков типовых нагрузок электропривода:

1. **Ступенчатый момент** применяется для задания приближённых нагрузочных диаграмм и задается: последовательностью пар (k_{RGi}, G_{RGi}) ; интервалом зависимой переменной; формулой постоянного (на интервале) момента. Для получения ступенчатой нагрузки задаётся постоянным моментом, а коэффициенты и условия их выбора вводятся в табличной форме. Разработанная модель блока *Step-modulus* и пример её работы показаны на рис. 1.
2. **Трапецевидный момент** реализуется на блоке *Trapezoidal saw*. Также в этом блоке предусмотрена задержка (G) и пауза (t). Введение в модель моментной паузы позволяет существенно упростить моделирование знакопеременного трапецевидного момента путем алгебраического сложения с ним статического постоянного отрицательного момента. Пример работы блока *Trapezoidal saw* показан на рис. 2.
3. **Кусочно-линейный аппроксимированный момент** реализуется блоком *Piecewise-linear approximation*.

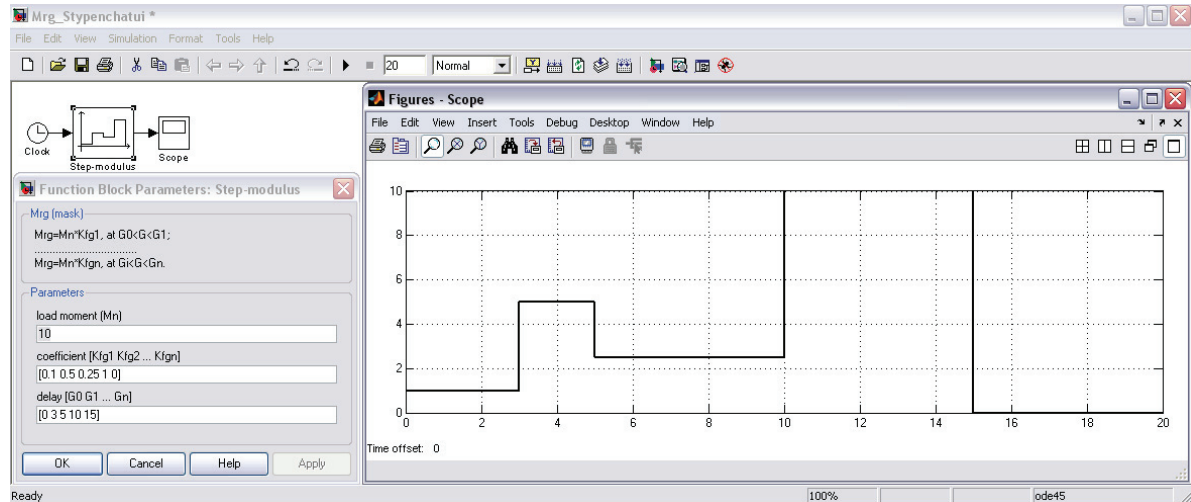


Рис. 1. Модель блока ступенчатой нагрузки

- Задается интервал изменения координаты (экзогенной либо эндогенной переменной) J_n (начало и конец), а также коэффициент K_{jn} для расчета линейного соотношения момента на этом интервале. Момент задается постоянной величиной, а J_n и K_{jn} – в виде таблицы, как показано на рис. 3.
4. **Фоновый момент** задается блоком *Amplitude-deterministic torque*, который создает амплитудно-детерминированную нагрузку, т. е. на вход блока подается амплитуда (возможно задание любой из статических или динамических зависимостей), которая умножается на случайный вектор с единичной амплитудой (рис. 4).

- В строке задания *Pause in the percentage* в процентном соотношении указывается вероятность получения нулевого значения. В строке *Sample time* задается частота изменения случайных величин. Так, в блоке предусмотрена возможность получения постоянных случайных величин установкой галочки в *Remembers the constant random*, где их вариация задается в строке *Version of a constant random variables*. Пример с заданием амплитуды в виде синусоиды и трапеции представлен на рис. 5.
5. **Частотный момент** сопротивления задается блоком *Frequency-deterministic torque*, который создает частотно-детерминированную нагрузку.

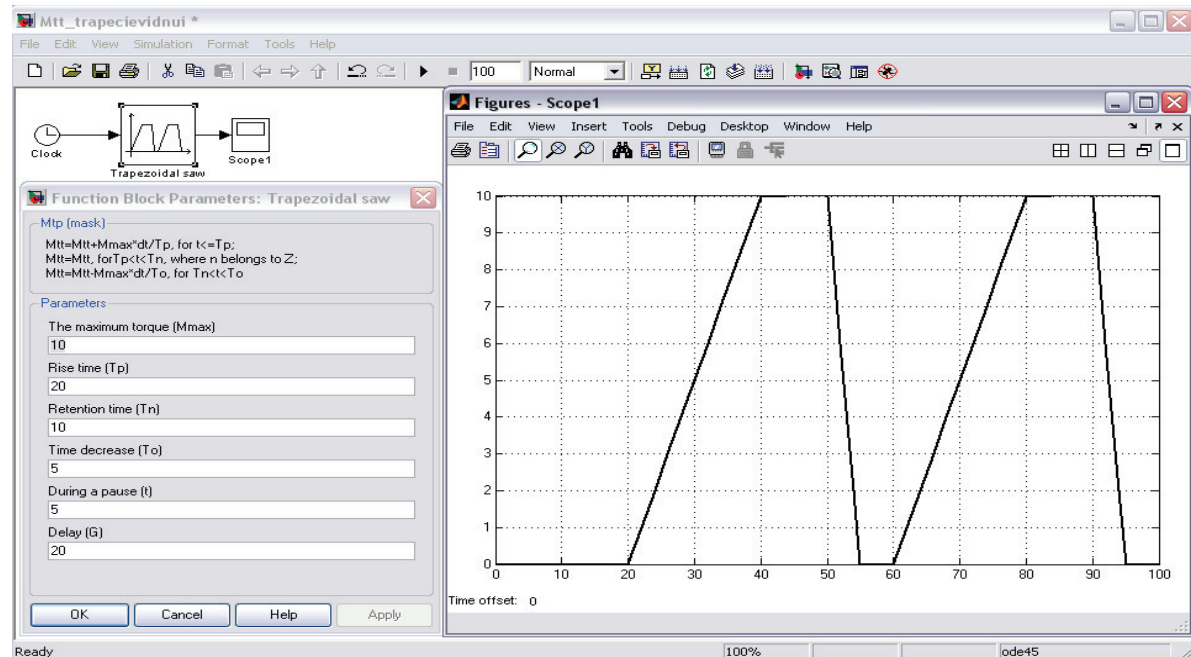


Рис. 2. Модель блока трапецевидной пилы нагрузки

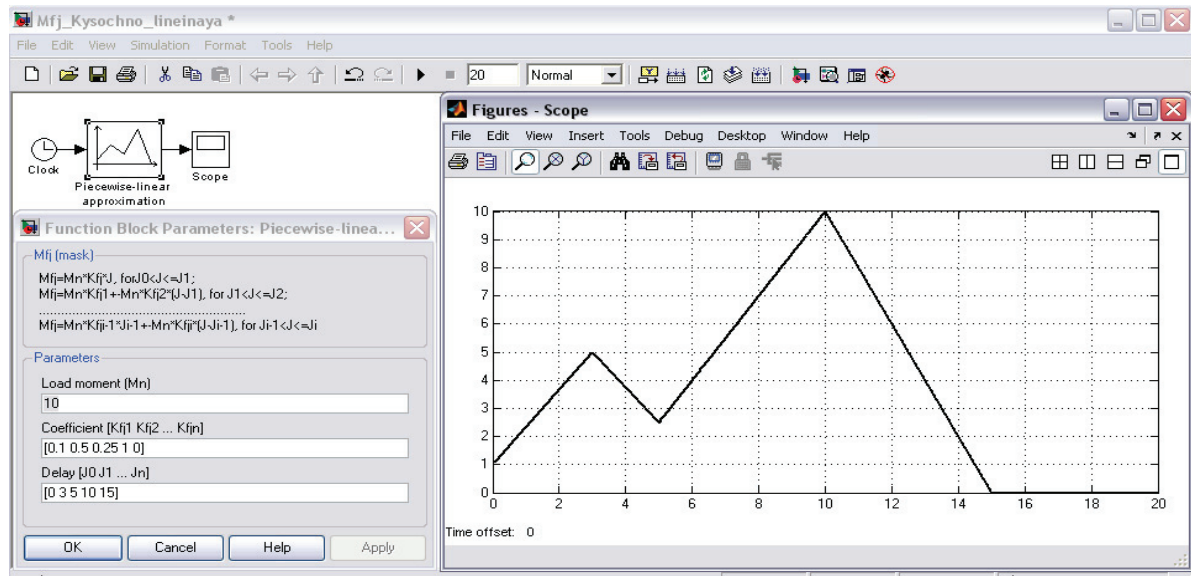


Рис. 3. Модель блока кусочно-линейной аппроксимированной нагрузки

В этом случае амплитуда случайных величин имеет значение от +1 до -1 (рис. 6).

В строке *Sample time* задается величина шага. Так же как и в блоке *Amplitude-deterministic* имеется возможность получения постоянных случайных величин. Пример с заданием амплитуды в виде синусоиды и трапеции представлен на рис. 7.

6. **Периодический момент** задается блоком *Harmonic*, который создает нагрузку с выраженной огибающей и переменной амплитудой. В строке *Deviation of the amplitude* задается предел отклонения входной амплитуды. Пример с заданием амплитуды в виде синусоиды и трапеции представлен на рис. 8.

Блоки предлагаемого программного комплекса позволяют в качестве исходных данных использовать любые переменные имитационной модели технологического процесса, а выходные функциональные зависимости применять как в модели, так и при формировании входных воздействий на физическую установку. Программный комплекс ориентирован на применение в среде MATLAB Simulink.

Предложенная библиотека нагрузок производственных механизмов электроприводов не является исчерпывающей, в дальнейшем планируется её расширение и совершенствование возможностей предложенного программного комплекса в части алгоритмического моделирования блоков.

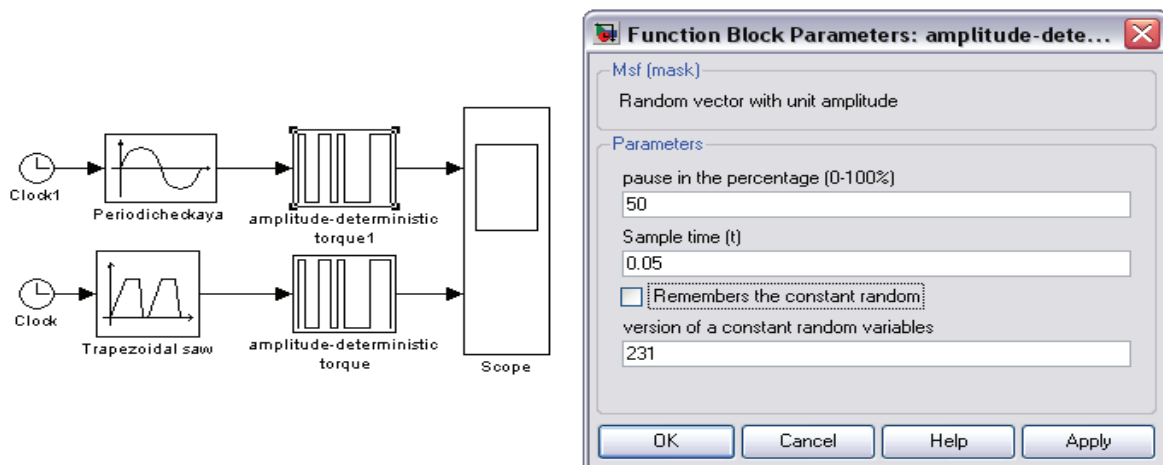


Рис. 4. Задания параметров модели блока фоновой нагрузки

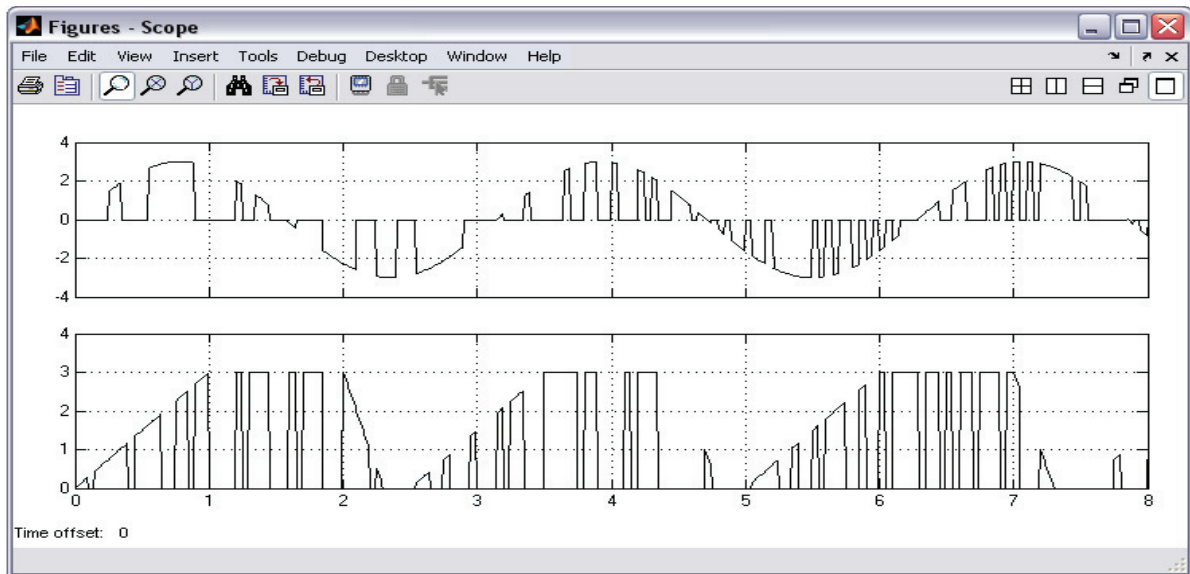


Рис. 5. Выходные характеристики блока *Amplitude-deterministic torque* с синусоидальной и трапецевидной амплитудой на входе

Выводы

Предлагаемый программный комплекс ориентирован на моделирование нагрузок типовых производственных механизмов с учетом их специфики и особенностей функционирования, что значительно ускоряет процессы исследования и проектирования автоматизированного электропривода и систем автоматического управления.

Интерфейсы стандартизованных блоков нагрузки построены с учетом сложившейся классификации нагрузок производственных механизмов,

что позволяет существенно упростить задание их количественных и качественных характеристик.

Принципы построения блоков программного комплекса позволяют расширять предлагаемую библиотеку путем коррекции существующих или синтеза новых моделей.

Предлагаемый программный комплекс позволяет реализовать принципы систем визуального программирования, алгоритмического построения моделей блоков и программирования на языках совместимых с MATLAB Simulink.

Работа выполнена в рамках Госзаказа «Наука» № 7.2826.2011.

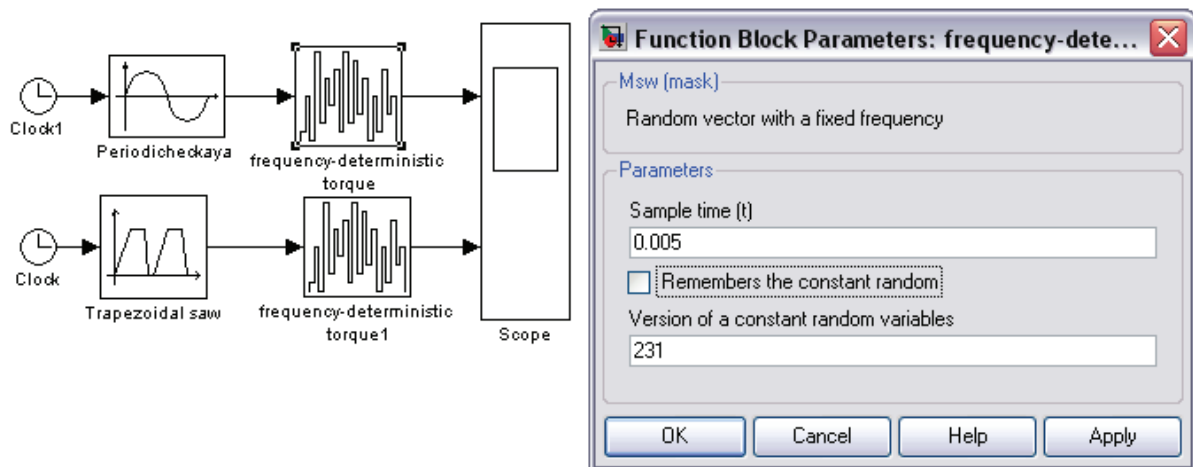


Рис. 6. Задания параметров модели блока частотной нагрузки

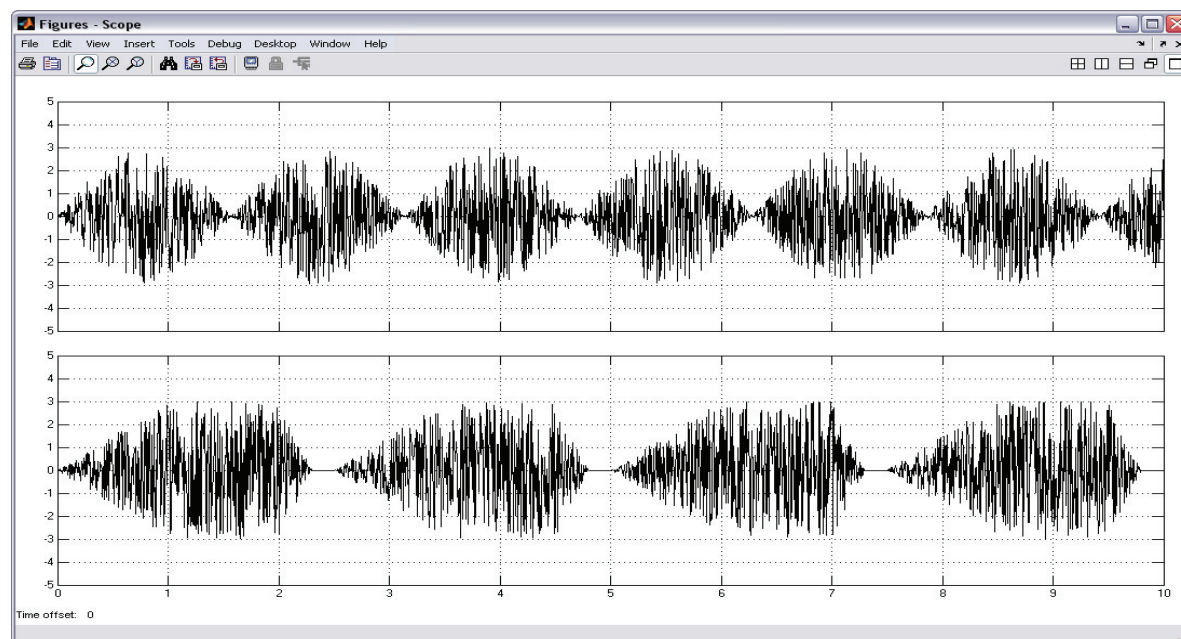


Рис. 7. Выходные характеристики блока *Frequency-deterministic torque* с синусоидальной и трапецевидной амплитудой на входе

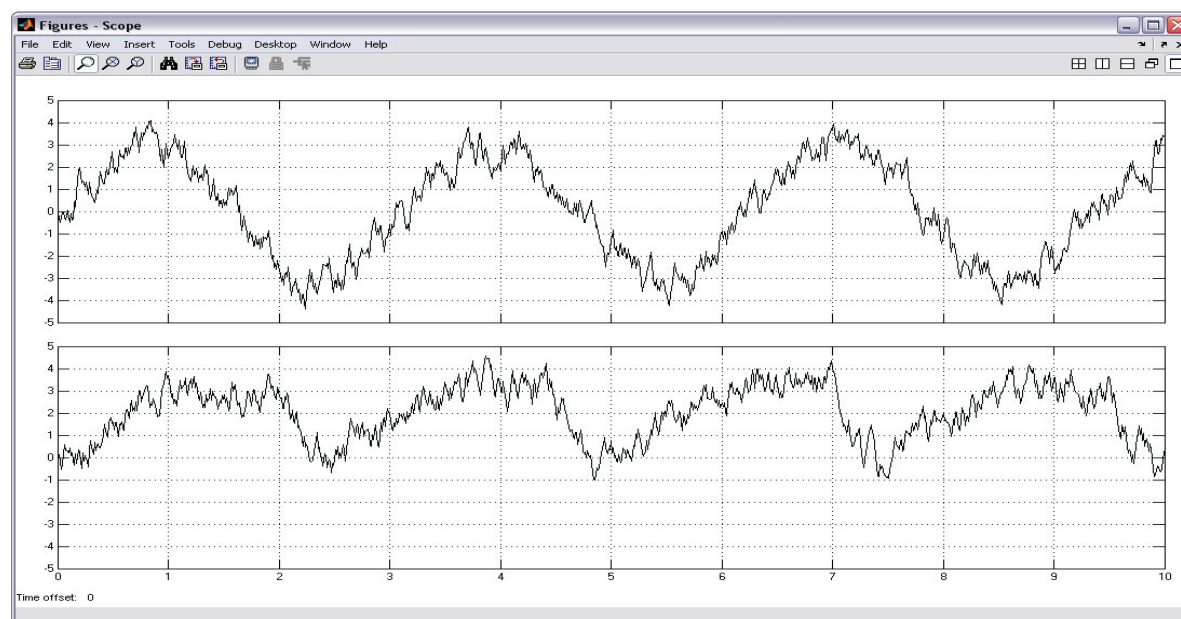


Рис. 8. Выходные характеристики блока *Harmonic* с синусоидальной и трапецевидной амплитудой на входе

СПИСОК ЛИТЕРАТУРЫ

1. Фролов Ю.М., Бурковский В.Л. Математическое моделирование в автоматизированном проектировании электроприводов. – Воронеж: Изд-во ВГТУ, 2000. – 143 с.
2. Слепокуров Ю.С. MATLAB 5. Анализ технических систем. – Воронеж: Изд-во ВГТУ, 2001. – 167 с.
3. Герман-Галкин С.Г. Компьютерное моделирование полупроводниковых систем в MatLab 6.0. – СПб.: КОРОНА принт, 2001. – 320 с.
4. Черных И.В. Моделирование электротехнических устройств в MATLAB, SimPowerSystems и Simulink. – М.: ДМК Пресс, 2008. – 288 с.

Поступила 28.01.2013 г.

Интеллектуальные системы

УДК 004.822:004.89

О КОНЦЕПЦИИ ИНТЕГРИРОВАННОЙ МОДЕЛИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

Ю.А. Загорулько

Институт систем информатики им. А.П. Ершова СО РАН, г. Новосибирск
E-mail: zagor@iis.nsk.su

Рассматривается концепция интегрированной модели, которая объединяет взаимодополняющие методы и средства представления знаний. На базе этой модели строятся язык представления и обработки знаний и инструментальные средства, обеспечивающие в полном объеме известные на данный момент потребности разработки баз знаний прикладных интеллектуальных систем.

Ключевые слова:

Интегрированная модель представления знаний, сетевая модель, продукционная модель, недоопределенная модель, императивная модель, онтология, объектно-ориентированная семантическая сеть, функциональная сеть.

Key words:

Integrated knowledge representation model, network model, production system model, subdefinite model, imperative model, ontology, object-oriented semantic network, functional network.

Введение

При создании интеллектуальной системы (ИС) самым трудоемким процессом остается разработка ее базы знаний. В связи с этим возникает насущная потребность в моделях представления знаний и построенных на их основе средствах создания баз знаний, обладающих не только высокой выразительностью и гибкостью, но и являющихся достаточно простыми и понятными в использовании как инженерам знаний — специалистам по представлению знаний, так и экспертам — специалистам в конкретных областях знаний.

В рамках работ по искусственному интеллекту накоплен широкий спектр средств и методов представления и обработки знаний. Так, семантические сети [1] используются как универсальная память для хранения любой информации, которую можно представить в терминах объектов и отношений между ними. Продукционные системы [2] рассматриваются как удобное средство для выражения операционной семантики понятий предметной области и организации логического вывода. Фреймы [3], представляющие объектно-ориентированный подход в искусственном интеллекте, служат для повышения уровня представления знаний. В рамках парадигмы программирования в ограничениях (constraint programming) [4] знания можно

задавать в виде системы ограничений на значения параметров объектов предметной области. Аппарат недоопределенных типов данных и разработанный на его основе метод недоопределенных вычислительных моделей [5] дают возможность оперировать неточно заданными значениями и объектами. Большим потенциалом обладают языки логического программирования [6], позволяющие описывать сложные процессы обработки информации в декларативном стиле. В последнее время при разработке интеллектуальных систем успешно применяются дескриптивные логики [7] и построенные на их основе языки и системы представления знаний [8]. Широкую популярность у разработчиков баз знаний получили онтологии [9], с помощью которых можно достаточно просто представить непротиворечивую (согласованную) систему понятий моделируемой области знаний.

Как правило, существующие инструментальные системы и оболочки для построения ИС снабжены каким-то одним из упомянутых выше средств представления и обработки знаний. В то же время при создании прикладных ИС часто возникает необходимость их комплексного использования, поскольку ни одно из этих средств, взятое в отдельности, не может обеспечить в полном объеме потребностей разработки реальной

прикладной системы. В связи с этим инструментальные средства разработки ИС должны включать различные взаимодополняющие друг друга методы и средства представления и обработки знаний. При этом возможны разные способы интеграции таких средств, начиная с интеграции на уровне подсистем и компонентов ИС и кончая интеграцией на уровне единого формализма.

Первый способ интеграции проще в реализации, но имеет ряд недостатков. Главным из них является отсутствие концептуального единства используемых методов и средств и как следствие этого — отсутствие концептуального единства создаваемых баз знаний. Это затрудняет разработку базы знаний, так как отдельные компоненты базы знаний приходится специфицировать не только с помощью разных средств представления знаний, но и использовать для этого разные программные системы, для интеграции которых в одну ИС приходится привлекать не только инженеров знаний, но и программистов. Также при данном способе интеграции приходится расходовать машинные ресурсы на организацию взаимодействия программных систем, реализующих отдельные методы и средства представления знаний.

Более предпочтительным выглядит другой способ интеграции, при котором все необходимые методы и средства объединяются в одном формализме (модели и/или языке представления знаний). Такая интеграция не только создает предпосылки для построения мощных баз знаний (за счет совместного использования различных взаимодополняющих друг друга методов и средств), но и обеспечивает концептуальное единство создаваемых баз знаний ИС, так как все компоненты базы знаний становятся связанными общими понятиями. На основе такого формализма могут быть созданы инструментальные средства, с помощью которых эксперты смогут сами строить базу знаний ИС в полном объеме, не прибегая к помощи посредников — инженеров знаний и программистов. Благодаря этому значительно упрощается разработка базы знаний ИС и повышается ее качество. Кроме того, интеграция всех требуемых средств в рамках одной модели (языка) создает предпосылки для создания на ее основе эффективных прикладных ИС, так как устраняются неизбежные расходы ресурсов на организацию взаимодействия программных систем, реализующих отдельные методы и средства представления знаний.

В статье рассматривается концепция интегрированной модели представления знаний, которая объединяет различные взаимодополняющие друг друга методы и средства представления знаний. На базе этой модели может быть построен язык представления и обработки знаний и инструментальные средства, обеспечивающие в полном объеме известные на данный момент потребности разработки баз знаний прикладных интеллектуальных систем.

Требования к интегрированной модели представления знаний

Для того чтобы модель представления знаний была пригодна для широкого класса интеллектуальных систем, она должна обеспечивать представление всех видов знаний, необходимых для разработки и функционирования ИС.

Прежде всего, в базе знаний ИС необходимо представлять знания о предметной области (ПО), в рамках которой будет функционировать система (рис. 1). Сюда входят знания об основных сущностях (понятиях и объектах) предметной области, а также знания о том, каким образом эти сущности связаны между собой. К последним относятся знания об отношениях, непосредственно связывающих понятия, а также более сложные виды знаний, представляющие различного рода зависимости между понятиями ПО, как логические, так и функциональные. В частности, при разработке большого класса ИС возникает необходимость в представлении знаний о ПО в виде правил «ЕСЛИ—ТО». Также довольно часто требуется представлять знания в виде ограничений на значения параметров, описывающих некоторый объект или систему объектов.

Для лучшего структурирования ПО и обеспечения более лаконичного ее описания важно иметь возможность выстраивания понятий предметной области в иерархию «общее—частное» и поддержки наследования свойств по этой иерархии.

Базы знаний большинства ИС также содержат конкретные знания из предметной области (предметные знания), которые представляются в виде экземпляров понятий (конкретных объектов) и связей между ними — экземпляров отношений или ограничений, заданных на значениях атрибутов экземпляров понятий.

Другим важным типом знаний, которые нужно представлять в ИС, являются знания о решаемых в рамках моделируемой ПО задачах и способах их решения (методах и алгоритмах). Эти знания характеризуют проблемную область (ПрО) интеллектуальной системы. Такого рода знания могут быть как декларативными, так и процедурными. К первому типу относятся знания, описывающие пространство решаемых ИС задач, включая разбиение задач на подзадачи и связь подзадач с методами их решения, ко второму — знания, представляющие как методы решения задач, так и конкретные алгоритмы.

Кроме знаний, представленных в БЗ, практически в любых ИС необходимо представлять знания, описывающие фрагмент действительности (ситуацию), который задает контекст, и входные данные для задач, решаемых ИС. Такие знания, как и предметные знания, обычно представляются в виде экземпляров понятий и отношений и/или ограничений, заданных на значениях атрибутов экземпляров понятий.

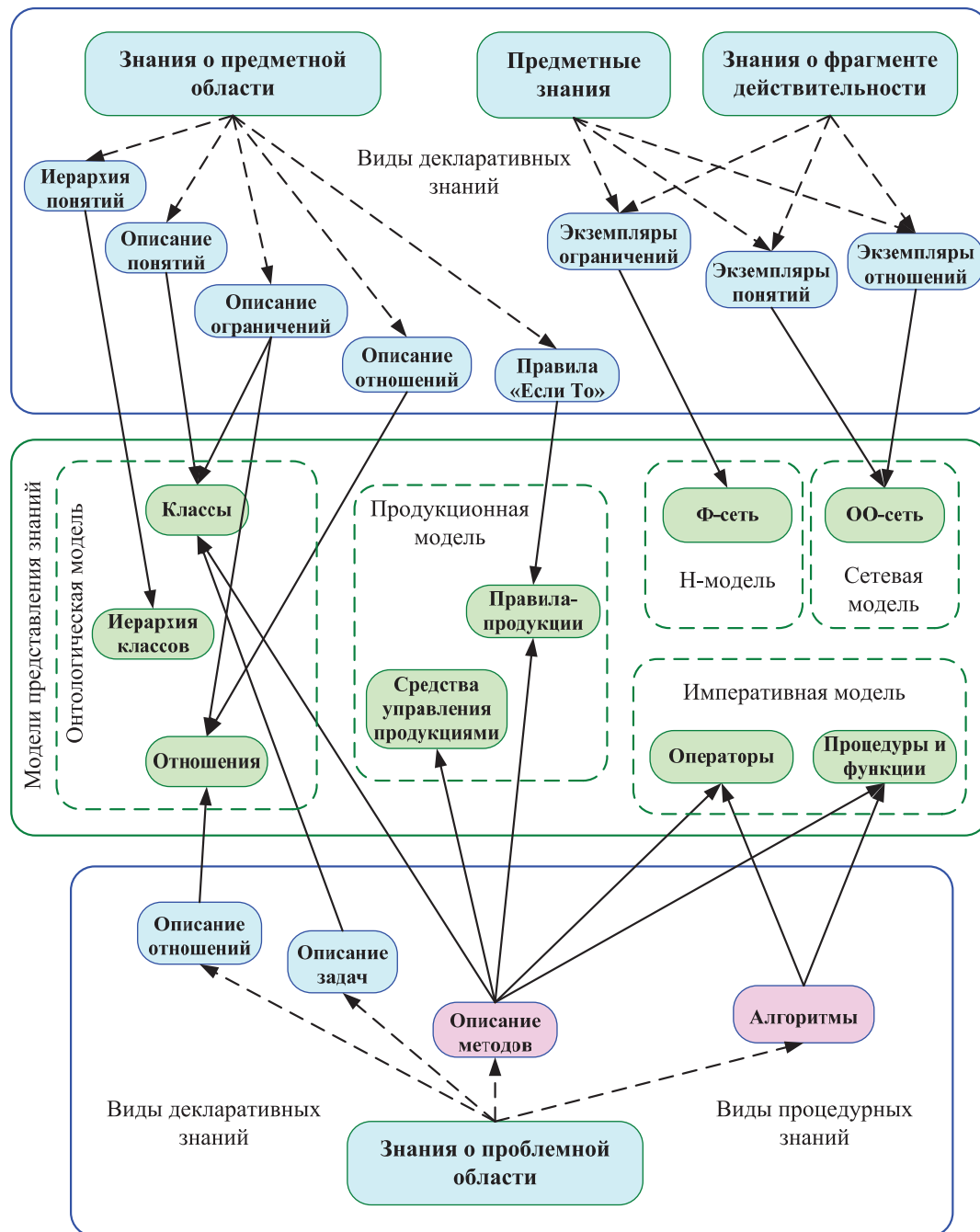


Рис. 1. Виды знаний и возможные способы их представления

При описании предметных знаний, контекста и входных данных задачи довольно часто возникает необходимость представлять знания об объектах, имеющих неточные значения атрибутов, путем задания оценок таких значений, например, в виде множеств возможных значений и/или ограничений на значения. Для представления таких знаний необходимо обеспечить возможность использования в экземплярах понятий атрибутов, имеющих недоопределенные значения, и задания экземпляров ограничений, связывающих такие атрибуты.

Рассмотренные выше виды знаний, необходимые для полноценного функционирования ИС,

не могут быть представлены средствами одной модели. Для этого требуются средства нескольких моделей. Так, для представления знаний о предметной области требуется задействовать средства двух моделей — онтологической и продукционной (рис. 1). Часть предметных знаний и знаний о фрагменте действительности может быть представлена средствами сетевой модели (в объектно-ориентированной семантической сети), другая часть — средствами недоопределенной модели (в функциональной сети или Ф-сети). Для представления знаний о проблемной области требуются средства трех моделей — онтологической, продукционной и императивной.

Таким образом, возникает необходимость в модели представления знаний, которая обладает набором методов и средств, достаточным для представления в ИС всех необходимых знаний. Прежде всего, такая модель должна обеспечивать представление как декларативных, так и процедурных знаний, что является необходимым требованием, так как без представления этих двух типов знаний невозможна разработка полноценной ИС. Кроме того, она должна обеспечивать представление и обработку как точных, так и неточных данных, что сделает ее пригодной для решения задач в условиях неопределенности знаний и данных.

Состав и изобразительные возможности интегрированной модели представления знаний

Исходя из перечисленных выше требований, в данной работе предлагается концепция интегрированной модели представления знаний, объединяющей в рамках одного формализма различные взаимодополняющие друг друга средства и методы представления и обработки знаний следующих моделей – онтологической, сетевой, продукционной, императивной и недоопределенной. (Последняя включает аппарат недоопределенных типов данных и разработанный на его основе метод недоопределенных вычислительных моделей [5], который, как показано в [10], фактически является универсальным методом программирования в ограничениях.) Выше было показано, что комбинация методов и средств представления знаний, предоставляемых этими моделями, обеспечивает в полном объеме известные на данный момент потребности прикладных ИС в представлении и обработке знаний.

Интегрированная модель представления знаний (ИМПЗ-модель) может быть представлена шестеркой:

$$M = \langle ONT, SN, FN, PS, IP, NF \rangle,$$

ONT – прикладная онтология, описывающая основные понятия и отношения (классы объектов и отношений) предметной и проблемной областей приложения; *SN* – объектно-ориентированная семантическая сеть (ОО-сеть), классы объектов и отношений которой, как и их свойства, определяются онтологией *ONT*; *FN* – функциональная сеть, включающая локальные вычислительные модели, заданные на значениях атрибутов объектов семантической сети *SN*; *PS* – система правил-продукций, описанных в терминах классов и отношений онтологии *ONT*; *IP* – средства императивного программирования; *INF* – общий механизм вывода/обработки знаний и данных, представленных в ИМПЗ-модели, объединяющий механизмы вывода/обработки каждой из интегрируемых моделей.

Представление знаний о предметной области ИС и решаемых ею задачах обеспечивается прикладной онтологией.

При этом онтология включает:

- 1) конечное непустое множество классов *C*, представляющих понятия (сущности) ПО;

- 2) конечное множество бинарных отношений *R*, определенных на классах *C*;
- 3) конечное множество атрибутов *A*, описывающих декларативные свойства классов и отношений;
- 4) множество типов данных *T*, «поставляющих» значения для атрибутов;
- 5) множество ограничений *F*, заданных на атрибутах классов понятий;
- 6) множество математических свойств бинарных отношений *P*.

Все классы онтологии *C* выстроены в иерархию «общее–частное», по которой организуется наследование свойств (включая атрибуты, отношения и ограничения) вышестоящих классов нижестоящими.

Благодаря включению в ИМПЗ-модель аппарата недоопределенных типов данных все простые типы данных, входящие в *T*, имеют недоопределенные расширения, обеспечивающие возможность представления недоопределенных значений в виде интервала или множества допустимых значений и оперирования ими. Это позволяет, в частности, задавать атрибутам объектов ОО-сети недоопределенные значения.

Функциональные зависимости между атрибутами сущностей $A_i \in A$ могут задаваться в описаниях классов объектов и отношений в виде ограничений $F_i \in F$, связывающих значения атрибутов объектов. (В описаниях бинарных отношений ограничения задаются на атрибутах их аргументов.) Такие ограничения представляют собой множество логических выражений (формул), связывающих арифметическими, теоретико-множественными и логическими операциями значения атрибутов одного или нескольких объектов. Следует заметить, что благодаря включению в ИМПЗ-модель метода недоопределенных моделей с помощью ограничений задаются не только условия корректности значений связанных ими атрибутов, но и функциональные интерпретации каждого из этих значений относительно других, что обеспечивает потенциальную возможность вычисления (уточнения) неизвестных (неточных) значений атрибутов через известные (или более точные).

В зависимости от потребностей представления знаний о ПО любому бинарному отношению $R_k \in R$ может быть «приписано» одно или несколько непротиворечащих друг другу математических свойств $P_m \in P$ из следующего набора – рефлексивность, симметричность, транзитивность, антирефлексивность, антисимметричность, – а также задано обратное отношение.

Конкретные знания о ПО, как и знания о моделируемом в ИС фрагменте действительности или контексте задачи, представляются в объектно-ориентированной семантической сети *SN*, включающей множество экземпляров классов (объектов) и отношений. Для представления этого вида знаний также служат экземпляры ограничений, накладываемых на атрибуты объектов (экземпляров классов) ОО-сети и хранимые в функциональной сети, связанной с ОО-сетью через атрибуты объектов.

Так, если в описании какого-либо класса или отношения заданы ограничения, то при создании каждого его экземпляра на основе этих ограничений строится локальная вычислительная модель (ЛВМ), вершинами которой являются входящие в ограничения атрибуты объекта (в случае отношения — атрибуты его объектов-аргументов), а дугами — существующие между ними функциональные отношения.

Построенные таким образом ЛВМ всех объектов и экземпляров отношений ОО-сети объединяются в глобальную вычислительную модель, которую будем называть функциональной сетью *FN*. Для уточнения значений слотов объектов, ЛВМ которых представлены в функциональной сети, используется метод недоопределенных вычислительных моделей [5].

Описание предметных знаний в виде правил «ЕСЛИ–ТО» средствами системы продукционных правил *PS* является естественным и удобным для многих областей. В интегрированной модели система продукционных правил работает над семантической сетью, состав понятий и отношений которой определяется онтологией. В связи с этим посылка и заключение любого правила могут описываться в терминах классов и отношений онтологии. Это позволяет не только удобно представлять экспертные знания в виде правил «ЕСЛИ–ТО», но и строить довольно мощные по своим изобразительным и операционным возможностям правила и с их помощью описывать на достаточно высоком уровне довольно сложные процессы вывода и обработки информации.

Рассмотрим, какие средства предоставляет ИМПЗ-модель для представления знаний о решаемых в рамках моделируемой ПрО задачах и методах их решения.

Знания о пространстве задач могут быть представлены в прикладной онтологии, где описываются типы задач и методов, предназначенных для их решения, и задается разбиение задач на подзадачи и связь задач с методами их решения (знания второго вида также могут быть представлены средствами ОО-сети). При этом сами методы решения могут быть представлены продукционными правилами, объединенными в группы и/или продукционные модули. Для этих целей в модель включаются средства структурирования множества правил и управления ими, такие как операторы активации и деактивации групп правил, которые могут сочетаться с императивными средствами управления.

Для поддержки эффективной реализации методов и алгоритмов, а также обеспечения создания полноценных прикладных ИС в ИМПЗ-модель включены традиционные для императивной модели (парадигмы программирования) средства. Они применяются для разработки алгоритмов в случаях, когда их реализация другими методами (например, с помощью продукционных правил) была бы слишком неэффективной по времени и/или затратной по памяти, а также для реализации численных вычислений и обеспечения интерфейсных возможностей ИС.

В дополнение к традиционным императивным средствам, таким как процедуры и функции, операторы присваивания, ветвления, цикла и т. п., ИМПЗ-модель включает механизм альтернатив, который является мощным средством поддержки процессов вывода и обработки информации в условиях, когда неточно заданы не только объекты, но и отношения между ними. Данный механизм реализуется оператором альтернативы, который позволяет задавать и перебирать альтернативы (варианты вычислений), обеспечивая при возникновении противоречия во время обработки очередной из них переход к следующей альтернативе.

Интеграция указанных методов и средств выполняется на основе объектно-ориентированного подхода. При этом:

- понятия и отношения описываемых ПО и ПрО представляются классами объектов и отношений онтологии *ONT*, инкапсулирующими в себе семантические свойства и ограничения на свои атрибуты и аргументы;
- объекты (экземпляры классов) и экземпляры отношений представляются в семантической сети *SN*, структура и свойства которой определены в онтологии;
- ограничения *F* на значения атрибутов объектов представляются в функциональной сети *FN*, связанной через эти атрибуты с семантической сетью *SN*;
- все необходимые процессы вывода и обработки информации (реализующие процесс решения задачи) осуществляются системой продукционных правил *PS*, работающей над семантической сетью, механизмом удовлетворения ограничений, реализуемым в функциональной сети, а также средствами императивного программирования *IM*.

Взаимосвязи интегрируемых моделей показаны на рис. 2. Здесь видно, что сетевая и продукционная модели используют сущности, определенные в онтологической модели. В частности, структура и свойства элементов семантической сети определяются в онтологии, а посылки и заключения продукционных правил описываются в терминах классов и отношений онтологии. В свою очередь, средства оперирования, предоставляемые онтологической моделью, используют средства оперирования сетевой модели (для того, чтобы привести ОО-сеть в соответствие с онтологией). Средства оперирования продукционной модели используют и функционал сетевой модели (для сопоставления посылок правил с семантической сетью и выполнения действий над нею, определенных в правилах), и средства оперирования, заданные в императивной модели. Кроме того, сетевая модель использует средства оперирования недоопределенной модели, а последняя — сущности, определенные в сетевой модели.

Выводы

В связи с отсутствием инструментальных систем, обеспечивающих в полном объеме потребности разработчиков баз знаний интеллектуальных

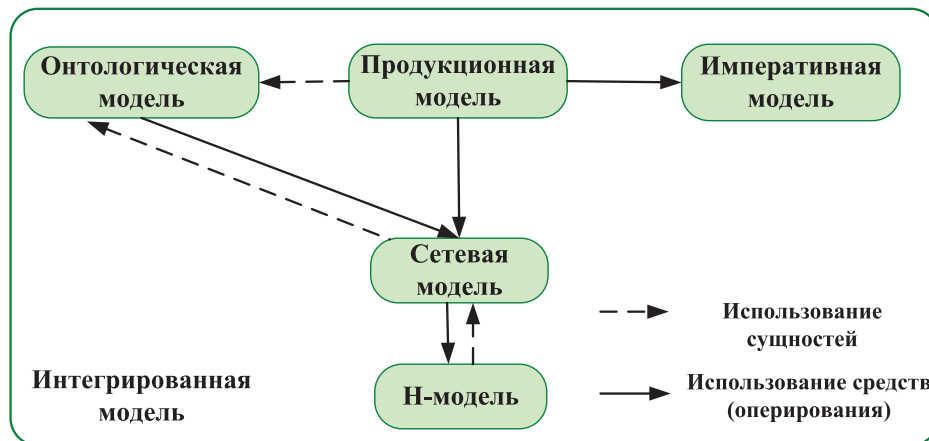


Рис. 2. Взаимодействие интегрируемых моделей

систем, в качестве базиса для представления знаний в ИС предложена модель, интегрирующая взаимодополняющие друг друга методы и средства представления и обработки знаний следующих моделей: онтологической, сетевой, продукционной, недоопределенной и императивной.

Благодаря тому, что интеграция указанных методов и средств выполнена на основе объектно-ориентированного подхода, ИМПЗ-модель позволяет представлять одновременно декларативные и процедурные знания, а также знания о вычислениях (т. е. знания, позволяющие уточнять значения параметров описываемого объекта или ситуации). Кроме того, такой способ интеграции позволяет добиться эффективной реализации интеллектуальной системы и сделать доступными все интегрируемые средства в рамках одного языка представления и обработки знаний.

На основе ИМПЗ-модели построен объектно-ориентированный язык представления и обработки знаний, который включает средства для совместно-

го описания как декларативных, так и вычислительных свойств понятий и отношений некоторой ПО, а также средства оперирования их экземплярами в продукционном стиле. Благодаря этому на таком языке можно естественным образом описывать и решать широкий класс задач, в том числе задачи, требующие сочетания логического вывода и вычислений над неточно заданными значениями.

Набор языковых и программных средств, реализующих интегрированную модель представления знаний, образует интегрированную систему представления и обработки знаний, которая послужила базисом инструментальных средств разработки интеллектуальных систем. На основе этих средств были построены оболочки, предназначенные для разработки различных типов систем, основанных на знаниях, — от оболочек информационных систем, обеспечивающих пользователю содержательный доступ к знаниям и данным определенной области знаний [11], до оболочек экспертных систем и систем поддержки принятия решений [12].

СПИСОК ЛИТЕРАТУРЫ

1. Deliyanni A., Kowalski R.A. Logic and Semantic Networks // Comm. ACM. — 1979. — V. 22. — № 3. — P. 184–192.
2. Davis R., King J. An overview of production systems // Computer Science Department Stanford University, Report STAN-CS-75–524. — 1975. — P. 38.
3. Минский М. Структура для представления знания // Психология машинного зрения. — М.: Мир, 1978. — С. 249–338.
4. Mayoh B. Constraint Programming and Artificial Intelligence // Constraint Programming. NATO ASI Series F: Computer and Systems Sciences. — 1993. — V. 131. — P. 17–50.
5. Нариньяни А.С. Недоопределенность в системах представления и обработки знаний // Изв. АН СССР. Техн. кибернетика. — 1986. — № 5. — С. 3–28.
6. Братко И. Алгоритмы искусственного интеллекта на языке Prolog. — М.: Издательский дом «Вильямс», 2004. — 637 с.
7. Baader F., Calvanese D., McGuinness D.L., Nardi D., Patel-Schneider P.F. The Description Logic Handbook: Theory, Implementation, Applications. — Cambridge: Cambridge University Press, 2003. — 574 p.
8. OWL Web Ontology Language Guide. W3C Recommendation 10 February 2004. URL: <http://www.w3.org/TR/owl-guide/> (дата обращения: 10.09.2012).
9. Guarino N. Formal Ontology in Information Systems // Proceedings of FOIS'98. — Trento, Italy, 6–8 June 1998. — Amsterdam: IOS Press, 1998. — P. 3–15.
10. Telerman V., Ushakov D. Data Types in Subdefinite Models // Art. Intell. and Symbolic Mathematical Computation, Lecture Notes in Computer Science / ed. J. Calmet et al. — Berlin: Springer, 1996. — V. 1138. — P. 305–319.
11. Загоруйко Ю.А. Программная оболочка для построения интеллектуальных научных Интернет-ресурсов // Информационные и математические технологии в науке и управлении: Труды XVII Байкальской Всеросс. конф. — Иркутск: ИСЭМ СО РАН, 2012. — Т. 3. — С. 50–56.
12. Загоруйко Г.Б., Загоруйко Ю.А. Проект открытого расширяемого инструментария для построения интеллектуальных СППР // Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2011): Матер. Междунар. научно-техн. конф. — Минск: БГУИР, 2011. — С. 233–236.

Поступила 14.09.2012 г.

УДК 004.942

ПОСТРОЕНИЕ МОДЕЛИ ДОВЕРИЯ К ОБЪЕКТАМ АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ПРЕДОТВРАЩЕНИЯ ДЕСТРУКТИВНЫХ ВОЗДЕЙСТВИЙ НА СИСТЕМУ

А.А. Бешта, М.А. Кирпо

Волгоградский государственный университет

E-mail: abewta@rambler.ru

Описана формальная модель деструктивного воздействия на систему, и выделены основные возможные деструктивные воздействия. Описана предложенная авторами модель доверия к объекту автоматизированной информационной системы на основе мониторинга действий объекта. Предложена сервис-ориентированная концепция предоставления услуг по обеспечению безопасности информации и реализующая ее многоагентная распределенная архитектура.

Ключевые слова:

Многоагентная система, деструктивное воздействие, оценка доверия к объекту, сервис-ориентированная архитектура.

Key words:

Multiaгент system, destructive influence, object trust assessment, service oriented architecture.

Автоматизированные информационные системы (АИС), реализуемые на компьютерах, состоят из множества разнородных, распределенных в пространстве компонентов: аппаратных средств, системных утилит, пользовательских приложений, средств защиты информации, массивов данных и персонала. При этом данные компоненты генерируют множество событий, связанных с действиями пользователей, процессами обработки информации, получением и отправкой сетевых пакетов, доступом к файлам. Дальнейший анализ этих событий позволяет выявить различные нарушения информационной безопасности.

Для повышения эффективности работы АИС, снижения нагрузки на ресурсы и уменьшения объема анализируемой информации предлагается использовать сервис-ориентированную архитектуру системы защиты информации. Кроме того, подобная архитектура позволяет повысить гибкость и масштабируемость АИС и легко интегрировать в нее новые компоненты [1].

Формально АИС можно представить в виде:

$$S = \{E^T, R^E\},$$

где E^T — множество объектов различных типов T

информационной системы S , $E^T = \bigcup_{i=1,n} E_i^T$, где n —

количество объектов класса T ; R^E — множество отношений между объектами, $R^E = E^T \times E^T$.

Каждый класс E^T представляет собой совокупность однотипных объектов, описанных с помощью набора свойств:

$$E_i^T = \bigcup_{k=1,m} P_k^{E_i^T},$$

где $P_k^{E_i^T}$ — k -е свойство i -го объекта E_i^T типа T .

Все объекты АИС могут быть разделены на две группы: активные и пассивные. Активные объекты могут выполнять различные операции над другими объектами (пассивными и активными), в том числе над собой. К пассивным объектам относятся все другие объекты.

Типы объектов T можно определить следующим набором:

$$T = \tilde{T} \cup \bar{T} = \langle C, U \rangle \cup \langle H, A, R \rangle,$$

где $\tilde{T} = \langle C, U \rangle$ — активные объекты типов C (программный компонент, процесс) и U (пользователь), $\bar{T} = \langle H, A, R \rangle$ — пассивные объекты типов H (аппаратный компонент), A (актив), R (разрешение).

Тогда система будет состоять из следующих типов объектов $E^T = \{E^H, E^C, E^U, E^A, E^R\}$, где E^H — множество узлов; E^C — множество компонентов; E^U — множество пользователей; E^A — множество активов; E^R — множество разрешений.

Множество отношений между объектами R^E описывается непересекающимся объединением:

$$R^E = R_{is}^E \cup R_{part}^E, R_{is}^E \cap R_{part}^E = \emptyset,$$

где R_{is}^E — отношение категоризации между объектами; R_{part}^E — отношение принадлежности между объектами.

Такое представление системы позволяет определить распределение отдельных объектов и их свойств по АИС, то есть:

$$p(E_i^T) = \frac{\sum E_i^T}{\sum E^T},$$

где $p(E_i^T)$ — доля объектов E_i^T среди объектов определенного типа E^T .

$$p(P_k^{E_i^T}) = \frac{\sum P_k^{E_i^T}}{\sum P^{E^T}},$$

где $p(P_k^{E_i^T})$ — распределение свойства $P_k^{E_i^T}$ среди свойств объектов типа E^T .

Таким образом, несмотря на разнородность компонентов АИС, можно определить конечное множество типовых объектов, характерных для всей системы, и выделить свойства, идентичные для многих объектов.

Для определения деструктивного воздействия на объекты АИС сначала необходимо определить

нормальное воздействие, осуществляемое активными объектами в процессе работы.

Нормальное допустимое воздействие определяется в виде правила R , установленного в системе, которое устанавливает допустимое действия Act для активного объекта по отношению к другому объекту:

$$R = (E_i^T \rightarrow Act) | E_j^T.$$

Каждое правило означает, что активный объект E_i^T может выполнить некоторое действие по отношению к объекту E_j^T .

В качестве множества простых действий Act можно определить:

$$Act = \langle Read, Write, Execute \rangle,$$

где $Read$ – операция чтения; $Write$ – операция записи; $Execute$ – операция выполнения.

Тогда нормальное воздействие $Sa_{E_i^T}$, осуществляемое активным объектом E_i^T , описывается следующим образом:

$$Sa_{E_i^T} \rightarrow P_k^{E_i^T} = v | \exists R = (E_i^T \rightarrow Act) | E_j^T$$

и означает, что некоторый объект E_i^T системы имеет свойство $P_k^{E_i^T}$ со значением v , которое позволяет активному объекту E_i^T выполнить действие Act по отношению к некоторому объекту E_j^T (возможно, тому же самому).

Но может существовать такое значение $v^* \neq v$ свойства $P_k^{E_i^T}$, что оно позволяет создать в системе новое правило $R^* \notin R$, в котором нарушается установленное соотношение между E_i^T , Act , E_j^T , что означает выполнение в системе действия Na , которое не установлено в системе в виде правила. Такое действие является деструктивным по отношению к объекту E_j^T и может нанести вред как отдельным объектам, так и всей автоматизированной системе в целом:

$$Na_{E_i^T} \rightarrow P_k^{E_i^T} = v^* | \exists R^* = (E_i^T \rightarrow Act) | E_j^T, R^* \notin R,$$

то есть у некоторого объекта E_i^T существует такое значение v^* некоторого его свойства $P_k^{E_i^T}$, что имеется возможность в обход установленных в системе правил R установить новое правило R^* , позволяющее для некоторого активного объекта E_i^T выполнить какое-то действие Act по отношению к некоторому объекту E_j^T .

Теперь можно описать различные типы деструктивных воздействий на объекты АИС в терминах этой модели.

Ознакомление:

$$\begin{aligned} Know_{E_i^T} \rightarrow P_k^{E_i^T} &= v^* | \exists R^* = \\ &= (E_i^T \rightarrow Read) | E_j^T, R^* \notin R. \end{aligned}$$

Копирование:

$$\begin{aligned} Copy_{E_i^T} \rightarrow P_k^{E_i^T} &= v^* | \exists R^* = \\ &= (E_i^T \rightarrow Write) | E_j^T, R^* \notin R, i \neq j. \end{aligned}$$

Модификация:

$$\begin{aligned} Modify_{E_i^T} \rightarrow P_k^{E_i^T} &= v^* | \exists R^* = \\ &= (E_i^T \rightarrow Write) | E_j^T, R^* \notin R. \end{aligned}$$

Удаление:

$$\begin{aligned} Rem_{E_i^T} \rightarrow P_k^{E_i^T} &= v^* | \exists R^* = \\ &= (E_i^T \rightarrow Write) | Null, R^* \notin R. \end{aligned}$$

Запуск:

$$\begin{aligned} Exec_{E_i^T} \rightarrow P_k^{E_i^T} &= v^* | \exists R^* = \\ &= (E_i^T \rightarrow Execute) | E_j^T, R^* \notin R. \end{aligned}$$

С другой стороны, для всех активных объектов АИС можно поставить в соответствие уровень доверия к объекту $B_{E_i^T} \in (0, 1)$.

В данном случае под уровнем доверия к объекту понимается ожидаемая реакция объекта – что можно ожидать от объекта в различных ситуациях и взаимодействиях [2].

Тогда высоким уровнем доверия $B_{E_i^T} \rightarrow 1$ может обладать объект, который не является источником деструктивных воздействий.

Низким уровнем доверия $B_{E_i^T} \rightarrow 0$ обладают объекты, только что появившиеся в системе, ранее неизвестные объекты, то есть объекты, которые являются или могут являться источником деструктивных воздействий.

То есть уровень доверия к объекту – это вероятность того, что объект не является источником деструктивного воздействия $Na_{E_i^T}$:

$$B_{E_i^T} = 1 - p(Na_{E_i^T}).$$

На конечном рабочем месте пользователя в некоторый момент времени могут функционировать только объекты определенных типов.

Возникает задача контроля уровня доверия к объектам системы и выполнения функций защиты при функционировании активных объектов с уровнем доверия ниже установленной нормы $B_{E_i^T} < \beta$. При этом необходимо выполнять защитные функции, связанные с конкретным объектом E_i^T , а не всей системой.

При оценке уровня доверия к объектам необходимо учитывать, что распространенность объекта повышает его уровень доверия. В то время как малораспространенные объекты имеют низкий уровень доверия. Если объект является или может являться источником (участником) деструктивных воздействий, то его уровень доверия понижается.

Пусть уровень доверия к объекту E_i^T складывается из голосов $\gamma = (\gamma^+ \cup \gamma^-)$, поданных за этот объект: γ^+ – положительный голос и γ^- – отрицательный голос. Если в системе обнаружено событие, указывающее на то, что объект попытался выполнить или выполнил некоторое деструктивное воздействие в обход установленных в системе правил, за него подается отрицательный голос γ^- . Если за некоторое время наблюдения объект не был источником деструктивного воздействия, за него подается положительный голос γ^+ .

С учетом особенностей для получения оценки уровня доверия к объекту можно использовать функцию следующего вида:

$$F(\varepsilon, \theta, \gamma) = \frac{\gamma^+ - (\gamma^-)^\theta}{\gamma + \frac{\varepsilon^2}{\gamma}}, \quad (1)$$

где γ^+ – количество положительных голосов, поданных за объект; γ^- – количество отрицательных голосов, поданных за объект; γ – общее количество голосов, поданных за объект; ε – коэффициент, определяющий степень значимости положительного голоса (коэффициент достаточности); θ – коэффициент, определяющий степень значимости отрицательного голоса (коэффициент критичности).

Эта функция имеет некоторые ограничения:

- так как значение доверия к объекту находится в интервале $B_{E_i^T} \in (0, 1)$, а область значений функции (1) находится в интервале $F(\varepsilon, \theta, \gamma) \in (-\infty, 1)$, то при $F(\varepsilon, \theta, \gamma) < 0$ нельзя говорить о доверии к объекту. Поэтому можно определить $\gamma = \Psi$, при котором $F(\varepsilon, \theta, \Psi) = 0$;
- величина Ω зависит от количества отрицательных голосов γ^- и коэффициента θ , определяющего степень значимости отрицательного голоса. Из равенства $F(\varepsilon, \theta, \Psi) = 0$ можно показать, что $\Omega = (\gamma^-)^\theta + \gamma^-$;
- можно определить значение $\gamma = \Psi$, при котором функция достигает середины уровня доверия $F(\varepsilon, \theta, \Psi) = 1/2$. Если $\gamma^- = 0$, то равенство $F(\varepsilon, \theta, \Psi) = 1/2$ достигается при $\Psi = \varepsilon$. Если $\gamma^- \neq 0$, то равенство $F(\varepsilon, \theta, \Psi) = 1/2$ достигается при $\Psi = \Omega + \sqrt{\Omega^2 + \varepsilon^2}$.

С учетом установленных ограничений значение оценки уровня доверия к объекту АИС вычисляется следующим образом:

$$B_{E_i^T} = \begin{cases} \frac{\gamma^+ - (\gamma^-)^\theta}{\gamma + \frac{\varepsilon^2}{\gamma}}, & \gamma > \Omega, \\ 0, & \gamma < \Omega. \end{cases} \quad (2)$$

Выражение (2) можно назвать (ε, θ) доверительной моделью объекта.

Коэффициент достаточности ε указывает на то, какое количество голосов должен получить объект, чтобы достигнуть уровня доверия 0,5, и позволяет контролировать скорость роста доверия к объекту.

При выборе коэффициента достаточности ε следует руководствоваться тем, что он должен иметь целое положительное значение, и тем, что при $\gamma^- = 0$ и $\gamma = \varepsilon$ уровень доверия $B_{E_i^T} = 1/2$.

На рис. 1 показано влияние коэффициента достаточности ε на значение оценки уровня доверия. Все голоса, поданные за объект, являются положительными.

Коэффициент критичности θ позволяет контролировать величину падения уровня доверия к объекту при появлении деструктивного воздействия.

Выбор коэффициента критичности θ не так однозначен, однако можно учитывать следующие рекомендации:

- при $\gamma = \varepsilon$, $\gamma^- = 1$ величина θ не имеет значения, а уровень доверия снизится на величину $1/\varepsilon$;
- при $\gamma = \varepsilon$, $\theta = 1$ уровень доверия снизится на величину $\gamma^- = \varepsilon$, тогда $B_{E_i^T} = 0$ при $\gamma^- = \varepsilon/2$;
- при $\gamma = \varepsilon$, $\theta \neq 1$ уровень доверия снизится на величину Ω/ε , но при $\Omega > \varepsilon$ уровень доверия $B_{E_i^T} = 0$;
- при $\gamma = \varepsilon$, $\theta = \log(\varepsilon - 2)$ и двух отрицательных голосах $\gamma^- = 2$ уровень доверия $B_{E_i^T} = 0$.

На рис. 2 показано влияние отрицательных голосов на значение оценки уровня доверия при параметрах $\varepsilon = 20$ и $\theta = 2$.

При оценке уровня доверия к объекту E_i^T возникает задача сбора голосов $\gamma = (\gamma^+ \cup \gamma^-)$, поданных за данный объект.

Пусть на компонентах АИС распределены агенты мониторинга.

Тогда голоса за объект могут подаваться агентами мониторинга следующим образом:

$$\gamma = \begin{cases} \gamma^+, & \text{если } Sa_{E_i^T}, \\ \gamma^-, & \text{если } Na_{E_i^T}. \end{cases}$$

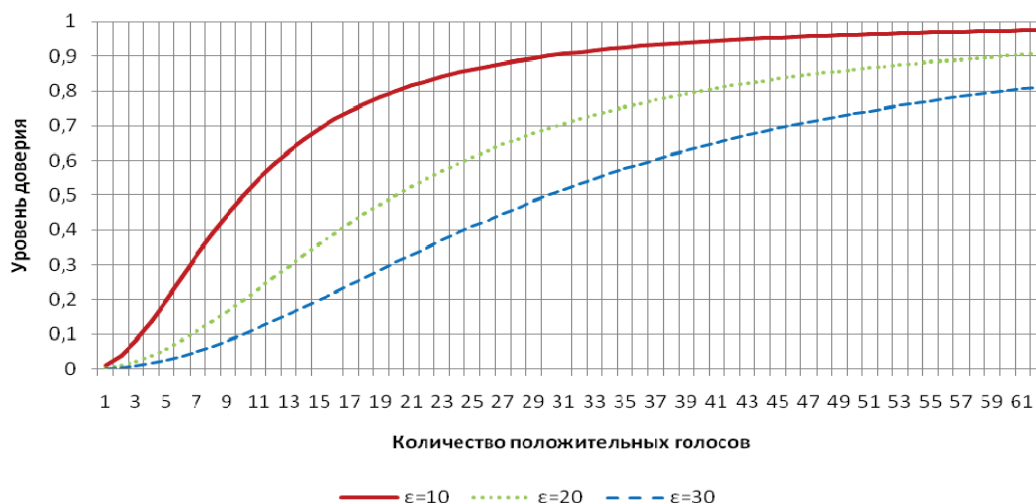


Рис. 1. Влияние параметра ε на уровень доверия к объекту

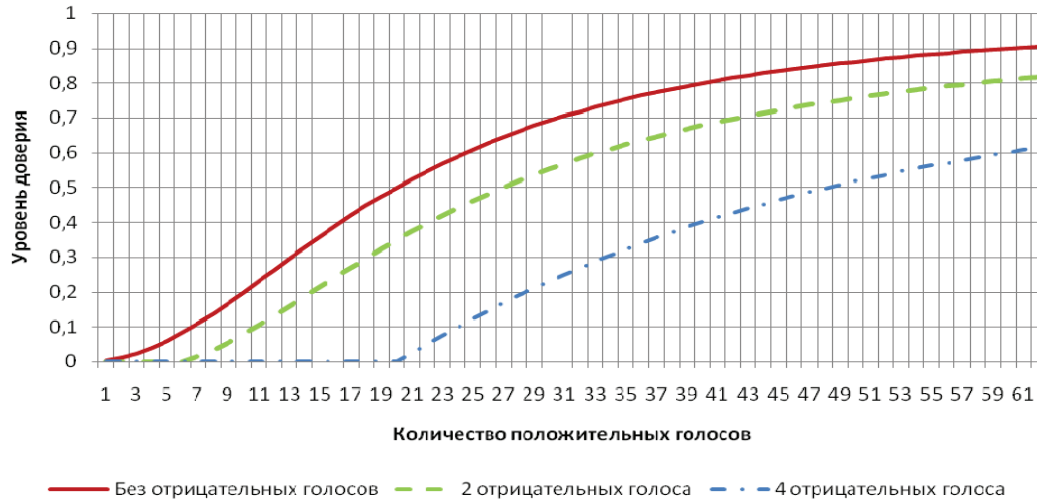


Рис. 2. Влияние отрицательных голосов на уровень доверия к объекту ($\gamma^-=0$, $\gamma^-=2$ и $\gamma^-=4$)

Если агент обнаруживает попытку объекта E_i^T совершить деструктивное воздействие $Na_{E_i^T}$, он подает отрицательный голос γ^- . Если объект E_i^T совершает нормальное воздействие $Sa_{E_i^T}$, агент мониторинга подает положительный голос γ^+ .

Если уровень доверия опускается ниже установленного критерия $B_{E_i^T} < \beta$, агент мониторинга выполняет блокировку подозрительного объекта или другое управляющее воздействие.

Тогда с учетом объектов, определенных выше, многоагентная система защиты может выглядеть следующим образом:

$$A^S = \{A_M, \langle A_{sc}, A_{ssw}, A_{usw}, A_{ac}, A_{net}, A_{db}, A_{dev} \rangle\},$$

где A_M — агент координатор мониторинга; A_{sc} — агент мониторинга системных компонент; A_{ssw} — агент мониторинга системного программного обеспечения; A_{usw} — агент мониторинга пользовательского программного обеспечения; A_{ac} — агент мониторинга доступа; A_{net} — агент мониторинга сетевых соединений; A_{db} — агент мониторинга баз данных; A_{dev} — агент мониторинга внешних устройств.

Таким образом, агенты координаторы, расположенные на конечных рабочих местах пользователей, выполняют мониторинг объектов на данном рабочем месте.

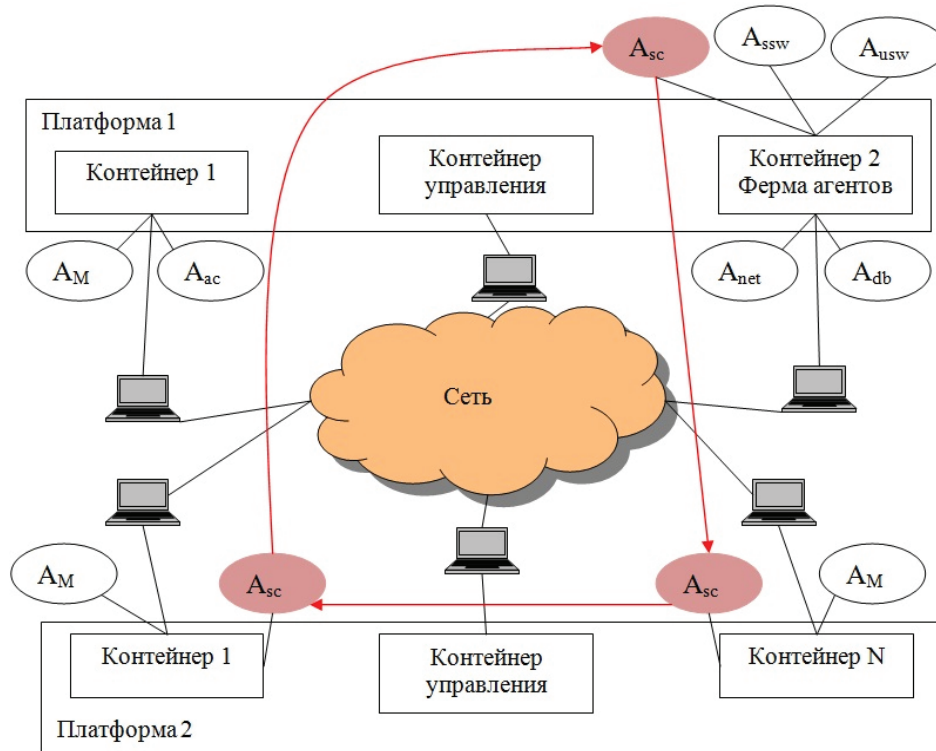


Рис. 3. Расположение агентов в системе и их миграция

При появлении объектов определенного типа агент координатор запрашивает специализированного агента мониторинга. Если этот агент обнаруживает объект с низким уровнем доверия, или возникает подозрение на выполнение деструктивных воздействий, агент мониторинга выполняет определенные функции безопасности.

При завершении использования объектов определенного типа работа специализированного агента мониторинга завершается.

Специализированные агенты мониторинга находятся в центральном хранилище — ферме агентов. Каждый агент публикует сведения о предоставляемых сервисах безопасности, и агенты координаторы запрашивают выполнение функций безопасности этими специализированными агентами. При запросе агент мониторинга мигрирует с фермы на конечное рабочее место, выполняет работу и возвращается обратно на ферму (рис. 3).

Таким образом, агент мониторинга выступает сервисом, который заказывает агент координатор.

На схеме также представлены блоки [3]:

- контейнер — агентная среда на конечном рабочем месте в автоматизированной информационной системе;
- платформа — группа контейнеров, объединенная одним блоком управления, реализующим агентную парадигму.

Описанный подход позволяет контролировать изменения среды функционирования АИС и появление новых компонентов, повышает эффективность функционирования конечных рабочих мест пользователей за счет использования средств защиты как сервисов, которые вызываются по необходимости. При этом все компоненты имеют некоторый уровень доверия, что позволяет гибко управлять защитой АИС и контролировать ее защищенность.

СПИСОК ЛИТЕРАТУРЫ

1. Ткаченко Н.И., Спирин Н.А. Применение сервис-ориентированной архитектуры при интеграции систем управления технологическими процессами // Известия Томского политехнического университета. — 2010. — Т. 317. — № 5. — С. 61–67.
2. Новиков Д.А. Математические модели формирования и функционирования команд. — М.: Изд-во физико-математической литературы, 2008. — 184 с.
3. Bellifemine F., Caire G. Developing Multi-Agent Systems with JADE. — Chichester: John Wiley & Sons Ltd., 2007. — 286 p.

Поступила 21.09.2012 г.

УДК 550.8.053

АНАЛИЗ ИНФОРМАЦИОННЫХ СВОЙСТВ ВЗАИМНЫХ ФАЗОВЫХ СПЕКТРОВ ОТРАЖЕННЫХ СЕЙСМИЧЕСКИХ ВОЛН

В.П. Иванченков, А.И. Кочегуров, М.А. Черкасова*

Томский политехнический университет

*ЗАО «Гринатом», г. Северск

E-mail: kai@cc.tpu.edu.ru

На основе принятой модели слоистых поглощающих сред рассмотрены свойства взаимных фазовых спектров сейсмических волн, отраженных от кровли и подошвы исследуемой толщи, определены основные предпосылки их применения для прогноза геологического разреза.

Ключевые слова:

Линейная система, модель слоистой поглощающей толщи, взаимный фазовый спектр сигналов, коэффициенты отражения и преломления волн.

Key words:

Linear system, model of layered absorbing column, mutual phase spectrum of signals, reflection and wave refraction factors.

При решении задач прогноза геологического разреза (ПГР), в том числе прогноза нефтегазоносности осадочных толщ, по данным сейсмических наблюдений наиболее широко используются в качестве диагностических признаков динамические характеристики отраженных волн, непосредственно связанные с их амплитудой и энергией [1, 2]. Информация о свойствах фазочастотных характеристик (ФЧХ) сейсмических волн до последнего

времени практически не использовалась. Между тем в фазу сейсмических сигналов, а точнее в сложный закон изменения их фазовых спектров, заложена информация, позволяющая в условиях априорной неопределенности надежно обнаруживать и разрешать сигналы на фоне интенсивных помех, производить оценку их кинематических параметров [3, 4]. Как показано в [5, 6], текущие фазовые спектры отраженных сейсмических волн могут

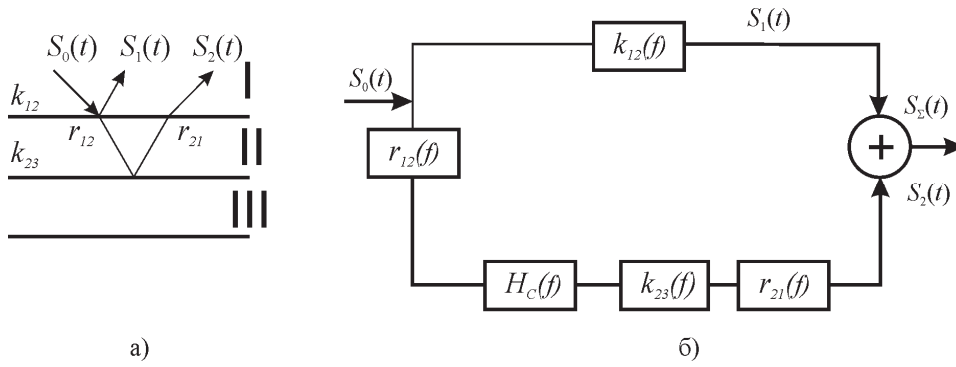


Рис. 1. Модель плоско-параллельной слоистой поглощающей толщи

служить важными характеристиками при прогнозе типов геологического разреза и оценке продуктивности осадочных толщ. В статье проводится анализ свойств взаимных фазовых спектров сейсмических волн, отраженных от кровли и подошвы исследуемых толщ, рассматриваются основные предпосылки их применения в задачах ПГР.

В рамках предложенного ранее системного подхода [6] предварительно остановимся на рассмотрении простой модели слоистой поглощающей толщи с горизонтальными границами раздела выделяемых слоев (рис 1, а). В лучевом приближении при нормальном падении волны $S_0(t)$ процесс формирования отраженных волн от кровли $S_1(t)$ и подошвы $S_2(t)$ линейно-неупругого поглощающего слоя II можно представить в виде эквивалентной системы, показанной на рис. 1, б.

При этом в отсутствии помех взаимный спектр (ВСП) для сигналов $S_1(t)$ и $S_2(t)$ можно определить:

$$Q_{12}(f) = [S_1^*(f) \cdot S_2(f)] = |Q_{12}(f)| e^{j\phi_{12}(f)},$$

где $\phi_{12}(f)$ – взаимный фазовый спектр сигналов $S_1(t)$ и $S_2(t)$. $S_1(t) = F\{S_0(t)\}$, $S_2(t) = F\{S_2(t)\}$ – спектры волн, отраженных от кровли и подошвы II-го слоя, F – оператор прямого преобразования Фурье. В соответствии с рис. 1, б, спектры $S_1(f)$ и $S_2(f)$ запишутся:

$$\begin{aligned} S_1(f) &= S_0(f) k_{12}(f) = |S_1(f)| e^{j\phi_{12}(f)}; \\ S_2(f) &= S_0(f) r_{12}(f) H(f) k_{23}(f) r_{21}(f) = \\ &= |S_2(f)| e^{j\phi_{22}(f)}, \end{aligned}$$

где $S_0(t) = F\{S_0(t)\}$, $k(f)$, $r(f)$ – коэффициенты отражения и преломления для границ раздела соответствующих слоев, $H(f)$ – частотная характеристика системы, определяющая свойства поглощающего слоя II.

Тогда взаимный спектр $Q_{12}(f)$ можно представить:

$$\begin{aligned} Q_{12}(f) &= |S_0(f)|^2 k_{12}^*(f) k_{23}(f) H(f) r_{12}(f) r_{21}(f); \\ |Q_{12}(f)| &= |S_0(f)|^2 \cdot |k_{12}(f)| \cdot |k_{23}(f)| \times \\ &\times |H(f)| \cdot |r_{12}(f)| \cdot |r_{21}(f)|; \\ \phi_{12}(f) &= \\ &= \phi_{k_{23}}(f) - \phi_{k_{12}}(f) + \phi_H(f) + \phi_{r_{12}}(f) + \phi_{r_{21}}(f). \end{aligned} \quad (1)$$

где $\phi_k(f)$, $\phi_r(f)$ – аргументы комплексных коэффициентов отражения и преломления, $\phi_H(f)$ – фазочастотная характеристика частной системы, определяющей распространение волны в поглощающем слое II в двух направлениях.

Как видно из (1), взаимный фазовый спектр (ВФС) волн, отраженных от кровли и подошвы II-го слоя, непосредственно зависит от разности аргументов коэффициентов отражения и суммы аргументов коэффициентов преломления границ раздела слоев. ВФС также определяется набегом фазочастотной характеристикой $\phi_H(f)$, обусловленной прохождением волны в поглощающем слое.

Важно отметить, что ВФС волн, отраженных от кровли и подошвы исследуемой толщи, не зависит от фазового спектра падающей волны $S_0(t)$, что является одним из положительных факторов, который позволяет повысить надежность дальнейшего прогноза свойств изучаемого геологического разреза.

Исходя из рис. 1, можно также представить частотную характеристику рассматриваемой эквивалентной системы в следующем виде:

$$\begin{aligned} H_{\Sigma}(f) &= r_{12}(f) H(f) k_{23}(f) r_{21}(f) + k_{12}(f) = \\ &= |H_{\Sigma}(f)| e^{j\phi_{\Sigma}(f)}. \end{aligned}$$

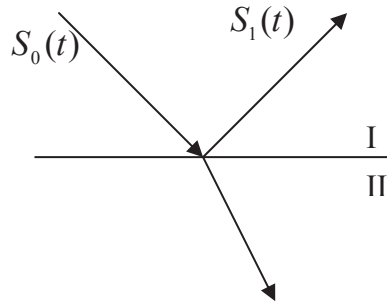


Рис. 2. Два изотропных полупространства, разделенных плоской границей

Согласно [6, 7], для двух изотропных полупространств, разделенных плоской границей, аргументы коэффициентов отражения и преломления запишутся как (рис. 2):

$$\phi_k = \arctg \frac{4\pi\gamma_{12}(f)(\delta_2(f) - \delta_1(f))}{4\pi^2(1 - \gamma_{12}^2(f)) + (\delta_1^2(f) - \delta_2^2(f))\gamma_{12}^2(f)}; \quad (2)$$

$$\phi_r = \arctg \frac{2\pi(\delta_1(f) - \delta_2(f))}{4\pi^2 + \delta_1(f)\delta_2(f) + \gamma_{12}(f)(4\pi^2 + \delta_2^2(f))}, \quad (3)$$

где $\gamma_i(f) = \rho_i V_i(f)$, ($i=1,2$); $\gamma_{12}(f) = \frac{\rho_1 V_1(f)}{\rho_2 V_2(f)}$ – отношение акустических жесткостей контактирующих сред, а $V(f)$ и $\delta(f)$ соответственно определяют скорость распространения продольных волн и декремент поглощения в среде на частоте f . При этом

$$\delta(f) = \frac{\alpha(f)V(f)}{f},$$

где $\alpha(f)$ – коэффициент поглощения.

По результатам различных исследований часто принимается, что коэффициент поглощения $\alpha(f)$ связан с частотой линейной зависимостью [1]:

$$\alpha(f) = \beta f,$$

где β – параметр поглощения.

Исходя из (1), взаимный фазовый спектр отраженных волн зависит от ФЧХ системы, определяющей свойства поглощающего слоя II. Принимая для поглощающего слоя минимально-фазовую модель среды и учитывая ранее сделанное допущение о линейной зависимости коэффициента поглощения от частоты f , выражения для частотной характеристики $H(f)$ можно представить как [6]:

$$H(f) = |H(f)|e^{-j\varphi_H(f)},$$

$$|H(f)| = e^{-2\alpha(f)h},$$

$$\varphi_H(f) = 4\pi fh \left[\frac{1}{V(f_0)} - \frac{\beta}{\pi^2} \ln \frac{f}{f_0} \right], \quad (4)$$

где $V(f_0)$ – фазовая скорость распространения волн на некоторой базовой частоте f_0 , h – мощность слоя.

Из соотношений (1–4) видно, что взаимный фазовый спектр волн, отраженных от кровли и подошвы анализируемой толщи, непосредственно зависит от акустических и поглощающих свойств контактирующих толщ. На рис. 3 в качестве примера приведены графики аргументов коэффициентов отражения и преломления волн, определенных при различных значениях отношений декрементов поглощения $\delta_{12}(f)$ и акустических жесткостей $\gamma_{12}(f)$ контактирующих сред. Исследования показали, что аргумент коэффициента отражения во всех случаях, когда декременты поглощения контактирующих сред не равны между собой, зависит от частоты. Влияние поглощения на аргумент коэффициента отражения тем значимее, чем меньше дифференциация акустических жесткостей контактирующих сред. Так, при $\gamma_{12}(f_0)=0,9$ и $\delta_{12}(f_0)=0,05–1$

изменение фазы по частоте $\frac{f}{f_0} = 2$ составляет 1,5 рад., а в случае $\gamma_{12}(f_0)=0,62$ при тех же самых условиях значение уменьшается до 1 рад.

При этом следует отметить, что в этом случае достаточно большие фазовые сдвиги ($\approx 0,25$ рад.)

наблюдаются даже при очень малом изменении (порядка 0,05) декремента поглощения. Это свидетельствует о высокой чувствительности $\varphi_k(f)$ к небольшим изменениям поглощающих свойств отложений. Фазовые набегі за счет преломления на границах раздела $\varphi_k(f)$ в поглощающих средах существенно меньше, чем за счет отражения. С ростом частоты при любых сочетаниях $\gamma_{12}(f_0)$ и $\delta_{12}(f_0)$ аргумент коэффициента преломления уменьшается и имеет отрицательную величину.

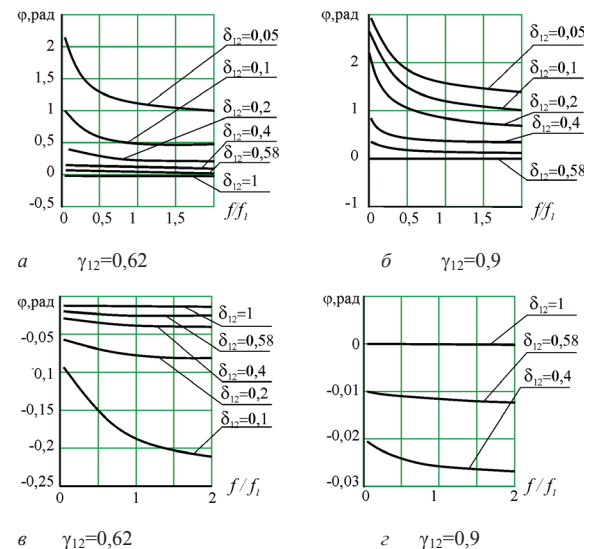


Рис. 3. Зависимость аргументов коэффициентов отражения (а, б) и преломления (в, г) от частоты f/f_0

По аналогии с изложенным выше может быть построена модель поглощающей толщи, состоящей из произвольного числа слоев и определена частотная характеристика $H_\Sigma(f)$ ее эквивалентной системы [6]. На основе определенной $H_\Sigma(f)$ может быть рассчитана синтетическая сейсмограмма для любой точки поверхности наблюдения:

$$S_\Sigma(t) = F^{-1}\{S_0(f)H_\Sigma(f)\},$$

где F^{-1} – оператор обратного преобразования Фурье.

Совокупность рассчитанных сейсмограмм (сейсмограмма) для различных точек наблюдения для принятой модели слоистой поглощающей среды позволяет провести исследования влияния петрофизических параметров анализируемой толщи на ВФС отраженных волн.

Предварительно исследования свойств ВФС проводились для 4-х моделей слоистых поглощающих сред. Общая структура принятых моделей приведена на рис. 4.

Аргиллиты
Продуктивная толща
Аргиллиты

Рис. 4. Структура моделей слоистых поглощающих сред

Каждая модель включает три толщи, где вторая толща – песчаный коллектор (продуктивная тол-

ша), а толщи I и III – слои аргиллитов. В первой модели песчаный слой предполагается насыщенным газом, во второй – нефтью, в третьей – водой. В четвертой модели имитировалось наличие непроницаемого карбонизированного песчаника.

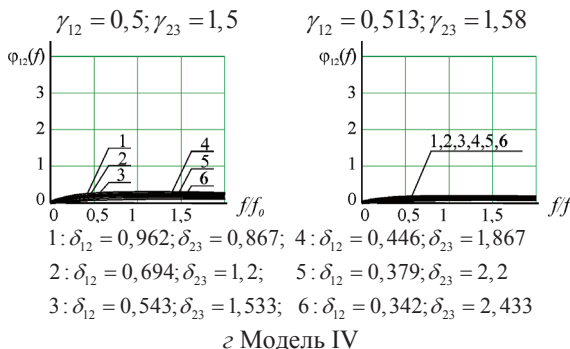
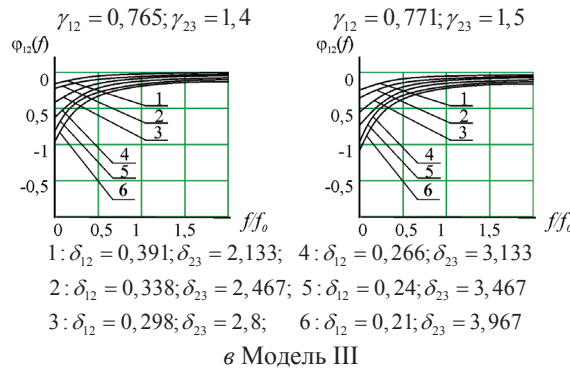
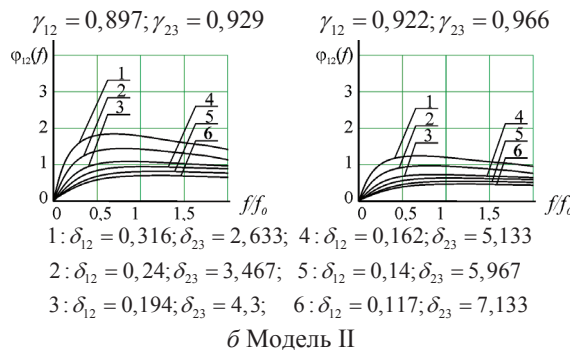
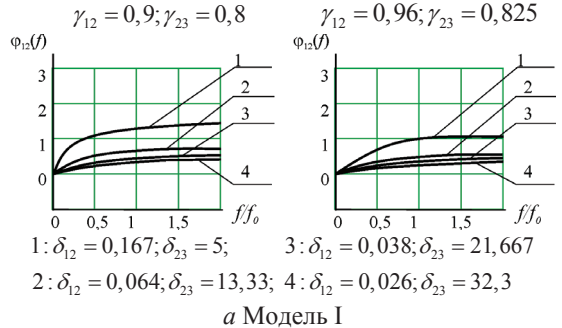


Рис. 5. Взаимные фазовые спектры волн, отраженных от кровли и подошвы II слоя: а) газонасыщенного песчаника, б) нефтенасыщенного песчаника, в) водонасыщенного песчаника, г) карбонизированного песчаника

При исследованиях скорости распространения продольных волн, плотности пород и параметры поглощения во всех слоях выбирались исходя из обобщенных данных, характерных для верхнеюрской продуктивной толщи Крапивинской площади Томского Приобья. Мощность (толщина) второго слоя задавалась 15 м. На рис. 5 показаны графики взаимных фазовых спектров волн, отраженных от кровли и подошвы II-го слоя, рассчитанных для различных отношений декрементов поглощения $\delta_{12}(f)$, $\delta_{23}(f)$ и акустических жесткостей $\gamma_{12}(f)$, $\gamma_{23}(f)$ контактирующих сред.

Анализ графиков показывает, что насыщение породы-коллектора газом или нефтью непосредственно приводит к значимому изменению взаимных фазовых спектров (модель I и II на рис. 5). При этом наибольшие отклонения взаимного фазового спектра при определенных сочетаниях отношений акустических жесткостей и декрементов поглощения контактирующих сред могут достигать 1,8 рад. В случае насыщения песчаного коллектора водой (модель III на рис. 5) в диапазоне изменения отношения частот f/f_0 от 0,5 до 2 ВФС волн слабо зависят от частоты и их значения сосредоточены в области от 0...–1 рад. При отсутствии коллектора (модель IV на рис. 5) взаимные фазовые спектры отраженных волн практически не зависят от частоты и их значения при различных сочетаниях петрофизических параметров контактирующих слоев близки к 0. Из приведенных зависимостей также видно, что графики ВФС в случае насыщения песчаного коллектора газом или нефтью во всем диапазоне анализируемых частот обладают большей кривизной и имеют больший интервал изменения значений, чем в случае отсутствия коллектора или его насыщения водой. Это позволяет при решении задач прогноза нефтегазоносности осадочных толщ ввести в качестве диагностических признаков параметры ВФС, учитывающие отклонение их значений от средних. В частности, для этих целей может быть использован центральный момент 2-го порядка ВФС:

$$\sigma_{\phi}^2 = \frac{\sum_{i=1}^n (\phi_{12}(f_i) - \bar{\phi}_{12}(f))^2}{n-1}, \quad (5)$$

где $\bar{\phi}_{12}(f) = \frac{1}{n} \sum_{i=1}^n \phi_{12}(f_i)$ – среднее значение взаимного фазового спектра; n – число гармоник, учитываемых при определении взаимного фазового спектра отраженных волн.

Наряду с взаимным фазовым спектром волн информацию о дисперсионности сред и поглощающих свойствах исследуемых толщ также несут взаимные фазовые и групповые задержки:

$$\tau_{\phi}(f) = \frac{\phi_{12}(f)}{2\pi f}, \quad t_{\text{гп}}(f) = -\frac{d\phi_{12}(f)}{2\pi df}.$$

Степень их отклонения может быть соответственно определена:

Аргиллиты (Баженовская пачка)			Аргиллиты (Баженовская пачка)		
песчаник			песчаник		
Карбонизированный	Газонасыщенный	Карбонизированный	Водонасыщенный	Газонасыщенный	Водонасыщенный
Аргиллиты			Аргиллиты		
а Модель I			б Модель III		

Рис. 6. Структура моделей слоистых поглощающих сред при изменении их свойств по латерали

$$\sigma_{\tau}^2 = \frac{\sum_{i=1}^n (\tau_{\phi}(f_i) - \bar{\tau}_{\phi})^2}{n-1}. \quad (6)$$

$$\sigma_t^2 = \frac{\sum_{i=1}^n (t_{\text{тр}}(f_i) - \bar{t}_{\text{тр}})^2}{n-1}. \quad (7)$$

С целью исследования информативности параметров (5–7) дополнительно были рассмотрены 4 модели слоистых сред, состоящие из трех толщ. Покрывающая и подстилающие толщи являются, как и ранее, слоями аргиллитов, а второй слой задавался как продуктивная толща, свойства которой изменялись по латерали (рис 6, а, б).

Рассматривались следующие изменения по латерали структуры II-го слоя:

- 1) Карбонизированный песчаник–газонасыщенный песчаник–карбонизированный песчаник (рис. 6, а);
- 2) Карбонизированный песчаник–нефтенасыщенный песчаник–карбонизированный песчаник;
- 3) Водонасыщенный песчаник–газонасыщенный песчаник–водонасыщенный песчаник (рис. 6, б);
- 4) Водонасыщенный песчаник–нефтенасыщенный песчаник–водонасыщенный песчаник.

Петрофизические параметры слоев (скорости распространения волн, плотности пород, параметры поглощения) были выбраны из тех же условий, что и для рассмотренных ранее моделей сред. Мощность II-го слоя была задана равной 50 м. Во всех случаях при исследовании варьировались петрофизические параметры II-го слоя, при неизменных параметрах покрывающей и подстилающей толщ. Для каждой из принятых моделей по разработанной программе осуществлялся расчет и построение наблюдаемых волновых полей (рис. 7), на которых автоматически прослеживались волны, отраженные от кровли и подошвы II-го слоя.

Для них определялся их взаимный фазовый спектр (рис. 8) и оценивались прогнозные параметры (5–7).

На рис. 9 в качестве примера приведены графики изменений по латерали центральных моментов 2-го порядка фазовой и групповой задержек для моделей сред, показанных на рис. 6.

Из рис. 9 видно, что на участках насыщения песчаного коллектора газом наблюдаются аномальные изменения указанных прогнозных параметров. При этом, как следовало ожидать, при структуре II-го слоя типа «вода–газ–вода» (рис. 6, б) относительное изменение параметров на различных участках профиля оказалось меньшим, чем в слу-

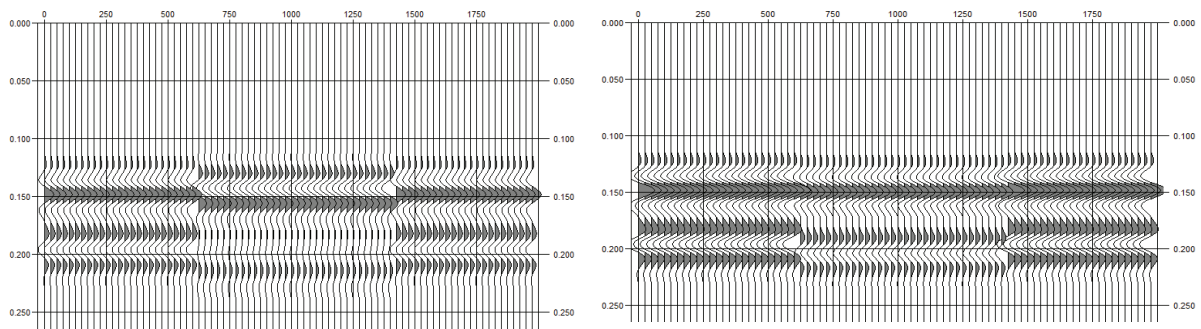


Рис. 7. Фрагменты волновых полей

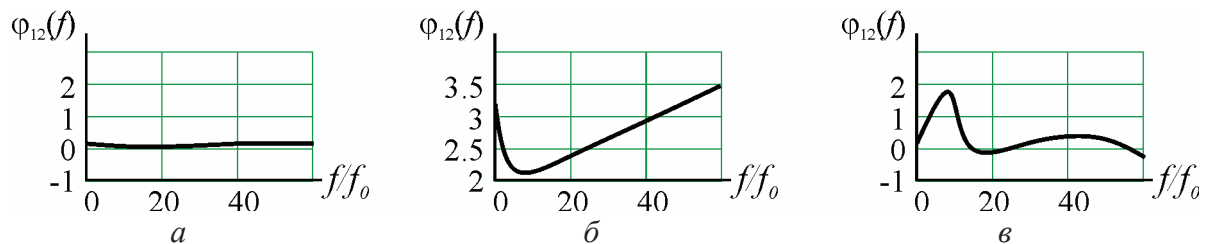


Рис. 8. Взаимные фазовые спектры волн, отраженных от кровли и подошвы II-го слоя: а) карбонизированный; б) газонасыщенный; в) водонасыщенный песчаник

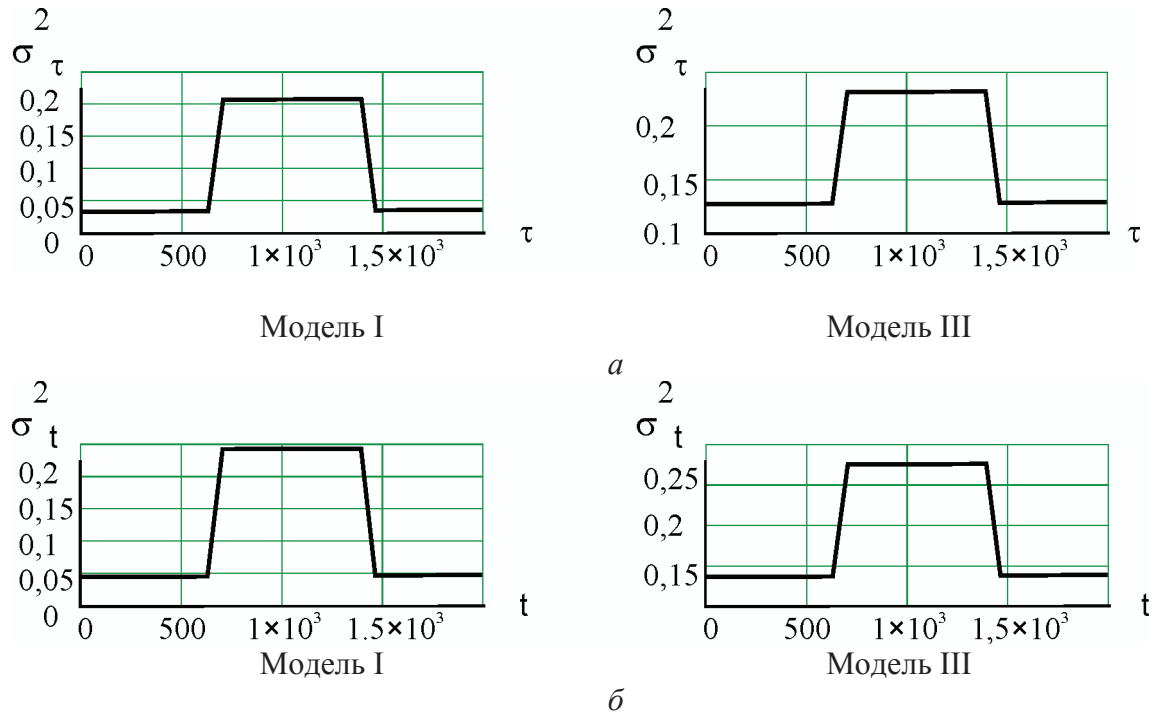


Рис. 9. Центральный момент 2-го порядка: а) взаимной фазовой задержки; б) групповой задержки

чае, когда структура II-го слоя по латерали была задана в виде, приведенном на рис. 6, а. Аналогичные закономерности в изменении указанных прогнозных параметров также наблюдались при насыщении породы коллектора нефтью (модели II и IV). Результаты проведенных исследований показали, что наиболее информативными параметрами являются центральные моменты взаимной фазовой и групповой задержек. Центральный момент взаимного фазового спектра при определенных сочетаниях петрофизических параметров пород продуктивного слоя имеет незначительные аномалии при задании структуры модели II-го слоя по латерали в виде «вода—нефть—вода».

С целью дальнейшего исследования информативности выбранных прогнозных параметров была проведена выборочная обработка сейсмических данных метода общей глубинной точки (ОГТ), полученных на Крапивинском месторождении Томского Приобья.

При этом осуществлялась обработка временных разрезов ОГТ для нескольких сейсмических профилей, на которых расположены разведочные скважины. Основным продуктивным объектом Крапивинского месторождения являются отложения верхней Юры. В этой связи рассматривалась толща, ограниченная подошвой Баженовской свиты (Bg) и пластом Ю₁³⁻⁴. На рис. 10, а, 12,

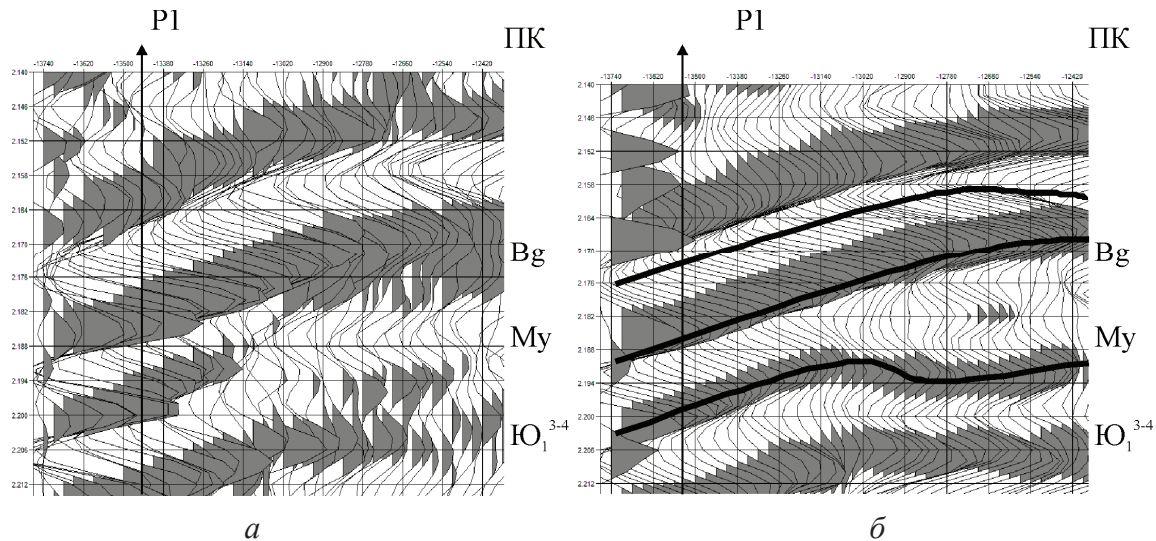


Рис. 10. а) фрагмент разреза ОГТ, профиль ПР 1; б) Фрагмент объектно-ориентированного фазового разреза ПР 1

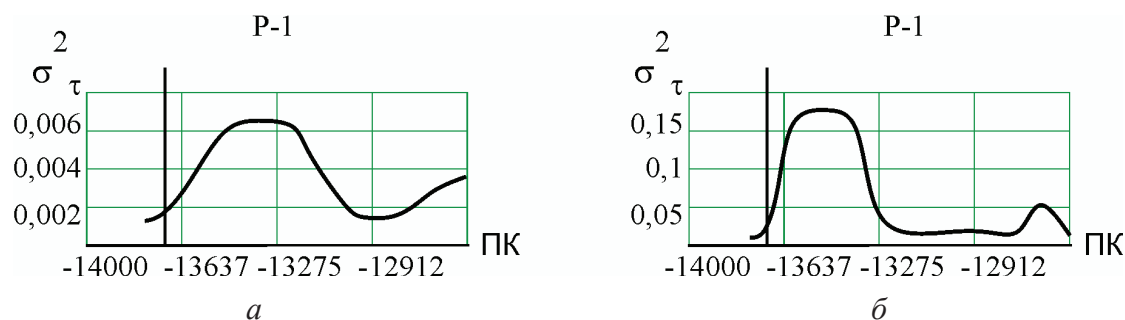


Рис. 11. Изменение прогнозных параметров вдоль профиля наблюдения ПР 1: а) центрального момента 2-го порядка взаимной фазовой задержки; б) центрального момента 2-го порядка взаимной групповой задержки

а приведены фрагменты временных разрезов ОГТ, полученных после стандартной обработки, для двух сейсмических профилей (условное обозначение ПР 1 и ПР 2), на пересечении которых расположена скважина Р-1 (условное обозначение) с дебитом нефти 130 м³/сут. Следует отметить, что на временных разрезах ОГТ не всегда удастся проследить оси синфазности отраженных волн, отождествляемых по стратиграфической привязке с продуктивным пластом Ю₁³⁻⁴ (рис. 10, а), что не позволяет провести вычисление оценок взаимных фазовых спектров целевых волн, привязанных к подошве Вg и пласту Ю₁³⁻⁴. В этой связи на основе ранее предположенных методов фазочастотного прослеживания (ФЧП) [4] была прове-

дена детальная обработка временных разрезов ОГТ и построены объектно-ориентированные разрезы, на которых на всем интервале выделяются целевые волны (рис. 10, б, 12, б). На основе данных разрезов была проведена оценка временного положения целевых волн и определение их взаимных фазовых спектров и указанных выше прогнозных параметров.

На рис. 11, 13 приведены графики принятых прогнозных параметров, определённых для указанных фрагментов разрезов ПР 1 и ПР 2. Из них видно, что на пикетах наблюдения обоих профилей, где расположена скважина Р-1, прогнозные параметры имеют ярко выраженные аномалии, свидетельствующие о наличии нефтяной залежи.

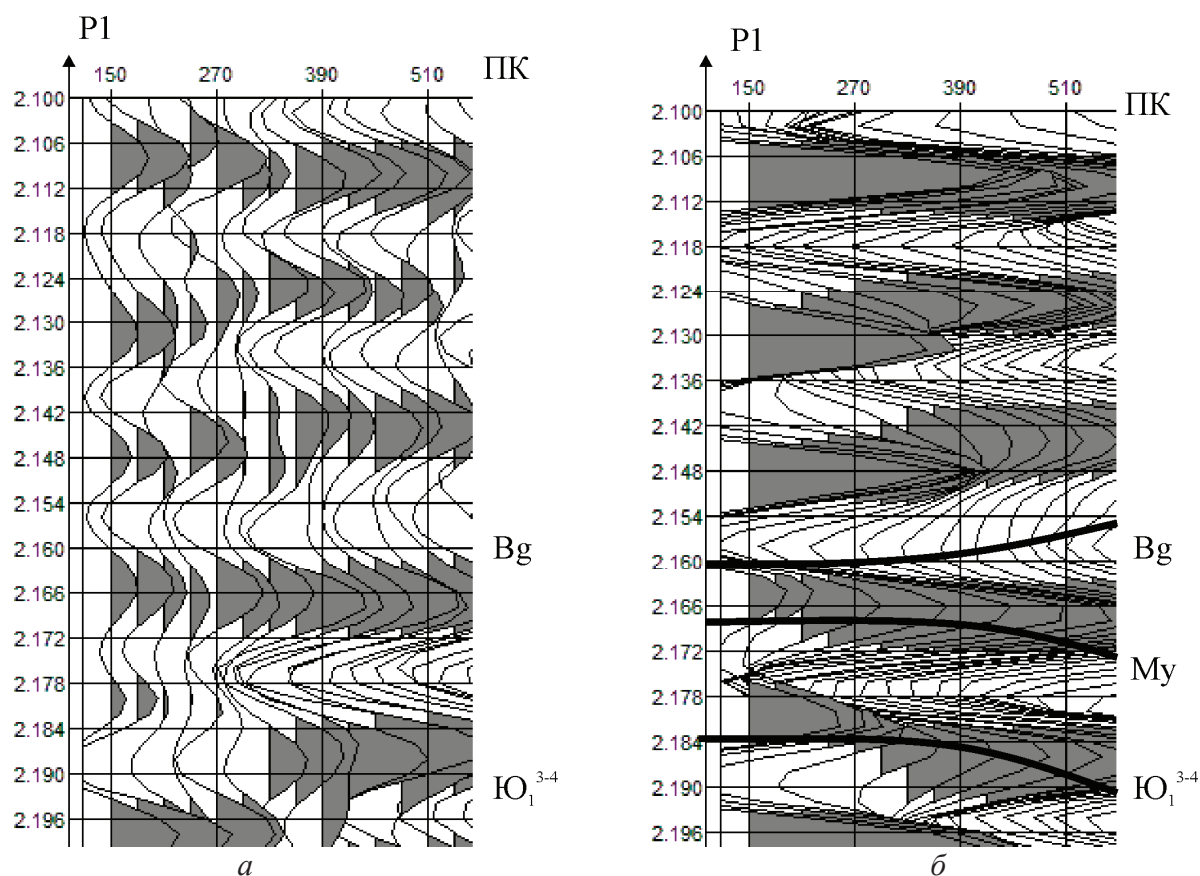


Рис. 12. а) фрагмент разреза ОГТ, профиль ПР 2; б) фрагмент объектно-ориентированного фазового разреза ПР 2

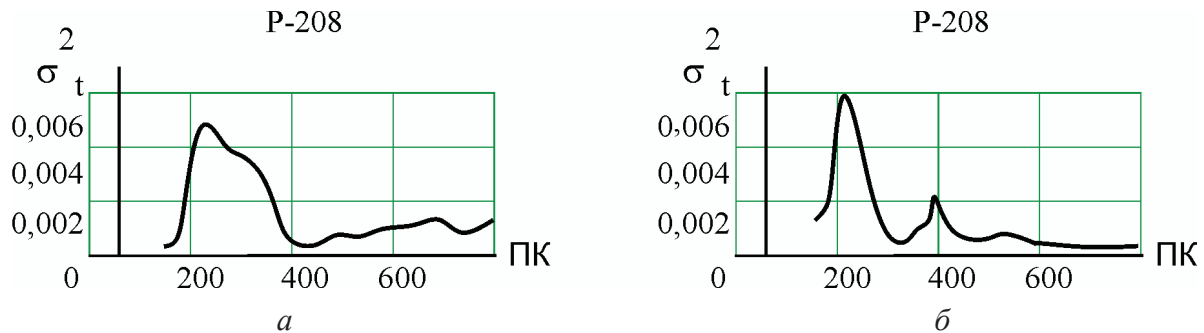


Рис. 13. Изменение прогнозных параметров вдоль профиля наблюдения ПР 2: а) центрального момента 2-го порядка взаимной фазовой задержки; б) центрального момента 2-го порядка взаимной групповой задержки

Выборочная обработка материалов ОГТ для ряда других сейсмических профилей, на которых расположены продуктивные скважины, в целом подтвердила высокую информативность выделенных прогнозных параметров, определяемых на основе взаимных спектров волн, отраженных от кровли и подошвы анализируемых толщ. Таким образом,

проведенные исследования на моделях слоистых поглощающих сред и результаты обработки реальных сейсмических материалов указывают на возможность использования взаимных фазовых спектров сейсмических волн в качестве новых важных диагностических признаков при прогнозе нефтегазоносности осадочных толщ.

СПИСОК ЛИТЕРАТУРЫ

1. Гурвич И.И., Боганик Г.Н. Сейсморазведка. — Тверь: Изд-во АИС, 2006. — 774 с.
2. Птецов С.Н. Анализ волновых полей для прогнозирования геологического разреза. — М.: Недра, 1989. — 135 с.
3. Иванченков В.П., Кочегуров А.И. Определение временного положения сейсмических сигналов по оценкам их фазочастотных характеристик // Геология и геофизика. — 1988. — № 9. — С. 77–83.
4. Иванченков В.П., Кочегуров А.И., Орлов О.В. Исследование разрешающей способности методов фазочастотного прослеживания сейсмических сигналов // Известия Томского политехнического университета. — 2012. — Т. 320. — № 5. — С. 80–85.
5. Иванченков В.П., Козлов А.А. Классификация типов геологического разреза методами кластерного анализа по данным сейсмических наблюдений // Известия Томского политехнического университета. — 2011. — Т. 318. — № 1. — С. 54–59.
6. Иванченков В.П., Кочегуров А.И., Орлов О.В. Фазочастотные характеристики сейсмических волн и основные предпосылки их применения для решения задач прогноза геологического разреза // Известия Томского политехнического университета. — 2008. — Т. 313. — № 5. — С. 132–138.
7. Авербух А.Г., Трапезникова Н.А. Отражения и преломления плоских волн при нормальном падении на границы // Физика земли. — 1972. — № 9. — С. 74–83.

Поступила 10.04.2013 г.

УДК 004.056.5, 004.89

СИСТЕМА АНАЛИЗА КАТАСТРОФУСТОЙЧИВОСТИ

В.С. Аткина

Волгоградский государственный университет

E-mail: atkina.vladlena@yandex.ru

Описана проблема создания катастрофустойчивых информационных систем и их значение при обеспечении информационной безопасности. Предположено, что при создании и сопровождении катастрофустойчивых информационных систем необходимо проводить анализ текущих показателей катастрофустойчивости системы и оценку эффективности катастрофустойчивых решений. Предложена и формально описана система анализа катастрофустойчивости, построенная на базе искусственной иммунной системы. На примере результатов проведенных экспериментальных исследований показана возможность применения предложенного решения в качестве инструментального средства поддержки принятия решений.

Ключевые слова:

Информационная система, катастрофустойчивость, искусственные иммунные системы, катастрофустойчивые решения, дестабилизирующие факторы, информационная безопасность.

Key words:

Information system, disaster recovery, artificial immune systems, disaster recovery solutions, destructive factors, information security.

На сегодняшний день деятельность любой организации (предприятия, учреждения) вне зависимости от принадлежности к государственной или коммерческой сфере тесно связана с использованием информации в различных ее видах, и, как правило, значительный процент информации представлен в электронном виде. Следовательно, процесс функционирования типовой организации состоит в постоянной обработке больших объемов информации, их анализе, принятии решений и управленческой деятельности. Для автоматизации данных процессов организацией используются информационные системы (ИС) различного типа. А это означает, что для успешного существования и развития организации необходимо обеспечивать безопасность ИС и циркулирующей в ней информации.

При этом одним из важнейших компонентов безопасности информации будет обеспечение доступности данных и надежности их обработки, что является особенно актуальным в настоящее время в связи с постоянным возникновением чрезвычайных ситуаций различного рода в самых различных областях человеческой деятельности. В соответствии с [1] к подобным чрезвычайным ситуациям можно отнести стихийные бедствия, имеющие самые различные последствия для всех сфер жизни общества (например, события в Японии 2011 г.), террористическую угрозу (особенно после известных событий 11 сентября 2001 г. в США), техногенные катастрофы. Анализ показывает, что все перечисленные угрозы, как правило, имеют комбинированный характер и приводят к возникновению и развитию зачастую неконтролируемого потока негативных последствий. В этих условиях обеспечение непрерывности бизнес-процессов, сохранности и доступности информации, а также повышение катастрофустойчивости соответствующих производственных и ИС, входящих в состав современных организаций (в том числе и виртуальных), является одним из важнейших стратегических направлений развития экономики.

Создание катастрофустойчивых информационных систем (КАИС) обеспечивается за счет внедрения специальных катастрофустойчивых решений, представляющих собой комплекс организационно-технических мероприятий и планов поведения персонала в случае наступления катастрофы. Основной целью применения данных решений является обеспечение непрерывности функционирования ИС в условиях деструктивного воздействия, сокращение объемов потери данных и минимизация времени восстановления работоспособности ИС. При этом каждое из решений отличается своей стоимостью, временем внедрения и эффективностью. Поэтому для того чтобы определить, какое из решений или их сочетание будет наиболее выгодно организации-владельцу ИС по соотношению затрат на внедрение, стоимости информационных ресурсов и требований к уровню катастрофустойчивости, необходимо провести анализ текущего состояния ИС; оценку катастрофустойчивых решений и выбрать наиболее подходящие с учетом специфики каждой конкретной организации [2, 3].

Для решения поставленных задач автором предлагается система анализа катастрофустойчивости, построенная на базе искусственной иммунной системы. Формально данную систему можно описать следующим кортежем:

$$M_{DRIS} = (\{M_{IS}\}, \{R_{DRIS}\}, \{DF\}, \{DRS\}, IMS),$$

где $\{M_{IS}\}$ – множество, описывающее КАИС, ее структуру, характеристики и показатели катастрофустойчивости; $\{R_{DRIS}\}$ – множество требований организации-владельца к степени катастрофустойчивости системы и потенциально возможным затратам; $\{DF\}$ – множество дестабилизирующих факторов существенной среды; $\{DRS\}$ – множество катастрофустойчивых решений; IMS – функция управления процессом принятия решений, построенная на базе модели иммунной системы.

Множество, описывающее модель анализируемой КАИС в соответствии с [4], представляет собой

множество $\{M_{IS}\}=\{\{S\},\{C\},\{DRS\}\}$, где $\{S\}$ – множество состояний системы в различные периоды функционирования. Каждое состояние $S_i \in \{S\}$ описывается следующим кортежем $S_i = \{G_{is}, L, T_R, D_{class}, N_{Dlost}, Z\}$, где G_{is} – граф, описывающий физическую структуру КАИС; L – уровень катастрофоустойчивости, T_R – время восстановления функционирования; D_{class} – класс доступности (готовности); N_{Dlost} – объем потерянных данных; Z – живучесть; C – стоимость системы.

$\{DRS\}$ – множество катастрофоустойчивых решений, уже внедренных (имеющихся) в КАИС на момент исследования, при этом следует учитывать, что множество всех доступных в рамках модели катастрофоустойчивых решений $\{DRS\}$ включает подмножество реализованных в КАИС решений $\{DRS\}$, т. е. $\{DRS\} \subset \{DRS\}$.

Множество требований, предъявляемых организацией-владельцем КАИС к уровню ее катастрофоустойчивости, представляет собой «эталонные» значения множества показателей оценки катастрофоустойчивости, которые являются входными данными и задаются на этапе сбора информации об исследуемой КАИС. Для описания множества требований организации владельцем используется следующий кортеж:

$$\{R_{DRIS}\} = \{L^v, T_R^v, D_{class}^v, N_{Dlost}^v, Z^v, T_{DRS}^v, C_{DRS}^v, R_{isko}^v\},$$

где $L^v, T_R^v, D_{class}^v, N_{Dlost}^v, Z^v$ – показатели катастрофоустойчивости; T_{DRS}^v – приемлемое время развёртывания катастрофоустойчивых решений; C_{DRS}^v – максимальные финансовые затраты на развёртывание катастрофоустойчивых решений; R_{isko}^v – предельно допустимый уровень риска; v – степень важности выполнения каждого требования, описывается множеством базовых значений $v = \{1, 2, 3\} = \{\text{низкая важность, средняя важность, высокая важность}\}$.

Множество $\{DF\}$ представляет собой множество дестабилизирующих факторов, аварий и катастроф, порождаемых существенной средой. Каждый элемент из множества $DF_i \in \{DF\}$ описывается вектором $DF_i = (P, U)$, где P – вероятность наступления дестабилизирующего фактора, U – потенциально возможный ущерб.

В соответствии с [5–7] функция принятия решений IMS , реализованная с использованием технологии функционирования искусственных им-

мунных систем, может быть представлена следующим кортежем:

$IMS = (S, \{DF\}, \{DRS\}, \{R_{DRIS}\}, Sp, \{DET\}, \{ANG\}, \{ANT\})$, где $Sp = (L, T_R, D_{class}, N_{Dlost}, Z)$ – вектор текущих значений показателей катастрофоустойчивости исследуемой ИС, выделяется модулем «макрофагом» разработанной ИМС из вектора текущего состояния системы $S_i \in \{S\}$.

$DET = \{det_1, det_2, \dots, det_m\}$ – набор множества детекторов двух типов мощностью m , отвечающих за решение задач классификации состояний КАИС и выявление потенциально опасных дестабилизирующих факторов, которые участвуют в формировании «иммунной памяти», путем добавления в базу данных информации о результатах проведенной классификации и принятых детекторами решениях. Детекторы первого типа $DET^f \in DET$ представляют собой шаблоны, описывающие «нормальные» состояния системы, реагируют на возможные отклонения значений вектора Sp от «нормальных» значений, описанных в шаблонах, и сообщают об «аномальном» состоянии системы, что указывает на необходимость проведения корректировки текущей катастрофоустойчивости системы путем подбора катастрофоустойчивых решений. Детекторы второго типа $DET^{df} \in DET$ отвечают за классификацию дестабилизирующих факторов и выявление наиболее критичных из них.

ANG – множество антигенов;

ANT – множество антител.

На вход функции IMS подаются следующие данные (рис. 1): вектор $S_i \in \{S\}$, характеризующий текущее состояние исследуемой КАИС; множество актуальных для исследуемой КАИС дестабилизирующих факторов $\{DF\}$ с заданными значениями вероятности реализации и потенциальным ущербом; множество определенных организацией-владельцем требований $\{R_{DRIS}\}$; множество катастрофоустойчивых решений $\{DRS\}$.

На выходе функции будет информация о принятых решениях и вектор измененного состояния КАИС S_{new} .

Основываясь на принципах, изложенных в [6, 8, 9], в структуре разработанной модели иммунной системы можно выделить следующие основные блоки (рис. 2):

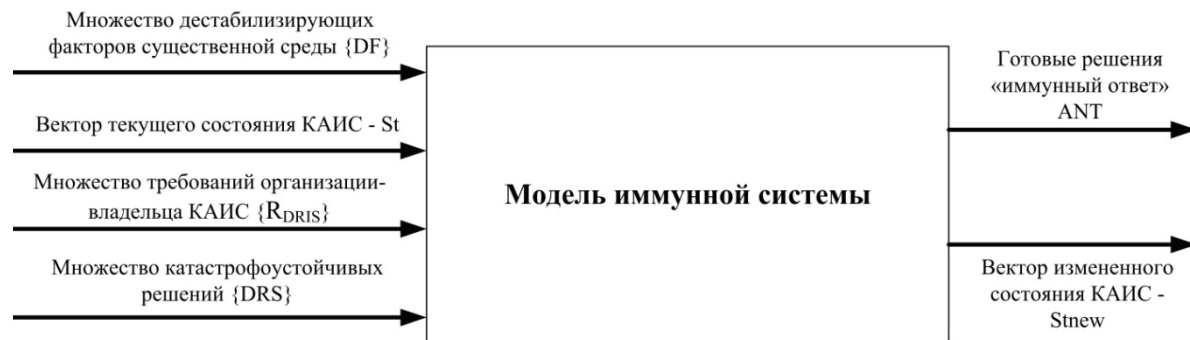


Рис. 1. Модель искусственной иммунной системы

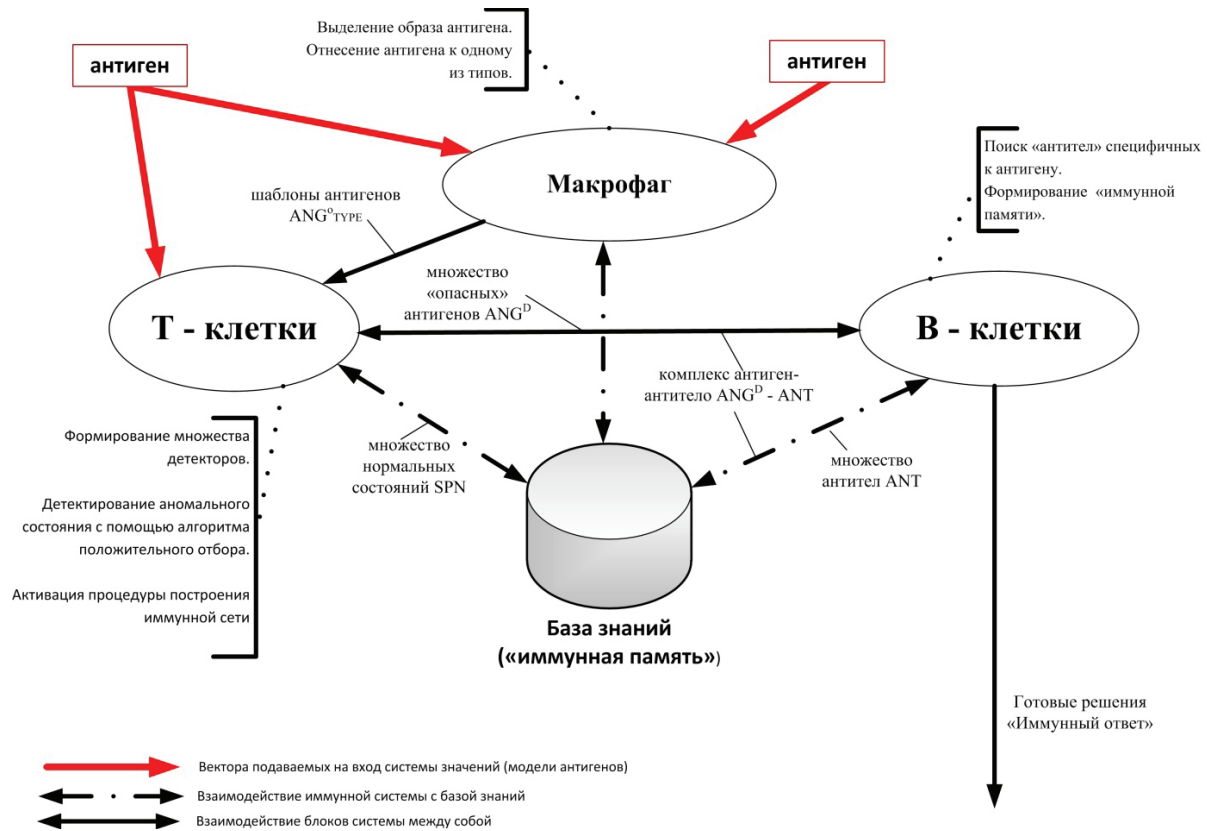


Рис. 2. Структура иммунной системы принятия решений

- «макрофаг» отвечает за выделения образов (шаблонов) антигенов из поданных на вход иммунной системы векторов значений и отнесение к одному из двух типов антигенов ANG^o_{TYPE} ;
- «Т-клетки Тимуса» предназначены для идентификации и выявления «аномальных» состояний КАИС;
- «В-клетки», инициирующие запуск процедуры «иммунного ответа», — поиска наиболее эффективных катастрофоустойчивых решений в случае обнаружения «аномальных» состояний системы и выявления потенциальных «антигенов» — наиболее критичных по уровню риска, вероятности реализации и объему потенциального ущерба дестабилизирующих факторов;
- «клетки иммунной памяти», содержащие информацию о ранее принятых системой решениях, правилах классификации и распознавания состояний системы, представляют собой базу знаний ИМС.

Для проведения анализа и оценки катастрофоустойчивости ИС в соответствии с предложенной методикой и формальной моделью автором было разработано алгоритмическое и программное обеспечение системы анализа катастрофоустойчивости ИС, архитектура которой представлена на рис. 3.

На основании собранных данных о технико-эксплуатационных характеристиках анализируемой ИС и требований, предъявляемых организацией-владельцем ИС к показателям катастрофоустойчивости, данная система позволяет [1, 3, 10, 11]:

- рассчитать такие показатели катастрофоустойчивости системы, как уровень катастрофоустойчивости; класс доступности, время восстановления; объем потерянных данных, живучесть системы;
- составить модель дестабилизирующих факторов (множество катастроф различного характера с указанием вероятности реализации и потенциального ущерба по каждому воздействию), потенциально опасных для ИС;
- оценить степень соответствия показателей катастрофоустойчивости системы требованиям организации-владельца;
- составить карту рисков и отобрать наиболее критичные для ИС дестабилизирующие факторы;
- составить различные варианты проектов катастрофоустойчивых решений и найти наиболее эффективный из них.

В соответствии с подходами к построению моделей ИС, предложенными в работах [12–14], при проведении экспериментальных исследований разработанной системы были использованы модели локальных и распределенных ИС. При проведении моделирования были получены следующие данные, представленные в таблице. В таблице приведены значения рассчитанных показателей катастрофоустойчивости ИС с различными технико-эксплуатационными характеристиками до и после применения предложенных разработанной системой проектов катастрофоустойчивых решений.

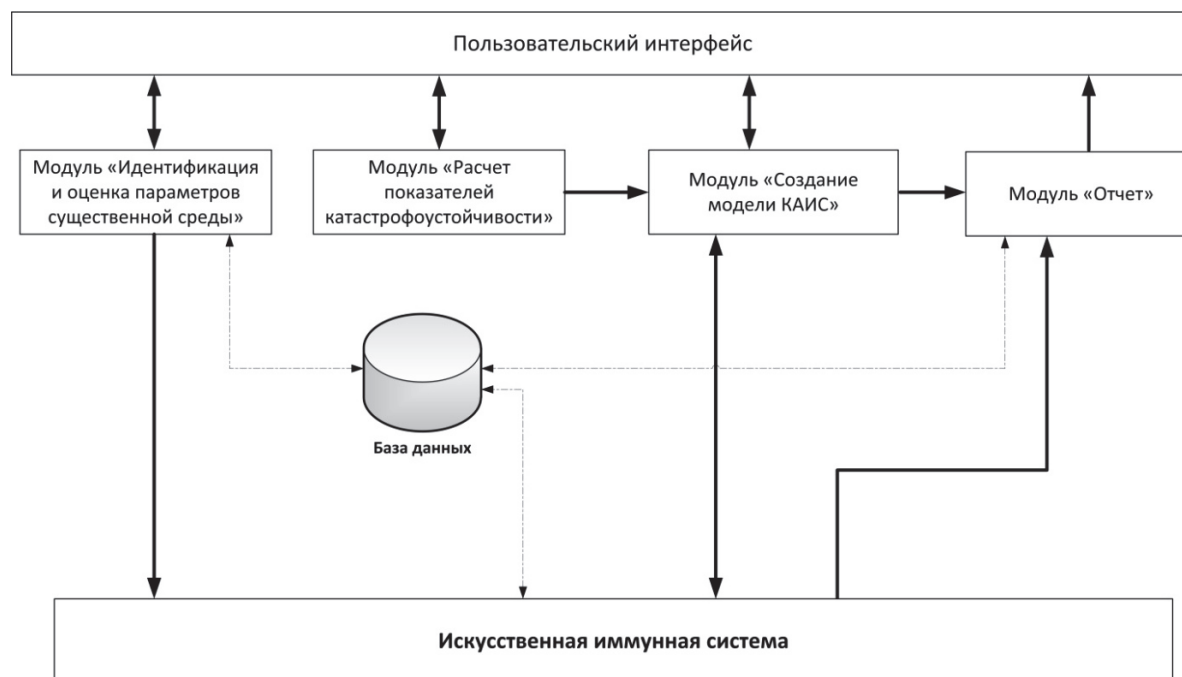


Рис. 3. Архитектура системы анализа катастрофоустойчивости

Таблица. Значение показателей катастрофоустойчивости ИС до и после принятия катастрофоустойчивых решений

Информационная система						
Тип	на базе локальной сети			распределенного типа		
Показатели катастрофоустойчивости	1	2	3	1	2	3
T_R (до принятия катастрофоустойчивых решений), ч	46	50	25	24	30	8
T_R' (после принятия катастрофоустойчивых решений), ч	32	35	16	16	20	7
L (до принятия катастрофоустойчивых решений)	2	2	3	4	3	5
L (после принятия катастрофоустойчивых решений)	3	3	5	5	4	5
D_{class} (до принятия катастрофоустойчивых решений)	1	1	2	4	3	5
D_{class}' (после принятия катастрофоустойчивых решений)	1	2	3	4	4	5
Z (до принятия катастрофоустойчивых решений)	0,6	0,58	0,83	0,83	0,86	0,95
Z' (после принятия катастрофоустойчивых решений)	0,75	0,72	0,91	0,91	0,89	0,96

СПИСОК ЛИТЕРАТУРЫ

1. Аткина В.С. Инновационные подходы в оценке и исследовании катастрофоустойчивости информационных систем // Актуальные вопросы информационной безопасности региона в условиях модернизации общества и внедрения инновационных технологий: Матер. Регион. научно-практ. конф. – Волгоград, 9–10 июня 2011. – Волгоград: Изд-во ВОЛГУ, 2011. – С. 168–172.
2. Будзко В.И. Количественные оценки отказоустойчивых и катастрофоустойчивых решений // Вопросы защиты информации. – 2003. – № 2. – С. 19–32.

Анализ представленных данных показывает, что применение рекомендованных решений на практике поможет улучшить значения ряда показателей катастрофоустойчивости системы: сократить среднее время восстановления системы на 32 %; повысить средний уровень живучести ИС на 10 %; повысить уровень катастрофоустойчивости системы в 67 % на одну позицию и в 16 % – на две; в 50 % случаях также наблюдается повышение класса доступности системы.

Таким образом, можно сделать вывод о том, что разработанная система анализа катастрофоустойчивости может применяться как на этапе проектирования и разработки катастрофоустойчивых ИС, так и в процессе их эксплуатации в качестве инструмента мониторинга и поддержки принятия решений. Полученные в результате использования системы анализа катастрофоустойчивости данные будут иметь большое значение при разработке плана реализации катастрофоустойчивых решений, поэтапное внедрение и своевременная корректировка которых позволит не только уменьшить потери в случае реализации катастрофы, но и по возможности полностью нейтрализовать ее воздействие.

3. Аткина В.С. Подход к оценке катастрофоустойчивости информационной системы // Информационные технологии, системный анализ и управление: Матер. VIII Всерос. науч. конф. молодых ученых, аспирантов и студентов. – Таганрог, 2010. – С. 240–243.
4. Atkina V.S. Semantic model of disaster recovery information system // European Science and Technology: international scientific conference. – Wiesbaden, Germany, 2012. – P. 162–164.
5. Гладыш С.В. Иммунотехнологии в управлении инцидентами информационной безопасности // Искусственный интеллект. – 2008. – Вып. 1. – С. 123–130.

6. Литвиненко В.И., Бидюк П.И., Фефелов А.А., Баклан И.В. Гибридная иммунная сеть для решения задач структурной идентификации // Нейронные сети. – 2006. – № 9. – С. 143–155.
7. Аткина В.С. Применение иммунной сети для анализа катастрофоустойчивости информационных систем // Известия ЮФУ. Технические науки. Информационная безопасность. – 2011. – № 12. – С. 203–210.
8. Фокин В.А. Статистическое моделирование данных при оценке состояния биологических систем // Известия Томского политехнического университета. – 2007. – Т. 311. – № 5. – С. 132–135.
9. Хаитов Р.М., Игнатъева Г.Л., Сидорович И.Г. Иммунология. – М.: Медицина, 2000. – 432 с.
10. Аткина В.С. Использование программного комплекса для исследования катастрофоустойчивости информационных систем // Вестник Волгоградского государственного университета. Серия 10. Инновационная деятельность. – 2011. – Вып. 5. – С. 14–18.
11. Аткина В.С. Оценка эффективности катастрофоустойчивых решений // Вестник Волгоградского государственного университета. Серия 10. Инновационная деятельность. – 2012. – Вып. 6. – С. 89–93.
12. Погребной В.К. О построении активных моделей распределенных систем реального времени // Известия Томского политехнического университета. – 2008. – Т. 312. – № 5. – С. 78–84.
13. Погребной А.В., Погребной Д.В. Проектирование структуры локальной сети для распределенной вычислительной системы реального времени // Известия Томского политехнического университета. – 2007. – Т. 311. – № 5. – С. 91–96.
14. Вейбер В.В., Кудинов А.В., Марков Н.Г. Задача сбора и передачи технологической информации распределенного промышленного предприятия // Известия Томского политехнического университета. – 2011. – Т. 319. – № 5. – С. 69–74.

Поступила 23.09.2012 г.

УДК 004.931

РАСПОЗНАВАНИЕ СТРУКТУРИРОВАННЫХ СИМВОЛОВ НА ИЗОБРАЖЕНИЯХ С ИСПОЛЬЗОВАНИЕМ ГИСТОГРАММ СРЕДНЕЙ ИНТЕНСИВНОСТИ И СВЕРТОЧНОЙ НЕЙРОННОЙ СЕТИ

А.А. Друки

Томский политехнический университет
E-mail: druki2008@yandex.ru

Разработаны и представлены алгоритм выделения области расположения символов на сложном фоне и алгоритм выделения символов на основе гистограмм средней интенсивности. Для решения задачи распознавания символов разработана и представлена сверточная нейронная сеть.

Ключевые слова:

Обработка изображений, распознавание символов, нейронные сети, гистограммы средней интенсивности.

Key words:

Image processing, character recognition, neural networks, histogram of average intensity.

Введение

Современные технологические, производственные и офисные системы в процессе своего функционирования используют информацию о маркировке объектов. Информация о маркировке грузов, вагонов, контейнеров, автомобильных номерных знаков позволяет рациональным образом организовывать процесс технологической обработки информации, вести учет и контроль изделий, материалов, транспортных средств. В основе процессов использования маркировки (текстово-цифровых меток) лежит технология автоматизированного распознавания структурированных символов. Потребность в такой технологии вызвала необходимость создания методов, моделей и систем распознавания структурированных символов [1].

В настоящее время такие технологии реализуются тремя традиционными методами – структурным, признаковым и шаблонным. Каждый из этих методов ориентирован на свои условия применения, для которых они являются эффективными.

Вместе с тем всем этим методам присущи недостатки. Наиболее существенные из них – низкая устойчивость к изменениям масштаба, смещениям, поворотам, смене ракурса и прочим искажениям.

Эти недостатки особенно ярко проявились при масштабной эксплуатации программно-технологических систем, использующих в своей основе эти методы. Практически у всех систем распознавания структурированных символов точностные характеристики резко падают и становятся ниже технологически приемлемых при искажении аффинными и проекционными преобразованиями. Вместе с тем технологические условия получения информации о маркировке не позволяют полностью устранить эти искажения [2].

Анализ методов распознавания структурированных символов показал, что для решения данной задачи эффективно использовать искусственные нейронные сети, в связи с тем, что они являются слабо чувствительными к искажениям входного сигнала, а так же обеспечивают возможность полу-

чения классификатора, хорошо моделирующего сложную функцию распределения изображений символов, тем самым увеличивая точность решения по сравнению с остальными методами.

Целью работы является разработка методов, алгоритмов и программ распознавания автомобильных номерных знаков на изображениях, инвариантных к аффинным и проекционным искажениям и удовлетворяющих требованиям:

1. Программная система должна обеспечивать вероятность распознавания изображений, содержащих государственные регистрационные знаки транспортных средств на расстоянии 40 м, не менее 80 %.
2. Цикл обработки не должен превышать одной секунды.

Выделение области расположения символов на изображении

Алгоритм работы системы можно разделить на 3 этапа:

1. Выделение области расположения символов на изображении.
2. Выделение отдельных символов.
3. Распознавание символов.

Для анализа исходного изображения и поиска на нём объектов необходимо провести бинаризацию, которая представляет собой перевод полноцветного изображения в монохромное, где присутствуют только два типа пикселей (темные и светлые). Бинаризация изображения основывается на сравнении интенсивности каждого пикселя с пороговым значением интенсивности. Если значение интенсивности пикселя выше значения интенсивности порога, то данному пикселю присваивается значение 255, или в противном случае 0 [3].

Пороговое значение T вычисляется по следующей формуле:

$$T = \frac{I_{\max} + I_{\min}}{2},$$

где $I_{\max}$ — максимальное значение интенсивности изображения, $I_{\min}$ — минимальное значение интенсивности изображения.

Выделение области расположения символов на изображении начинается со сканирования входного изображения окном размером 3×3 :

P_0	P_1	P_2
P_7	$F(i,j)$	P_3
P_6	P_5	P_4

Внутри окна вычисляется количество точек с интенсивностью 255, затем вычисляется отношение количества данных точек и площади сканирующего окна и сравнивается с пороговым значением T . Если данное отношение превышает заданное пороговое значение, то данная область изображения определяется как область-кандидат на содержание пластины номерного знака автомобиля. Экспериментально было установлено, что данный порог равен 0,3.

Окно перемещается попиксельно по всему изображению и для точки, оказавшейся в центре окна, вычисляется новое значение интенсивности по формуле:

$$K_s(i, j) = \sqrt{X^2 + Y^2},$$

$$X = [P_2 + 2P_3 + P_4] - [P_0 + 2P_7 + P_6],$$

$$Y = [P_0 + 2P_1 + P_2] - [P_6 + 2P_5 + P_4],$$

где $K_s(i, j)$ — новое значение интенсивности точки с координатами (i, j) ; P_n — значение интенсивности n -го пикселя.

Затем на изображении производится исследование областей, имеющих наибольший контраст, и вычисляется количество их граней. Если количество граней находится в некотором диапазоне, данная область отмечается как кандидат на местоположение номерного знака. Результатом этого подхода является нахождение на изображении всех областей кандидатов, в которых может содержаться пластина номерного знака [4].

Основным достоинством данного метода является то, что он основан на использовании информации о символах и номерной пластине и не чувствителен к параметрам контрастности и цвета изображения.

Недостаток данного метода заключается в том, что результатом определения границ является большое количество линий как горизонтальных, так и вертикальных, следовательно, большое количество ложных областей изображения может быть отнесено к кандидатам на содержание номерного знака. Поэтому предлагается использовать выделение только вертикальных границ изображения, во избежание переизбытка количества границ и ложных областей кандидатов.

Выделение символов с помощью гистограмм средней интенсивности

Для выделения строки символов на изображении в каждой области-кандидате предлагается строить гистограммы средней интенсивности, каждая из которых строится под заданным углом.

Гистограмма средней интенсивности представляет собой график с осями координат x и y (рис. 1). Изображение сканируется слева направо, сверху вниз, при этом вычисляется средняя интенсивность в каждом столбце. В тех местах, где нет символа, средняя интенсивность будет значительно отличаться от интенсивности тех мест, где символы присутствуют. Далее, выполняя ту же операцию по строкам, получается набор отдельных символов, которые уже можно анализировать [1].

В результате работы этого подхода получается набор изображений объектов, в этот набор входят как символы, так и различные помехи, которые необходимо устранить. Это достигается путем проверки условий, которым должен удовлетворять символ как геометрический объект [5].

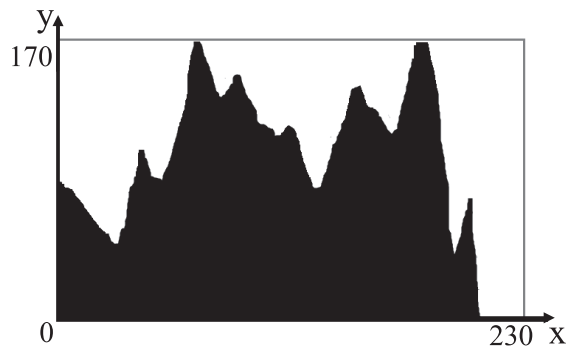


Рис. 1. Пример гистограмм средней интенсивности

Во-первых, проверяется выполнение условия $0,9 \leq \frac{b}{a} \leq 3,5$, где a и b – соответственно ширина и высота объекта. Объекты, не удовлетворяющие данному соотношению, отбрасываются из рассмотрения.

Второе условие имеет целью удаление мелких областей, не представляющих интереса. Среди объектов, удовлетворяющих вышеописанным условиям, выбирается объект с максимальной площадью описываемого прямоугольника $S_{\max} = ab$. Затем среди оставшихся областей выбираются те, которые удовлетворяют условию $\frac{S}{S_{\max}} \geq 0,1$, где S –

площадь прямоугольника объекта-кандидата.

Для того чтобы отделить строку символов от всего изображения, для начала предлагается вычислить горизонтальные гистограммы средней интенсивности. Так как на изображении самой яркой областью является фон номерной пластины, то два наибольших максимума будут соответствовать областям 1 и 2 (рис. 2).



Рис. 2. Построение горизонтальных гистограмм. Линиям 1 и 2 соответствуют два наибольших максимума. x – номер строки изображения, y – средняя интенсивность строки изображения

Далее строятся вертикальные гистограммы под углом взаимно перпендикулярным к n , и выделяется уже около 10 максимумов в промежутках между символами. Таким образом, выделяются области расположения отдельных символов на номерной пластине, которые можно дальше анализировать (рис. 3). Ненужные мелкие объекты, не являющиеся символами, отбрасываются из рассмотрения путём сравнения с некоторыми условиями.

При регистрации изображение номерных пластин подвергается аффинным и проекционным искажениям, следовательно, линии, соответствующие областям 1 и 2, будут располагаться не по го-

ризонтالي, а под неизвестным углом. В связи с чем предлагается строить не одну, а n гистограмм средней интенсивности, каждая из которых строится не по горизонтали, а под заданным углом (рис. 4).

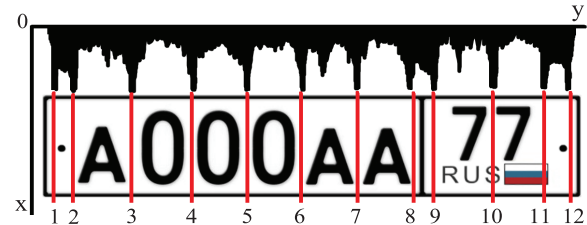


Рис. 3. Построение вертикальных гистограмм. Линиям от 1 до 12 соответствуют наибольшие максимумы. y – номер строки изображения, x – средняя интенсивность строки изображения

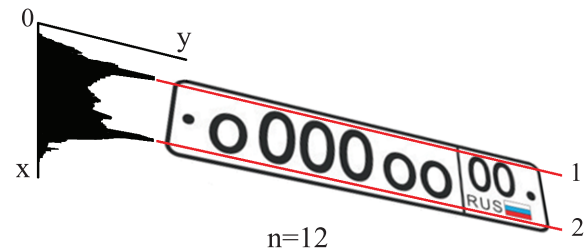


Рис. 4. Изображение номерной пластины, его гистограмма, построенная под углом 12° и выделенная область изображения, содержащая строку символов

Необходимое количество гистограмм средней интенсивности определяется из технических условий регистрации изображений, по данным условиям угол разворота изображения не превышает 15° по горизонтали как в правую, так и в левую сторону, следовательно, $n=31$. Из n построенных гистограмм выбирается та, которая содержит наибольшее значение по y , т. к. наибольшее значение будет соответствовать области 1 или 2 (рис. 4) [5, 6].

Распознавание символов с помощью сверточной нейронной сети

Анализ методов распознавания образов показал, что для решения данной задачи эффективно использовать искусственные нейронные сети. Преимущество использования нейронных сетей для обработки изображений – обучаемость системы для выделения ключевых характеристик символов из учебных наборов [7].

Наиболее часто в задачах распознавания и идентификации изображений используются классические нейросетевые архитектуры (многослойный персептрон, сети с радиально-базисной функцией и др.), но из анализа данных методов и экспериментальных исследований следует, что применение классических нейросетевых архитектур в данной задаче неэффективно по следующим причинам:

- изображения имеют большую размерность, соответственно возрастает размер нейронной сети;

- большое количество параметров увеличивает вместимость системы и соответственно требует большей тренировочной выборки, увеличивает время и вычислительную сложность процесса обучения;
- для повышения эффективности работы системы желательно применять несколько нейронных сетей (обученные с различными начальными значениями синаптических коэффициентов и порядком предъявления образов), но это увеличивает вычислительную сложность и время выполнения задачи;
- отсутствует инвариантность к изменениям масштаба изображения, ракурсов съёмки камеры и других геометрических искажений входного сигнала [7, 8].

Поэтому для распознавания символов на номерных знаках были выбраны сверточные нейронные сети, т. к. они обеспечивают частичную устойчивость к изменениям масштаба, смещениям, поворотам, смене ракурса и прочим искажениям.

Архитектура сверточной нейронной сети состоит из многих слоёв. Слои бывают двух типов: сверточные и подвыборочные, они чередуются друг с другом.

Нейроны в пределах слоя организованы в плоскости. В каждом слое имеется набор из нескольких плоскостей, причём нейроны одной плоскости имеют одинаковые синаптические коэффициенты, ведущие ко всем локальным участкам предыдущего слоя. Каждый нейрон слоя получает входы от некоторой области предыдущего слоя (локальное рецептивное поле), т. е. входное изображение предыдущего слоя как бы сканируется небольшим окном и пропускается сквозь набор синаптических коэффициентов, а результат отображается на соответствующий нейрон текущего слоя. Набор плоскостей представляет собой карты характеристик, и каждая плоскость находит «свои» участки изображения в любом месте предыдущего слоя. Размер локального рецептивного поля выбирается самостоятельно в процессе разработки нейронной сети [9].

Подвыборочный слой уменьшает масштаб плоскостей путём локального усреднения значений

выходов нейронов. Таким образом достигается иерархическая организация. Последующие слои извлекают более общие характеристики, меньше зависящие от искажений изображения.

Постепенно нейронная сеть обучается выделять ключевые характеристики поступающих на вход изображений [9, 10].

Разработана нейронная сеть, состоящая из 3-х скрытых слоёв, представлена на рис. 5. Входными данными нейронной сети являются отмасштабированные изображения размером 29×29 пикселей.

Следом за входным слоем находится первый скрытый слой, который является свёрточным. Он состоит из шести карт признаков размером 13×13. Каждый элемент карты признаков соединен с рецептивным полем размером 5×5 на входном изображении и единичным смещением. Следовательно, каждый элемент карты имеет 26 обучаемых весовых коэффициентов. Значения всех элементов карты признаков вычисляются путем последовательного сканирования рецептивным полем входного слоя. Таким образом, в первом скрытом слое содержится 26364 синаптических связей и 156 обучаемых параметров.

Размер свёрточной плоскости определяется в соответствии со следующими выражениями:

$$w_c = \frac{(w_u - 3)}{2},$$

где w_c — ширина свёрточной плоскости; w_u — ширина плоскости предыдущего слоя.

$$h_c = \frac{(h_u - 3)}{2},$$

где h_c — высота свёрточной плоскости; h_u — высота плоскости предыдущего слоя.

Второй скрытый слой также является слоем свёртки. Он состоит из 50 карт признаков размером 5×5. В этом слое 5×5×50=1250 нейронов. Каждый элемент в карте признаков связан с шестью областями размером 5×5 шести карт предыдущего слоя. Таким образом, во втором скрытом слое содержится 7550 весов и 188750 связей.

Два первых свёрточных слоя можно рассматривать как слои для извлечения признаков из изобра-

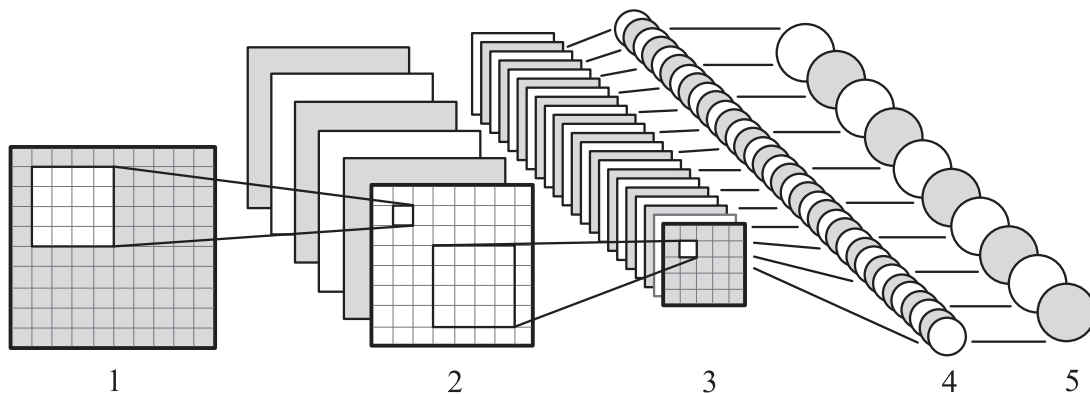


Рис. 5. Архитектура свёрточной нейронной сети: 1) вход; 2, 3) скрытые сверточные слои; 4) слой из обычных нейронов; 5) выходы сети

жения. Следующие два слоя являются слоями классификации. В этих слоях каждый нейрон соединён со всеми нейронами предыдущего слоя.

Третий скрытый слой состоит из 100 нейронов. В этом слое имеется 125100 связей и 125100 обучаемых параметров. Для десяти классов изображений, соответствующих десяти цифрам, 100 нейронов достаточно для хорошей обобщающей способности сети.

Данная нейронная сеть работает как классификатор, поэтому четвёртый выходной слой состоит из 21 нейрона, так как распознается 21 символ. Согласно ГОСТ Р 50577-93 на регистрационные знаки транспортных средств могут содержать следующие символы: А, В, Е, К, М, Н, О, Р, С, Т, Х, У и все цифры от 0 до 9. Следовательно система должна распознавать 21 символ.

В качестве активационной функции был выбран гиперболический тангенс:

$$f(a) = \operatorname{Atanh}(Sa).$$

где A – амплитуда этой функции, S – определяет ее положение относительно начала отсчета. Функция f – нечетная, ее горизонтальные асимптоты равняются A и $-A$.

Эта функция имеет ряд преимуществ для решения задачи:

- симметричные активационные функции, типа гиперболического тангенса, обеспечивают более быструю сходимость, чем стандартная логистическая функция;
- имеет непрерывную первую производную;
- имеет простую производную, которая может быть вычислена через ее значение, что дает экономии вычислений [11].

Формула функционирования нейрона свёрточного слоя:

$$y_k^{(i,j)} = b_k + \sum_{s=1}^K \sum_{t=1}^K w_{k,s,t} x^{((i-1)+s, (j-1)+t)},$$

где $y_k^{(i,j)}$ – нейрон k -й плоскости свёрточного слоя; b_k – нейронной смещение k -й плоскости; K – размер рецептивной области нейрона; $w_{k,s,t}$ – матрица синаптических коэффициентов; x – выходы нейронов предыдущего слоя.

Используется стандартный для нейронных сетей алгоритм обратного распространения ошибки. Для измерения качества распознавания использовалась функция среднеквадратической ошибки:

$$E_p = \frac{1}{2} \sum_j (t_{pj} - o_{pj})^2,$$

где E_p – величина функции ошибки для образа p ; t_{pj} – желаемый выход нейрона j для образа p ; o_{pj} – действительный выход нейрона j для образа p .

Окончательная коррекция синаптических коэффициентов происходит по формуле:

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_{pj} o_{pj},$$

где η – коэффициент пропорциональности, влияющий на скорость обучения.

На каждой итерации алгоритм обратного распространения ошибки рассчитывается для всего обучающего набора данных, чтобы вычислить средний или истинный градиент [12].

Когда ненастроенной сети предъявляется входной образ, она выдает некоторый случайный выход. Функция ошибки представляет собой разность между текущим выходом сети и идеальным выходом, который необходимо получить. Для успешного обучения сети требуется приблизить выход сети к желаемому выходу, т.е. последовательно уменьшать величину функции ошибки. Это достигается настройкой межнейронных связей. Каждый нейрон в сети имеет свои веса, которые настраиваются, чтобы уменьшить величину функции ошибки [13, 14].

Значения весовых коэффициентов были выбраны случайным образом из нормального распределения с нулевым средним и стандартным отклонением:

$$\delta_w = \sqrt{m},$$

где m – число связей входящих в нейрон.

Экспериментально доказано, что локальное рецептивное поле, равное 5×5 пикселей, позволяет избежать вычислительной избыточности и обеспечить надежное распознавание символов.

Для обучения сети использовалась база изображений, состоящая из 5000 символов. Значения исходных синаптических весов для всех слоёв сети с гиперболическим тангенсом выбираются на основе равномерного распределения с нулевым математическим ожиданием и дисперсией, обратной квадратному корню, из количества синаптических связей нейрона.

Нейронная сеть обеспечивает вероятность распознавания символов на уровне 95 % и скорость работы 0,5 секунды. В ходе экспериментов было установлено, что сеть ошибочно распознаёт символы при отклонении пластины номерного знака по горизонтали и вертикали более 45° и повороте пластины на плоскости до 15° . В этих случаях атрибуты символов наслаиваются друг на друга и не поддаются распознаванию даже человеческим глазом.

При различных шумах на изображениях, таких как грязь, снег, дождь, которые визуальнo меняют структуру символов, вероятность распознавания составляет 70 %, это является хорошим результатом по сравнению с другими методами. Одним из достоинств свёрточных нейронных сетей является то, что они игнорируют мелкие дефекты и шумы на изображениях, тем самым обеспечивая высокую вероятность распознавания зашумлённых изображений.

Заключение

На основе представленных алгоритмов выделения и распознавания символов разработана программная система, которая обеспечивает точность распознавания символов на уровне 95 % и соответствует техническим требованиям. Разработана че-

тырёхслойная сверточная нейронная сеть без слоёв подвыборки, обеспечивающая инвариантность к изменениям условий съёмки и поворотам изображений. Несмотря на простую структуру сети, эксперименты показали высокие результаты распознавания символов. Преимущества разработанных

методов заключаются в быстрой работе (0,5 секунд), инвариантности к искажениям входного сигнала и высокой вероятности распознавания, недостатки же заключаются в сложности настройки весовых коэффициентов сети и выборе её структуры.

СПИСОК ЛИТЕРАТУРЫ

1. Фаворская М.Н., Зотин А.С., Горошкин А.Н. Морфологическая обработка контурных изображений в системах распознавания текстовых символов // Вестник СибГАУ. – 2007. – № 1 (14). – С. 70–75.
2. Бредихин Р.Н. Об одном подходе к распознаванию оптических образов текстов // Вестник МЭИ. – 2005. – № 2. – С. 134–141.
3. Садыхов Р.Х., Ваткин М.Е. Модифицированный алгоритм обучения РБФ-сети для распознавания рукописных символов // Идентификация образов. – 2001. – Т. 1. – № 3. – Р. 7–16.
4. Нгуен Тоан Тханг, Спицын В.Г. Алгоритмическое и программное обеспечение для распознавания формы руки в реальном времени с использованием SURF-дескрипторов и нейронной сети // Известия Томского политехнического университета. – 2012. – Т. 320. – № 5. – С. 48–54.
5. Буй Тхи Тху Чанг, Фан Нгок Хоанг, Спицын В.Г. Распознавание лиц на основе применения метода Виолы-Джонса, вейвлет-преобразования и метода главных компонент // Известия Томского политехнического университета. – 2012. – Т. 320. – № 5. – С. 54–59.
6. Болотова Ю.А., Спицын В.Г., Фомин А.Э. Применение модели иерархической временной памяти в распознавании изображений // Известия Томского политехнического университета. – 2011. – Т. 318. – № 5. – С. 60–63.
7. Кермани Коланкех А., Спицын В.Г., Хамкер Ф. Нахождение параметров и удаление постоянной составляющей фильтра Гаусса для обработки изображений // Известия Томского политехнического университета. – 2011. – Т. 318. – № 5. – С. 57–59.
8. Plamondon R., Srinari S. On-line and off-line handwriting recognition: A comprehensive survey // IEEE Transactions on pattern analysis and machine intelligence. – 2000. – V. 22. – P. 105–109.
9. Ronse C., Najman L., Decenciere E. Mathematical morphology: 40 years on // Proceedings of the VII International symposium of mathematical morphology. – Netherlands, 2005. – № 30. – P. 350–370.
10. Le Cun Y., Bengio Y. Convolutional networks for images, speech and time series // The handbook of brain theory and neural networks. – 1998. – V. 7. – № 1. – P. 255–258.
11. Rowley H.A., Baluja S., Kanade T. Neural network-based face detection // Pattern anal. mach. intell. – 2000. – V. 5. – P. 23–38.
12. Feraud R., Bernier O., Viallet J., Collobert M. A fast and accurate face detector based on neural networks // Transactions on pattern analysis and machine intelligence. – 2002. – V. 3. – № 23. – P. 42–53.
13. Yu N., Notkin B.S., Sedov V.A. Neuro-iterative algorithm of tomographic reconstruction of the distributed physical fields in the fibre-optic measuring systems // Computer optics. – 2009. – V. 33. – № 4. – P. 446–455.
14. Wilson D.R., Martinez T.R. The general inefficiency of batch training for gradient descent learning // Neural Networks. – 2004. – V. 16. – № 4. – P. 1429–1451.

Поступила 30.01.2013 г.

УДК 004.021

ПРИМЕНЕНИЕ ПРОСТРАНСТВЕННЫХ МЕТОДОВ УЛУЧШЕНИЯ ИЗОБРАЖЕНИЙ НА ВИДЕОПОСЛЕДОВАТЕЛЬНОСТЯХ

А.А. Белоусов, В.Г. Спицын

Томский политехнический университет

E-mail: artem.a.belousov@gmail.com

Исследована применимость методов обработки изображений на основе модели Illuminance–Reflectance и выравнивания гистограмм для обработки видеоданных. Произведена модификация указанных методов для применения в режиме реального времени. Применение указанных методов позволяет значительно улучшить такие показатели кадров в видеопоследовательности, как количество и интенсивность краевых пикселей и уровень адаптации к зрению человека по яркости.

Ключевые слова:

Модель освещенность–отраженность, выравнивание гистограмм, улучшение видеопоследовательностей.

Key words:

Illuminance–Reflectance model, histogram equalization, video enhancement.

Введение

Видеозаписи являются неотъемлемой частью современного мира. Бытовые видеокамеры, камеры потокового вещания, системы видеонаблюдения, контроля качества, машинного зрения — все эти устройства используются ежедневно для получения, записи и передачи видеопоследовательностей. Однако очень часто из-за несовершенства технических устройств или сложных условий среды качество получаемых видеозаписей является недостаточным для полноценного использования.

Подобная ситуация сложилась и в области создания цифровых изображений, однако в настоящий момент разработано множество разнообразных методов улучшения изображений, первые из которых были представлены еще в середине XX в. При этом, несмотря на кажущуюся схожесть методов улучшения статических изображений и видеопоследовательностей, ввиду значительно большего объема последних и ряда других их особенностей на методы повышения качества видеозаписей накладываются более строгие ограничения. В первую очередь это касается их вычислительной сложности. В ряде случаев обязательной является возможность применения таких методов в реальном времени, при этом на обработку одного кадра отводится не более 45 мс.

В данной статье рассматриваются два пространственных метода улучшения изображений — выравнивание гистограмм и улучшение на основе модели Illuminance–Reflectance. Оба этих метода позволяют эффективно улучшать качество изображений. Кроме того, оба являются достаточно быстродействующими и детерминированными, а значит, потенциально, могут быть использованы для улучшения видеопоследовательностей. В работе изучаются свойства этих методов, а также производится их адаптация для применения на видеозаписях.

Повышение четкости изображения с использованием модели Illuminance–Reflectance

Описание метода

Модель Освещенность–Отраженность (Illuminance–Reflectance, I–R) — это физическое описание формирования карты яркостей точек в реальных сценах. Она разделяет яркость объекта на две части:

1. **Освещенность** — интенсивность света, падающего на объект.
2. **Отраженность** — отражающая способность поверхности объекта.

Такое разделение освещенности и отраженности позволяет производить улучшение изображений с учетом физических особенностей формирования изображений.

Один из методов обработки на основе такой модели был представлен в [1]. Схема предложенного в этой работе алгоритма представлена на рисунке. Помимо перевода изображения в яркостное представление и обратно этот метод включает 4 основных этапа.

На первом этапе производится вычисление освещенности и отраженности на основе имеющейся таблицы яркостей точек. Принято считать, что яркость каждой точки изображения является производением освещенности и отраженности:

$$I(x, y) = L(x, y)R(x, y), \quad (1)$$

где $R(x, y)$ и $L(x, y)$ — отраженность и освещенность точки с координатами (x, y) соответственно, а $I(x, y)$ — яркость этой точки.

Кроме того, предполагается, что освещенность представляет собой низкочастотные компоненты изображения, в то время как отраженность — высокочастотные. Это предположение основано на том, что отраженность на изображении изменяется значительно быстрее, нежели освещенность, поскольку количество источников света в реальных сценах

значительно меньше, нежели количество объектов. Из этого правила есть некоторые исключения, такие как границы теней, в которых происходит резкая смена освещенности, однако в упрощенной модели ими можно пренебречь.



Рисунок. Схема I-R алгоритма

Существуют различные методики вычисления освещенности точки [2, 3]. В описываемом алгоритме было выбрано применение низкочастотного фильтра Гаусса к таблице яркостей. В пространственном представлении этот фильтр является дискретной двумерной сверткой с Гауссовым ядром:

$$L(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(m, n) G(m+x, n+y), \quad (2)$$

где L – освещенность точки, I – ее яркость в исходной таблице, а G – двумерная функция Гаусса размером $M \times N$. Функция G определяется как

$$G(x, y) = q \cdot \exp\left(\frac{-(x^2 + y^2)}{c^2}\right), \quad (3)$$

где q – нормирующий коэффициент:

$$\sum_x \sum_y q \cdot \exp\left(\frac{-(x^2 + y^2)}{c^2}\right) = 1, \quad (4)$$

а c – константа, определяющая ширину фильтра.

После вычисления освещенности по формуле (2) происходит вычисление отраженности по формуле (1).

На следующем этапе производится сжатие динамического диапазона освещенностей точек. Для этого применяется обратный оконный сигмоид

(Windowed inverse sigmoid, WIS). Функция сигмоида определяется как

$$f(v) = \frac{1}{1 + e^{-av}},$$

Эта функция применяется для сжатия диапазонов освещенностей в соответствии с формулами (5)–(7):

$$L' = L[f(v_{\max}) - f(v_{\min})] + f(v_{\min}), \quad (5)$$

$$L'' = \frac{1}{a} \ln\left(\frac{1}{L'} - 1\right), \quad (6)$$

$$L_{enh} = \frac{L'' - v_{\min}}{v_{\max} - v_{\min}}. \quad (7)$$

Сжатие проходит в 3 этапа. На первом этапе по формуле (5) исходный диапазон нормализованной освещенности $[0, 1]$ проецируется на входной диапазон функции WIS $[f(v_{\min}), f(v_{\max})]$. Затем при помощи соотношения (6) происходит преобразование полученных освещенностей с помощью WIS. Заключительным этапом является обратная проекция освещенностей точек на исходный диапазон $[0, 1]$ по формуле (7).

Параметры $v_{\min}$ и $v_{\max}$ влияют на форму преобразования, тем самым позволяя задавать степень сжатия динамического диапазона. Параметры $v_{\max}$, $v_{\min}$ и a могут быть выбраны пользователем для настройки сжатия динамического диапазона. В [1] были протестированы различные наборы этих параметров, и был сделан вывод, что параметры $v_{\max}$ и a должны быть равны 3 и 1 соответственно, а параметр $v_{\min}$ следует выбирать адаптивно в зависимости от изображения. Также был предложен следующий вариант выбора параметра $v_{\min}$:

$$v_{\min} = \begin{cases} -6, I_m \leq 70 \\ \frac{I_m - 70}{80} \times 3 - 6, 70 < I_m \leq 150 \\ -3, I_m \geq 150, \end{cases}$$

где I_m – средняя яркость по изображению.

На следующем этапе алгоритма производится усиление среднечастотных компонент. Этот этап требуется потому, что из-за упрощенного вычисления освещенности и отраженности некоторые компоненты отраженности (среднечастотные) попадают в освещенность, а значит, их динамический диапазон тоже сжимается, что приводит к некоторому ухудшению исходного изображения, особенно в случае изображений с низким контрастом или плавными изменениями яркости. Для компенсации такого искажения применяется метод повышения локальных контрастов. Этот метод основан на применении следующих формул:

$$L_{enh}'(x, y) = L_{enh}(x, y)^{E(x, y)},$$

$$E(x, y) = \left(\frac{I_{conv}(x, y)}{I(x, y)} \right)^p,$$

где I_{comp} — это низкочастотные компоненты исходной яркости точек, вычисленные по формулам (2)–(4), однако с большим значением параметра s . Значение параметра P аналогично $v_{\min}$ зависит от изображения и выбирается по формуле (8).

$$P = \begin{cases} 2, \sigma \leq 30 \\ -0,03\sigma + 2,9, & 30 < \sigma \leq 80 \\ \frac{1}{2}, \sigma > 80, \end{cases} \quad (8)$$

где σ — среднеквадратическое отклонение яркости на изображении.

Завершающим этапом обработки является восстановление таблицы яркостей точек и перевод изображения из яркостного представления в полноцветное.

Исследование свойств метода

Для изучения описанного метода было проведено его тестирование. В качестве тестового набора было выбрано 27 изображений различного размера. Для каждого из изображений были вычислены следующие атомарные оценки:

- Количество краевых пикселей ($\eta(I)$);
- Интенсивность краевых пикселей ($E(I)$);
- Мера энтропии изображения ($H(I)$);
- Уровень адаптации к зрению человека по яркости (LQ).

Кроме того, была вычислена комплексная оценка по формуле [6]

$$F(x) = \ln(\ln(E(I)) + e) \frac{\eta(I)}{H_{\text{size}} W_{\text{size}}} e^{H(I)} LQ.$$

Затем была произведена обработка каждого изображения из тестового набора при помощи I–R алгоритма, и была произведена оценка обработанных изображений. Для удобства обработки результатов работы были получены относительные оценки, равные отношению оценки обработанного изображения к соответствующей исходной оценке, и были посчитаны средние относительные оценки по каждому из параметров.

В результате тестирования выяснилось, что I–R алгоритм не приводит к значительным изменениям уровня адаптации к зрению человека по яркости, а значение энтропии в 15 случаях из 27 незначительно снизилось. Значение же интенсивности и количество краевых пикселей были значительно улучшены. В среднем применение данного этапа обработки позволило повысить эти параметры приблизительно в 2,01 и 3,83 раза соответственно.

Выравнивание гистограммы изображения

Гистограмма изображения с L возможными уровнями яркости в диапазоне $[0, G]$ определяется как дискретная функция

$$h(r_k) = n_k,$$

где r_k — k -я интенсивность в диапазоне $[0, G]$, а n_k — количество пикселей на изображении с таким значением интенсивности. Однако чаще используют

ся гистограммы, нормализованные на единицу:

$$p(r_k) = \frac{h(r_k)}{n} = \frac{n_k}{n},$$

где n — общее количество точек изображения.

Для человеческого зрения более благоприятны изображения с большим количеством различных уровней яркости. Из этого следует, что если гистограмма изображения слишком узкая, то ее необходимо растянуть вдоль горизонтальной оси [2].

Один из методов выравнивания гистограмм был описан Рафаэлем Гонсалесом и его соавторами в [4, 5].

Пусть $p(r_j)$ — нормализованная гистограмма входного изображения. Тогда для получения выходных уровней яркости s_k необходимо выполнить следующее преобразование:

$$s_k = \sum_{j=1}^k p(r_j) = \sum_{j=1}^k \frac{n_j}{n}$$

для всех $k=1, 2, 3, \dots, L$, где s_k — значение яркости k -го уровня яркости на выходном изображении, соответствующее значению r_k на входном.

Необходимо заметить, что при дискретном представлении яркостей в изображении s_k будет округляться в сторону ближайшего дискретного значения, то есть при низкой встречаемости точек с некоторой яркостью вследствие округления возможна потеря этой градации.

Кроме того, при наличии на изображении больших равномерных областей после обработки соседние градации яркости из этих областей будут значительно отличаться, что приведет к ступенчатости изображения.

Для исключения таких эффектов было решено ввести минимальное и максимальное ограничение на разницу между соседними градациями яркости [6]. Для этого выражение было модифицировано следующим образом:

$$s_k = \sum_{j=1}^k p'(r_j),$$

$$p'(r_j) = \begin{cases} D_{\min}, & p(r_j) \leq D_{\min}, \\ p(r_j), & D_{\min} < p(r_j) \leq D_{\max}, \\ D_{\max}, & p(r_j) > D_{\max}, \end{cases}$$

где $D_{\min}$ и $D_{\max}$ — минимальное и максимальное ограничение разности между соседними градациями.

С учетом того, что минимальное ограничение вводится для исключения возможности потери информации, значение $D_{\min}$ должно быть равно $1/L$, где L — количество возможных уровней яркости на изображении. Максимальное ограничение выбирается исследователем.

Данный метод не производит изменение количества различных градаций яркости, присутствующих на изображении, а значит, энтропия изображения не изменяется, а в ряде случаев может незначительно снизиться из-за потери градаций

с очень малым значением гистограммы. Однако за счет увеличения разницы между соседними градациями яркости удается повысить количество и интенсивность краевых пикселей. Кроме того, за счет сдвига градаций и расширения гистограммы удается повысить уровень приспособленности к зрению человека по яркости.

Для проверки выдвинутых предположений метод расширения гистограмм был протестирован аналогично I–R алгоритму. Тестирование производилось на том же наборе изображений.

В соответствии с ожиданиями, применение данного метода позволило добиться значительного повышения количества и интенсивности краевых пикселей – в 2,251 и 1,972 раза соответственно. Значение энтропии было снижено на 1,5 %.

Значение уровня адаптации к зрению человека по яркости, как и ожидалось, увеличилось. Увеличение составило 11,1 %, однако для уровня адаптации такой метод сравнения является не вполне корректным в связи с тем, что значения данного параметра лежат в диапазоне [0, 1], а идеальным значением является 1. Более корректной в данном случае является оценка отклонения от идеального значения. Среднее значение уровня адаптации для необработанного набора изображений составило 0,819, т. е. отклонение от идеального значения составляет 18,1 %. Среднее значение уровня адаптации после обработки составило 0,853, а отклонение от идеального значения снизилось на 3,4 % до уровня 14,7 %.

Адаптация методов для обработки видеоданных

Подавляющее большинство методов для обработки изображения применимы и для покадровой обработки видеоданных. При этом из-за значительного количества кадров одним из наиболее существенных ограничений становится быстродействие этих методов. Наиболее жесткие требования к скорости обработки предъявляются при обработке потоковых данных в режиме реального времени, при этом многие методы становятся неприменимыми.

Для исследования применимости рассмотренных методов улучшения изображений для покадровой обработки видеоданных в режиме реального времени была исследована скорость их работы на отдельных изображениях.

Результаты исследования скорости обработки изображений с использованием модели Illuminance–Reflectance приведены в табл. 1. Из таблицы видно, что среднее время обработки изображения размером 800×800 точек составило более секунды. При этом более 70,6 % (777 мс в среднем) времени затрачивалось на вычисление освещенности и отраженности, которое является обязательным этапом, т. е. добиться значительного сокращения времени обработки не удастся. Очевидно, что данный метод неприменим в реальном времени.

Затраты времени на обработку при помощи метода выравнивания гистограмм оказались значи-

тельно меньше (табл. 2). Непосредственная обработка одного изображения размером 800×800 точек в среднем занимает 4,9 мс, однако при этом формирование таблицы яркостей и восстановление изображения занимают суммарно 133 мс.

Таблица 1. Затраты процессорного времени на обработку I–R алгоритмом

Этап обработки	Количество вызовов	Среднее время (мс)	Суммарное время (мс)	Процент от общего времени
Получение таблицы яркостей	12	58	696	5,27
Извлечение освещенности и отраженности	12	777	9323	70,60
Сжатие динамического диапазона	12	38	457	3,46
Усиление средних частот	12	112	1349	10,22
Восстановление изображения	12	115	1380	10,45
Полная обработка изображения	12	1100	13205	100

Таблица 2. Затраты процессорного времени на выравнивание гистограмм

Этап обработки	Количество вызовов	Время (мс)		Процент от общего времени
		Среднее	Суммарное	
Получение таблицы яркостей	12	58	696	32,59
Выравнивание гистограммы	12	4,9	59	2,77
Восстановление изображения	12	115	1380	64,64
Полная обработка изображения	12	177,9	2135	100

Такие затраты времени являются слишком большими для обработки в режиме реального времени, однако данные затраты можно сократить.

Во-первых, процедура получения таблицы яркостей при пропорциональной обработке цветовой информации является достаточно трудоемкой, т. к. в ней применяется 3 умножения на каждую точку. Процедура восстановления является еще более трудоемкой: в ней помимо 3-х умножений применяются еще 3 деления, которые занимают еще больше машинного времени. С учетом малых затрат времени на обработку таблицы яркостей имеет смысл преобразовывать таблицы яркостей отдельно по красному, зеленому и синему цвету, при этом получение таблиц яркостей для каждого из них сведется к простому копированию.

Вторая модификация метода для обработки видеоданных основана на том, что последовательные кадры видеоряда, как правило, отличаются очень незначительно, а значит, отсутствует необходимость пересчитывать гистограмму для каждого кадра. Поскольку гистограмма используется только для преобразования яркостей, можно применять вычисленную для определенного кадра гистограмму для последующих N кадров. Длина последовательности таких кадров определяется спецификой видеозаписи.

Применение этих упрощений позволило добиться следующих результатов (табл. 3).

Таблица 3. Затраты процессорного времени на усовершенствованное выравнивание гистограмм

Этап обработки	Количество вызовов	Время (мс)		Процент от общего времени
		Среднее	Суммарное	
Получение таблицы яркостей	12	38	456	38,89
Вычисление гистограммы	12	6,3	75,6	6,45
Выравнивание гистограммы	12	8,4	100,8	8,6
Восстановление изображения	12	45	540	46,06
Полная обработка изображения	12	97,7	1172,4	100

Можно заметить, что суммарные затраты времени сократились до 97,7 мс для первого кадра, для последующих кадров они составляют 91,4 мс. С учетом того, что время вычисления зависит от площади изображения, такая скорость вычисления становится уже приемлемой для обработки в реальном времени видеозаписей с размером кадра до 640×480 пикселей при частоте кадров 24 кадра/с. Для проверки скорости быстрого действия бы-

ла реализована библиотека, позволяющая в реальном времени обрабатывать видеопоток с указанными характеристиками.

Необходимо так же учесть, что рассматриваемая реализация не использует аппаратное ускорение, которое может дать дополнительный прирост производительности.

Заключение

Описанные методы позволяют производить коррекцию яркостей любых изображений и при этом являются достаточно быстродействующими для применения на видеозаписях. Кроме того, при помощи предложенных упрощений удалось добиться достаточного для применения в реальном времени быстрого действия метода выравнивания гистограмм. Предложенные методы были протестированы на реальных данных, и тестирование подтвердило их высокую эффективность.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 12–08–00296а.

СПИСОК ЛИТЕРАТУРЫ

1. Tao L., Asari V.K. An Efficient Illuminance–Reflectance Nonlinear Video Stream Enhancement Model // SPIE-IS&T Electronic Imaging. – 2006. – V. 6063. – P. 60630I-1–60630I-12.
2. Rizzi A., Gatta C., Marini D. From Retinex to ACE: Issues in developing a new algorithm for unsupervised color equalization // Journal of Electronic Imaging. – 2004. – V. 13. – № 1. – P. 75–84.
3. Sobol R. Improving the Retinex algorithm for rendering wide dynamic range photographs // Proc. SPIE 4662. – 2002. – P. 341–348.
4. Гонсалес Р., Вудс Р. Цифровая обработка изображений. – М.: Техносфера, 2005. – 1072 с.
5. Gonzalez R.C., Woods R.E., Eddins S.L. Digital Image Processing Using MATLAB. – Upper Saddle River: Prentice Hall, 2004. – 344 p.
6. Белоусов А.А., Спицын В.Г. Двухэтапный метод улучшения изображений // Современная техника и технологии: Труды XIII Междунар. научно-практ. конф. студентов, аспирантов и молодых ученых. – Томск, 26–30 марта 2007. – Томск: Изд-во ТПУ, 2007. – Т. 2. – С. 282–284.

Поступила 02.04.2013 г.

РАЗРАБОТКА РЕЛЯЦИОННЫХ СРЕДСТВ ДЕКОМПОЗИЦИИ ПРЕДИКАТОВ

М.Н. Рудометкина, В.Г. Спицын, В.А. Эттель*

Томский политехнический университет

*Карагандинский государственный технический университет

E-mail: monika_7272@mail.ru

Статья посвящена разработке новых средств декомпозиции предикатов на основе утверждений о зависимостях из области нормализации реляционных отношений. Дано точное определение процедуре декомпозиции по отношению к реляционным отношениям и к предикатам. Рассмотрены утверждения о зависимостях, позволяющих проводить декомпозицию реляционных отношений; предложены их обобщения. Опираясь на связь между реляционной алгеброй и алгеброй предикатов, сформулированы утверждения о зависимостях, преобразованных в новые средства декомпозиции предикатов.

Ключевые слова:

Алгебра предикатов, декомпозиция, реляционная алгебра, зависимость соединения.

Key words:

Predicate algebra, decomposition, relational algebra, dependence of the connection.

Одна из идей алгебраизации математической логики привела к алгебре предикатов, которая активно используется как формульный аппарат описания дискретных отношений. Рассматривается разработка формальных средств построения моделей логических сетей. Эти средства основаны на теоремах из раздела нормализации теории реляционных баз данных и позволяют преобразовывать произвольные уравнения алгебры предикатов в модели логических сетей. Аппаратная реализация логических сетей на программируемых логических интегральных схемах позволит создать новые высокопроизводительные процессоры, обрабатывающие модели естественного языка.

Предикат в базисе дизъюнктивно-конъюнктивной алгебры предикатов (ДКАП) обычно рассматривается как аналитическая запись отношения. Поэтому декомпозиция предиката — это декомпозиция отношения на аналитическом уровне (попросту манипуляция с формулами), когда обрабатываются не сами отношения, а соответствующие им формулы предикатов.

Уточним понятие декомпозиции, как оно понимается в реляционных базах данных. Для этого приведем понятие зависимости соединения, которое используется в теории зависимостей.

Пусть $\Gamma = \{X_1, \dots, X_l\}$ — семейство подмножеств множества H . Если $H = \bigcup_{i=1}^l X_i$, то Γ называется покрытием множества H . Покрытие Γ называется разбиением множества H , если множества $X_1, \dots, X_l$ попарно не пересекаются.

Пусть R — некоторое отношение, а семейство множеств $\{X_1, \dots, X_l\}$ образует покрытие множества $\text{at}(R)$. Отношение R удовлетворяет зависимости соединения $^\circ[X_1, \dots, X_l]$, если

$$R = R[X_1] \circ R[X_2] \circ \dots \circ R[X_l]. \quad (1)$$

Выражение (1) является так называемым алгебраическим ограничением целостности [1].

Другие виды зависимостей, которые используются в теории зависимостей — функциональные

и многозначные — являются частными случаями зависимости соединения. Всякая зависимость, выполненная в $R \in \text{Re}(T)$, позволяет представить R в виде естественного соединения своих проекций (1). В теории реляционных баз данных обычно именно эту процедуру и называют декомпозицией R . Исходя из выражения (1) видно, что непосредственно декомпозиция выполняется путем проектирования исходного отношения по разным группам его атрибутов. Это говорит о том, что оператором декомпозиции служит операция проекции, в то время как оператором композиции (обратной сборки) является естественное соединение.

Однако разбиение отношения может осуществляться разными способами: наряду с проекцией в качестве оператора декомпозиции может быть использована выборка. Тогда оператором композиции уже будет не естественное соединение, а операция объединения. Когда в теории баз данных идет речь о декомпозиции отношений, то практически всегда имеется в виду декомпозиция вида (1). Другие виды практически не используются. Тем не менее, для точности декомпозицию отношений в виде (1) иногда будем называть проекционно-соединительной декомпозицией.

Декомпозиция предиката P — это получение некоторых предикатов $P_1, P_2, \dots, P_l$ на основе предиката P с возможностью его обратного восстановления из этих предикатов, которое называется композицией предиката P . Важным частным случаем декомпозиции является так называемая бинарная декомпозиция предиката P , характеризующаяся тем, что каждый из предикатов $P_1, P_2, \dots, P_l$, полученных в результате декомпозиции предиката P , имеет в точности два существенных аргумента. Если оператором композиции (восстановления) служит конъюнкция, то декомпозицию называют конъюнктивной.

Предикатными аналогами реляционных операций проекции являются кванторы существования, аналогом естественного соединения — конъюнк-

ция. Тогда предикатным аналогом проекционно-соединительной декомпозиции будет кванторно-конъюнктивный вид декомпозиции. Его определение заключается в следующем:

Пусть $T \subseteq V \times U$ – осто́в с конечным множеством имен переменных $V = \{x_1, x_2, \dots, x_n\}$, $S = \text{Dom} x_1 \times \text{Dom} x_2 \times \dots \times \text{Dom} x_n$. Пусть P – некоторый предикат из $\text{Pre}(S)$, а семейство множеств $\{X_1, X_2, \dots, X_j\}$ образует покрытие множества V . Будем говорить, что предикат P удовлетворяет зависимости конъюнкции $\wedge\{X_1, X_2, \dots, X_j\}$, если

$$\exists Y_1(P) \wedge \exists Y_2(P) \wedge \dots \wedge \exists Y_j(P) = P, \quad (2)$$

где $Y_i = V \setminus X_i$, $i=1, \dots, j$.

Представление предиката P в виде (2) будем называть кванторно-конъюнктивной декомпозицией.

Некоторые методы декомпозиции предикатов сводятся к тому, что исходный предикат P заменяется некоторым вспомогательным предикатом P' , который строится на основе исходного P с помощью введения одной или нескольких вспомогательных переменных $y_1, y_2, \dots, y_k$. Этот предикат связан с исходным предикатом P равенством

$$P = \exists y_1 \exists y_2 \dots \exists y_k (P'). \quad (3)$$

Хотя фактически производится кванторно-конъюнктивная декомпозиция вспомогательного предиката P' , но исходный предикат P восстанавливается с помощью равенства (3). Принято считать, что произведена декомпозиция предиката P .

Таким образом, декомпозиция предиката может выполняться без помощи вспомогательных переменных, либо с их помощью. В первом случае декомпозиция называется эквивалентной, во втором – неэквивалентной [2. С. 84]. Так, декомпозиция предиката P' является эквивалентной, а декомпозиция исходного P – неэквивалентной. Кванторно-конъюнктивная декомпозиция является эквивалентной по определению.

Рассмотрим понятие зависимостей и некоторые утверждения относительно них. Последние позволяют для заданного отношения разрешить вопрос о допустимости той или иной проекционно-соединительной декомпозиции. Рассмотрим частные случаи: функциональные и многозначные зависимости, определения которых взяты из [1].

Сначала рассмотрим понятие функциональной зависимости [1. С. 59].

Пусть задано отношение $R \in \text{Re}(T)$, а X и Y – произвольные подмножества множества $\text{at}(R)$. Говорят, что Y функционально зависит от X , если для любых двух функциональных данных $f_1, f_2 \in R$ из равенства $f_1[X] = f_2[X]$ следует равенство $f_1[Y] = f_2[Y]$. Если Y функционально зависит от X , то пишут $X \rightarrow Y$ и говорят, что отношение R удовлетворяет функциональной зависимости $X \rightarrow Y$ или, что функциональная зависимость $X \rightarrow Y$ выполняется в отношении R .

Непосредственно из определения следует, что если зависимость $X \rightarrow Y$ выполняется в R , то она выполняется в любой его проекции, содержащей множество атрибутов $X \cup Y$.

Если $R \in \text{Re}(T)$ – произвольное отношение, то множество пар (X, Y) таких, что $X \rightarrow Y$ в R , обозначается через $\text{FD}(R)$. Оно называется структурой функциональных зависимостей отношения R [1. С. 205].

Пусть R – произвольное отношение. В [1. С. 204, 205] доказано, что структура функциональных зависимостей $\text{FD}(R)$ обладает следующими ниже свойствами (X, Y, Z, W – подмножества множества $\text{at}(R)$).

$$\text{Если } X \supseteq Y, \text{ то } X \rightarrow Y. \quad (4)$$

$$\text{Если } X \rightarrow Y \text{ и } Y \cup Z \rightarrow W, \text{ то } X \cup Z \rightarrow W. \quad (5)$$

Для дедуктивной системы, заданной правилами (4), (5), алгоритмически разрешима проблема выводимости или логического следования, что означает, что анализ структур функциональных зависимостей может осуществляться автоматически [1. § 5.1]. В частности, существуют алгоритмы, позволяющие для заданного множества функциональных зависимостей Σ найти множество всех зависимостей $\varphi(\Sigma)$, которые логически следуют из множества Σ по правилам (4), (5). Кроме того, имеется алгоритм, позволяющий определить, является некоторая функциональная зависимость Σ следствием множества зависимостей Σ , или нет. Для анализа отношений с функциональными зависимостями все эти алгоритмы основаны на аппарате булевых функций [1. § 5.3].

Следующее ниже утверждение является простым средством декомпозиции отношений. В теории зависимостей оно называется теоремой Хеза [3. С. 293], ее доказательство приводится в [3. С. 316].

Утверждение 1. Теорема Хеза. Пусть R – некоторое отношение из $\text{Re}(T)$, а семейство множеств $\{X, Y, Z\}$ образует разбиение множества $\text{at}(R)$. Если $X \rightarrow Y$ в R , то $R = R[X \cup Y] \circ R[X \cup Z]$. Иными словами Теорема Хеза говорит о том, что функциональная зависимость $X \rightarrow Y$ в отношении $R\{X, Y, Z\}$ влечет для него зависимость соединения $^{\circ}[X \cup Y, X \cup Z]$. И далее вытекает следующее следствие 2:

пусть $R \in \text{Re}(T)$, $\text{at}(R) = X \cup Y$, $Y = \{y_1, \dots, y_k\}$. Если $X \rightarrow Y$ для R , то R удовлетворяет зависимости соединения $^{\circ}[X \cup y_1, X \cup y_2, \dots, X \cup y_k]$.

Чтобы показать справедливость указанного выше следствия достаточно указать на множество зависимостей $X \rightarrow y_1, X \rightarrow y_2, \dots, X \rightarrow y_k$, которые следуют из зависимости $X \rightarrow Y$ в силу свойств (4), (5).

Рассмотрим понятие многозначной зависимости [1. С. 211].

Пусть задано отношение $R \in \text{Re}(T)$, а X и Y – произвольные подмножества множества $\text{at}(R)$. Говорят, что X многозначно определяет Y , если $R = R[X \cup Y] \circ R[X \cup Z]$, где $Z = \text{at}(R) \setminus (X \cup Y)$. Если X многозначно определяет Y , то пишут $X \twoheadrightarrow Y$ и говорят, что отношение R удовлетворяет многозначной зависимости $X \twoheadrightarrow Y$ или, что многозначная зависимость $X \twoheadrightarrow Y$ выполняется в отношении R .

Приведенное определение показывает, что многозначная зависимость является частным случаем зависимости соединения и так же определяется через алгебраическое ограничение целостности [1].

Если $R \in \text{Re}(T)$ – произвольное отношение, то множество пар (X, Y) таких, что $X \rightarrow Y$ в R , обозначается через $\text{MVD}(R)$. Оно называется структурой многозначных зависимостей отношения R [1. С. 212].

Пусть R – произвольное отношение. В [1. С. 163, 212] доказано, что структура многозначных зависимостей $\text{MVD}(R)$ обладает следующими ниже свойствами (X, Y, Z, W – подмножества множества $\text{at}(R)$).

Если $X \supseteq Y$, то $X \rightarrow Y$. (6)

Если

$X \rightarrow Y$ и $Y \cup Z \rightarrow W$, то $X \cup Z \rightarrow W \setminus (Y \cup Z)$. (7)

Если $X \rightarrow Y$ и $X \rightarrow Z$, то $X \rightarrow Y \cup Z$. (8)

Если $X \rightarrow Y$, то $X \rightarrow \text{at}(R) \setminus (X \cup Y)$. (9)

Свойство (9) говорит о том, что многозначные зависимости всегда образуют связные пары, поэтому для отношения R с $\text{at}(R) = \{X, Y, Z\}$ и зависимостью $X \rightarrow Y$ используется обозначение $X \rightarrow Y|Z$. Как и для функциональных зависимостей, для дедуктивной системы, заданной правилами (6)–(9), алгоритмически разрешима проблема выводимости [1. § 5.2], поэтому анализ структур многозначных зависимостей также может осуществляться автоматически с применением булевых функций [1. § 5.3].

Согласно [1. С. 163] справедливо следующее утверждение.

Утверждение 3. Пусть $R \in \text{Re}(T)$ – произвольное отношение, а X и Y – произвольные подмножества множества $\text{at}(R)$, $Z = \text{at}(R) \setminus (X \cup Y)$. Отношение R удовлетворяет зависимости $X \rightarrow Y$ тогда и только тогда, когда

$$\sigma_{(X=A) \wedge (Z=C)}(R)[Y] = \sigma_{X=A}(R)[Y] \quad (10)$$

для каждого $\{X: A\} \in R[X]$ и для каждого $\{Z: C\} \in \sigma_{X=A}(R)[Z]$.

Равенство (10) показывает, что для каждого $\{X: A\}$ из $R[X]$ множество $\sigma_{(X=A) \wedge (Z=C)}(R)[Y]$ не зависит от $\{Z: C\} \in \sigma_{X=A}(R)[Z]$. Это означает, что многозначная зависимость $X \rightarrow Y$ в R определяет отображение $\{X: A\} \rightarrow \sigma_{(X=A) \wedge (Z=C)}(R)[Y]$ из $R[X]$ в булеан $\mathbf{P}(R[Y])$, которое можно понимать как многозначную функцию из $R[X]$ в $R[Y]$.

Утверждение 3 в менее формальной форме выбрано в качестве определения многозначной зависимости у Дейта [3. С. 329]. За исключением обозначений это определение выглядит следующим образом.

Пусть имеется некоторое отношение $R \in \text{Re}(T)$, $X, Y, Z \subseteq \text{at}(R)$ и $X \cup Y \cup Z = \text{at}(R)$, причем множества X, Y, Z попарно не пересекаются. Y многозначно зависит от X тогда и только тогда, когда множество значений Y , соответствующее заданной паре (значение X , значение Z) отношения R зависит только от X , но не зависит от Z . В то же время принятое в данной статье определение многозначной зависимости К.Дж. Дейта приводит как теорему Фейгина [3. С. 330], которая формулируется в терминах «тогда и только тогда».

В [1. С. 163] доказывается еще одно утверждение.

Утверждение 4. Пусть $R \in \text{Re}(T)$ – произвольное отношение, а X и Y – произвольные подмножества множества $\text{at}(R)$, $Z = \text{at}(R) \setminus (X \cup Y)$. Отношение R

удовлетворяет зависимости $X \rightarrow Y$ тогда и только тогда, когда

$$\sigma_{X=A}(R)[Y \cup Z] = \sigma_{X=A}(R)[Y] \circ \sigma_{X=A}(R)[Z] \quad (11)$$

для каждого $\{X: A\} \in R[X]$.

Заметим, что естественное соединение двух отношений, не имеющих общих атрибутов, можно рассматривать как расширенную версию декартова произведения. Равенство (11) выражает именно такой случай. По сути, утверждение 4 выражает еще один способ, которым можно определить многозначную зависимость. Такую форму определения многозначной зависимости можно найти в [4. С. 92].

Согласно утверждению 4 наличие многозначной зависимости $X \rightarrow Y$ в отношении $R[X, Y, Z]$ означает, что его атрибуты Y и Z являются взаимно независимыми: множество значений каждого из них зависит только от значения атрибута X . Иными словами, многозначные зависимости $X \rightarrow Y|Z$ «появляются тогда, когда для осмысленной группы характеристик X множества значений характеристик из групп Y и Z обязаны встречаться в любых сочетаниях при фиксированном значении X -компоненты» [1. С. 216].

Доказано [1. С. 163, 164], что каждая функциональная зависимость является многозначной зависимостью, т. е. если $X \rightarrow Y$ в R , то $X \rightarrow Y$ в R . Поэтому $\text{FD}(R) \subseteq \text{MVD}(R)$. Таким образом, многозначные зависимости являются обобщением функциональных зависимостей.

Однако между функциональными и многозначными зависимостями имеется существенное различие. Согласно (9) зависимости $X \rightarrow Y$ и $X \rightarrow \text{at}(R) \setminus (X \cup Y)$ всегда образуют связанные пары, что обозначается как $X \rightarrow Y|Z$, где $Z = \text{at}(R) \setminus (X \cup Y)$. Свойство (9) не имеет аналога для функциональных зависимостей, поэтому в общем случае они не образуют связанных пар. Например, зависимость $X \rightarrow Y$ влечет $X \rightarrow \text{at}(R) \setminus (X \cup Y)$; однако это совсем не означает, что зависимость $X \rightarrow \text{at}(R) \setminus (X \cup Y)$ тоже является функциональной.

Следующее ниже утверждение 5 обобщает утверждение 4. На этот раз используется понятие зависимости соединения.

Утверждение 5. Пусть R – некоторое отношение из $\text{Re}(T)$, а семейство множеств $\{X, Z_1, Z_2, \dots, Z_m\}$ образует разбиение множества $\text{at}(R)$. Отношение R удовлетворяет зависимости соединения $^\circ[X \cup Z_1, X \cup Z_2, \dots, X \cup Z_m]$ тогда и только тогда, когда

$$\begin{aligned} \sigma_{X=A}(R)[Z_1 \cup Z_2 \cup \dots \cup Z_m] = \\ = \sigma_{X=A}(R)[Z_1] \circ \sigma_{X=A}(R)[Z_2] \circ \dots \circ \sigma_{X=A}(R)[Z_m] \end{aligned} \quad (12)$$

для каждого $\{X: A\} \in R[X]$.

Утверждение 6. Обобщение теоремы Фейгина. Пусть R – некоторое отношение из $\text{Re}(T)$, а семейство множеств $\{X, Z_1, Z_2, \dots, Z_m\}$ образует разбиение множества $\text{at}(R)$. Отношение R удовлетворяет зависимости соединения $^\circ[X \cup Z_1, X \cup Z_2, \dots, X \cup Z_m]$ тогда и только тогда, когда оно одновременно удовлетворяет многозначным зависимостям $X \rightarrow Z_1, X \rightarrow Z_2, \dots, X \rightarrow Z_m$.

Доказательство. Необходимость. Для доказательства воспользуемся следующим вспомогатель-

ным понятием. Пусть семейство множеств $\{X_1, \dots, X_m\}$ является покрытием множества U , т.е. $U = \bigcup_{i=1}^m X_i$. На основе множеств $X_1, \dots, X_m$ образуем новые множества $Y_1, \dots, Y_l$ по формуле

$$Y_j = \bigcup_{s=1}^{l_j} X_{i_s}, \quad j = 1, \dots, l, \quad i = 1, \dots, m.$$

Если каждое множество X_i входит хотя бы в одно множество Y_j , то $U = \bigcup_{j=1}^l Y_j$. Покрытие $\{Y_1, \dots, Y_l\}$ называется укрупнением покрытия $\{X_1, \dots, X_m\}$ [1. С. 232].

Рассмотрим одно из свойств зависимостей соединения: из $^\circ[X_1, \dots, X_m]$ следует $^\circ[Y_1, \dots, Y_l]$ для любого укрупнения покрытия $\{X_1, \dots, X_m\}$.

Пусть $Z = \bigcup_{k=1}^m Z_k$. Согласно определению многозначная зависимость $X \rightarrow Z_k$, выполненная в R , равносильна зависимости соединения $^\circ[X \cup Z_k, X \cup (Z \setminus Z_k)]$. Очевидно, что для любого $k=1, \dots, m$ покрытие $\{X \cup Z_k, X \cup (Z \setminus Z_k)\}$ является укрупнением покрытия $\{X \cup Z_1, X \cup Z_2, \dots, X \cup Z_m\}$. Поэтому для отношения R любая зависимость $^\circ[X \cup Z_k, X \cup (Z \setminus Z_k)]$ является следствием зависимости $^\circ[X \cup Z_1, X \cup Z_2, \dots, X \cup Z_m]$.

Достаточность. Пусть $Z = \bigcup_{k=1}^m Z_k$, $Q_A = \sigma_{X=A}(R)[Z]$, $A \in R[X]$. Тогда с учетом свойств операции проекции равенство (13) можно представить в виде:

$$Q_A = Q_A[Z_1] \circ Q_A[Z_2] \circ \dots \circ Q_A[Z_m]. \quad (13)$$

В силу утверждения 5 выполнение (14) при каждом $A \in R[X]$ равносильно наличию в R зависимости $^\circ[X \cup Z_1, X \cup Z_2, \dots, X \cup Z_m]$.

Учитывая введенные обозначения, утверждение 4 гласит, что зависимость $X \rightarrow Z_k$ ($k=1, 2, \dots, m$) в отношении R равносильна выполнению равенства

$$Q_A = Q_A[Z_k] \circ Q_A[Z \setminus Z_k] \quad (14)$$

при любом $A \in R[X]$. Тогда множество зависимостей $X \rightarrow Z_1, X \rightarrow Z_2, \dots, X \rightarrow Z_m$ выполненных в R , равносильно тому, что для любого $A \in R[X]$ справедлива система, которая состоит из равенств (14) с различными значениями $k=1, 2, \dots, m$.

Для доказательства достаточности данного утверждения необходимо показать, что множество зависимостей $X \rightarrow Z_1, X \rightarrow Z_2, \dots, X \rightarrow Z_m$, имеющих в R , влечет зависимость $^\circ[X \cup Z_1, X \cup Z_2, \dots, X \cup Z_m]$. Иными словами, требуется доказать, что справедливость системы из m равенств

$$\begin{cases} Q_A = Q_A[Z_1] \circ Q_A[Z \setminus Z_1] \\ Q_A = Q_A[Z_2] \circ Q_A[Z \setminus Z_2] \\ \dots \\ Q_A = Q_A[Z_m] \circ Q_A[Z \setminus Z_m] \end{cases}$$

при каждом $A \in R[X]$ влечет справедливость равенства (14) при каждом $A \in R[X]$. Покажем это. Так как $\text{at}(Q_A) = Z_1 \cup Z_2 \cup \dots \cup Z_m$, то включение

$$Q_A \subseteq Q_A[Z_1] \circ Q_A[Z_2] \circ \dots \circ Q_A[Z_m]$$

выполняется всегда. Чтобы доказать равенство (14), надо показать, что обратное включение тоже справедливо.

Пусть

$$t = \{Z_1 : C_1, Z_2 : C_2, \dots, Z_m : C_m\} \in Q_A[Z_1] \circ Q_A[Z_2] \circ \dots \circ Q_A[Z_m].$$

Значит, отношение Q_A содержит кортежи

$$t' = \{Z_1 : C_1, Z_2 : C'_2, \dots, Z_m : C'_m\}, \\ t'' = \{Z_1 : C''_1, Z_2 : C_2, \dots, Z_m : C''_m\},$$

$$\dots \\ t^{(m)} = \{Z_1 : C_1^{(m)}, Z_2 : C_2^{(m)}, \dots, Z_m : C_m\}.$$

Пусть $t_1 = t'$. Последовательно образуем кортежи $t_2, t_3, \dots, t_m$ по формуле

$$t_k = t^{(k)} = [Z_k] \circ t_{k-1} [Z \setminus Z_k], \quad i=2, \dots, m. \quad (15)$$

Если $t_{k-1} \in Q_A$, то, согласно (14), t_k тоже принадлежит Q_A . Кортеж $t_1 \in Q_A$ по условию, поэтому все последующие кортежи — $t_2, t_3, \dots, t_m$ — тоже принадлежат отношению Q_A . В соответствии с (15) имеем

$$t_2 = \{Z_1 : C_1, Z_2 : C_2, Z_3 : C'_3, \dots, Z_m : C'_m\},$$

$$\dots \\ t_k = \{Z_1 : C_1, \dots, Z_k : C_k, Z_{k+1} : C_{k+1}^{(k-1)}, \dots, Z_m : C_m^{(k-1)}\},$$

$$\dots \\ t_m = \{Z_1 : C_1, Z_2 : C_2, \dots, Z_m : C_m\}.$$

Так как $t = t_m$, то $t_1 \in Q_A$. Следовательно, $Q_A[Z_1] \circ Q_A[Z_2] \circ \dots \circ Q_A[Z_m] \subseteq Q_A$ и равенство (13) установлено. Поскольку выбор $A \in R[X]$ никак не влияет на ход вышеизложенных рассуждений, то равенство (14) справедливо при любом $A \in R[X]$. Утверждение доказано.

Множество многозначных зависимостей $X \rightarrow Z_1, X \rightarrow Z_2, \dots, X \rightarrow Z_m$, которым удовлетворяет R с $\text{at}(R) = X \cup Z_1 \cup \dots \cup Z_m$, как и равносильную ему одну зависимость соединения $^\circ[X \cup Z_1, X \cup Z_2, \dots, X \cup Z_m]$, для удобства иногда будем обозначать через $X \rightarrow Z_1 | Z_2 | \dots | Z_m$.

Свойство многозначных зависимостей (9) позволяет представить зависимость $X \rightarrow Z_1 | Z_2 | \dots | Z_m$ в виде.

$$\left. \begin{aligned} X &\rightarrow Z_1 | Z_2 \cup Z_3 \cup \dots \cup Z_m, \\ X &\rightarrow Z_2 | Z_1 \cup Z_3 \cup \dots \cup Z_m, \\ &\dots \\ X &\rightarrow Z_m | Z_1 \cup Z_2 \cup \dots \cup Z_{m-1}. \end{aligned} \right\} \quad (16)$$

Если $m=2$, то множество (16) состоит из пары многозначных зависимостей; если $m>2$, — то из $2m$ зависимостей. На практике возможны как две, так и более многозначных зависимостей в выражении (16), что обеспечивает практическую ценность утверждению 6. Как и теореме Фейгина это утверждение можно использовать для приведения схемы отношения к четвертой нормальной форме при проектировании реляционной базы данных [3].

Согласно утверждению 4 многозначная зависимость $X \rightarrow Y | Z$, выполненная в отношении R с $\text{at}(R) = X \cup Y \cup Z$, выражает взаимную независимость атрибутов Y и Z в отношении R , притом что множество значений каждого из них определяется значением атрибута X . Точно так же утверждение 5 позволяет рассматривать зависимость $X \rightarrow Z_1 | Z_2 | \dots | Z_m$, выполненную в R , $\text{at}(R) = X \cup Z_1 \cup \dots \cup Z_m$: она выражает взаимную независимость атрибутов $Z_1, Z_2, \dots, Z_m$ друг

от друга, притом что множество значений каждого из них определяется значением атрибута X .

Согласно теореме Хеза функциональная зависимость выступает в роли достаточного условия для декомпозиции заданного отношения на две своих проекции. Утверждения 3 и 4 можно рассматривать как критерии наличия определенной многозначной зависимости в заданном отношении. Утверждение 5 является критерием наличия зависимости соединения определенного вида, а утверждение 6 устанавливает равносильность такой зависимости группе многозначных зависимостей с одинаковым детерминантом. Благодаря этому утверждение 5 можно рассматривать как обобщение утверждения 4 на случай группы многозначных зависимостей.

Сформулируем и докажем некоторые из рассмотренных утверждений в терминах подстановочной дизъюнктивно-конъюнктивной алгебры предикатов (ПДКАП) – для данной цели базиса этой алгебры достаточно. Для этого сначала необходимо сформулировать понятия функциональной, многозначной зависимости и зависимости соединения для предикатов.

Согласно [1. С. 162] функциональная зависимость аргумента x_n от аргументов $x_1, \dots, x_{n-1}$ в предикате $P(x_1, \dots, x_n)$ выражается следующей формулой:

$$\forall x_1 \forall x_2 \dots \forall x_n \forall x'_n (P(x_1, \dots, x_{n-1}, x_n) \wedge \wedge P(x_1, \dots, x_{n-1}, x'_n) \Rightarrow x_n = x'_n). \quad (17)$$

Любая функциональная зависимость одной группы переменных предиката $P(x_1, \dots, x_n)$ от другой группы может быть задана либо несколькими формулами вида (17), либо одной такой формулой, в которой после импликации будет стоять конъюнкция нескольких предикатов равенства. Если X и Y являются подмножествами переменных предиката $P(x_1, \dots, x_n)$, и Y функционально зависит от X , то будем писать $X \rightarrow Y$ и говорить, что предикат $P(x_1, \dots, x_n)$ удовлетворяет функциональной зависимости $X \rightarrow Y$.

Зависимость конъюнкции для предикатов была определена выражением (2). Она является аналогом зависимости соединения отношений. Многозначную зависимость для предикатов можно выразить через зависимость конъюнкции. Будем говорить, что предикат $P \in \text{Pre}(S)$ удовлетворяет многозначной зависимости $X \twoheadrightarrow Y$, если он удовлетворяет зависимости конъюнкции $\wedge\{X \cup Y, X \cup Z\}$, где $Z = V \setminus (X \cup Y)$. Ясно, что в этом случае так же будет выполняться зависимость $X \twoheadrightarrow Z$, что можно обозначить как $X \twoheadrightarrow Y|Z$.

Как и для реляционных отношений, для предикатов также можно задать структуры зависимостей. Пусть $P \in \text{Pre}(S)$, $X, Y \subseteq V$. Тогда множество пар (X, Y) , таких, что $X \rightarrow Y$ в P , обозначим через $\text{FD}(P)$ и назовем структурой функциональных зависимостей предиката P . Аналогично множество пар (X, Y) , таких, что $X \twoheadrightarrow Y$ в P , обозначим через $\text{MVD}(P)$ и назовем структурой многозначных зависимостей предиката P .

Судя по всему, множества $\text{FD}(P)$ и $\text{MVD}(P)$ будут обладать точно такими же свойствами, как структуры зависимостей для отношений. Но для проверки этого предположения требуется дополнительное исследование, которое выходит за рамки данной статьи.

Переходим к разработке предикатных утверждений о зависимостях.

Прежде всего, рассмотрим теорему Хеза, следствие 2 и утверждение 6. Так как в них отсутствуют алгебраические выражения, то их не нужно формулировать отдельно в терминах подстановочной дизъюнктивно-конъюнктивной алгебры предикатов. Так как практически очевидно, что зависимости в предикатах подчиняются тем же свойствам, что и зависимости в реляционных отношениях, то указанные три утверждения можно применять как к отношениям, так и к предикатам. Если доказать, что указанные свойства совпадают, то весь аппарат теории зависимостей можно будет обоснованно применять для анализа зависимостей в предикатах, что существенно расширит возможности декомпозиции предикатов.

Для определенности уточним, что означают теорема Хеза, следствие 2 и утверждение 6 по отношению к предикатам.

Рассмотрим теорему Хеза. Пусть $P \in \text{Pre}(S)$, $V = X \cup Y \cup Z$. Если P удовлетворяет функциональной зависимости $X \rightarrow Y$, то для него выполняется зависимость конъюнкции $\wedge\{X \cup Y, X \cup Z\}$.

Следствие 2 по отношению к предикатам означает следующее. Пусть $P \in \text{Pre}(S)$, $V = X \cup Y$, $Y = \{y_1, \dots, y_k\}$. Если $X \twoheadrightarrow Y$ для P , то P удовлетворяет зависимости конъюнкции $\wedge\{X \cup y_1, X \cup y_2, \dots, X \cup y_k\}$.

Пусть теперь $V = X_1 \cup \dots \cup X_m \cup Y$. Тогда на основании утверждения 6 зависимости конъюнкции $\wedge\{X_1 \cup Y, X_2 \cup Y, \dots, X_m \cup Y\}$ в предикате $P \in \text{Pre}(S)$ будем отождествлять с множеством многозначных зависимостей $Y \twoheadrightarrow X_1, Y \twoheadrightarrow X_2, \dots, Y \twoheadrightarrow X_m$ в этом предикате и условно обозначать $Y \twoheadrightarrow X_1|X_2|\dots|X_m$.

Утверждение 5 содержит алгебраическое выражение (12), поэтому его следует переформулировать в терминах ПДКАП и доказать.

Для удобства в дальнейшем изменим обозначения в условии утверждения 5 и сформулируем его следующим образом. Пусть семейство множеств $\{X_1, X_2, \dots, X_m, Y\}$ образует разбиение множества $\text{at}(R)$ для некоторого $R \in \text{Re}(T)$. Отношение R удовлетворяет зависимости соединения $^\circ[X_1 \cup Y, X_2 \cup Y, \dots, X_m \cup Y]$ тогда и только тогда, когда

$$\sigma_{Y=A}(R)[X_1 \cup X_2 \cup \dots \cup X_m] = \sigma_{Y=A}(R)[X_1] \circ \sigma_{Y=A}(R)[X_2] \circ \dots \circ \sigma_{Y=A}(R)[X_m] \quad (18)$$

для каждого $\{Y, A\} \in R[Y]$.

Введем несколько удобных обозначений. Пусть задано произвольное разбиение $\Gamma = \{X_1, X_2, \dots, X_k\}$ множества $V = \{x_1, x_2, \dots, x_n\}$, $k \leq n$. Иногда вместо каждой переменной в отдельности $x_1, x_2, \dots, x_n$ бывает удобно рассматривать непересекающиеся группы переменных $X_1, X_2, \dots, X_k$ – слои разбиения множества V . В таких случаях вместо обычной записи $P(x_1, x_2, \dots, x_n)$ будет применяться условное обозначение $P(X_1, X_2, \dots, X_k)$.

Пусть $T \subseteq V \times U$ – осто́в с конечным множеством имен переменных V . Пространство S задано как декартово произведение областей изменения переменных из V . Если $X = \{x_1, \dots, x_k\} \subseteq V$, то через $\text{Dom} X$ будет обозначаться произведение $\text{Dom} x_1 \times \dots \times \text{Dom} x_k$ [1. С. 131]. Пусть P – произвольный предикат из $\text{Pre}(S)$.

Переходим к утверждению 5. Зависимость соединения $[X_1 \cup Y, X_2 \cup Y, \dots, X_m \cup Y]$ перейдет в зависимость конъюнкции $\wedge \{X_1 \cup Y, X_2 \cup Y, \dots, X_m \cup Y\}$, которая выражается равенством

$$P = \exists V \setminus (X_1 \cup Y)(P) \wedge \exists V \setminus (X_2 \cup Y)(P) \wedge \dots \wedge \exists V \setminus (X_m \cup Y)(P). \quad (19)$$

Равенство (18) примет вид:

$$\exists Y(P \wedge Y^A) = \exists V \setminus X_1(P \wedge Y^A) \wedge \exists V \setminus X_2(P \wedge Y^A) \wedge \dots \wedge \exists V \setminus X_m(P \wedge Y^A). \quad (20)$$

Представим выражения (19) и (20) в более удобной форме. Пусть $X_1, X_2, \dots, X_m \subseteq V$, введем обозначение

$$\exists(X_1 \cup X_2 \cup \dots \cup X_m)(P) = \exists X_1 \exists X_2 \dots \exists X_m(P), \\ P \in \text{Pre}(S).$$

Перепишем сомножители из (20) с помощью операции подстановки. Пользуясь тождеством $Y/A(P) \equiv \exists Y(P \wedge Y^A)$, $Y \subseteq V$, $Y \subseteq V$, $P \in \text{Pre}(S)$, имеем:

$$\begin{aligned} \exists V \setminus X_i(P \wedge Y^A) &= \\ &= \exists X_1 \dots \exists X_{i-1} \exists X_{i+1} \dots \exists X_m \exists Y(P \wedge Y^A) = \\ &= \exists X_1 \dots \exists X_{i-1} \exists X_{i+1} \dots \exists X_m (\exists Y(P \wedge Y^A)) = \\ &= \exists X_1 \dots \exists X_{i-1} \exists X_{i+1} \dots \exists X_m (Y/A(P)), \quad i = 1, \dots, m. \end{aligned}$$

Теперь сформулируем и докажем утверждение 5 для предикатов.

Утверждение 7. Пусть семейство множеств $\{X_1, X_2, \dots, X_m, Y\}$ образует разбиение множества V . Предикат $P \in \text{Pre}(S)$ удовлетворяет зависимости конъюнкции $\wedge \{X_1 \cup Y, X_2 \cup Y, \dots, X_m \cup Y\}$, т. е.

$$P = (\exists X_2 \exists X_3 \dots \exists X_m(P)) \wedge (\exists X_1 \exists X_3 \dots \exists X_m(P)) \wedge \dots \wedge (\exists X_1 \exists X_2 \dots \exists X_{m-1}(P)) \quad (21)$$

тогда и только тогда, когда

$$\begin{aligned} Y/A(P) &= (\exists X_2 \exists X_3 \dots \exists X_m(Y/A(P))) \wedge \\ &\wedge (\exists X_1 \exists X_3 \dots \exists X_m(Y/A(P))) \wedge \dots \wedge \\ &\wedge (\exists X_1 \exists X_2 \dots \exists X_{m-1}(Y/A(P))), \quad (22) \end{aligned}$$

для каждого $A \in P_T[Y] \subseteq \text{Dom} Y$, где P_T – область истинности предиката P .

Тогда условие (22) примет вид:

$$Y/A(P) = Y/A((\exists X_2 \exists X_3 \dots \exists X_m(P)) \wedge \dots \wedge (\exists X_1 \exists X_2 \dots \exists X_{m-1}(P))),$$

это равенство справедливо при любом $A \in P_T[Y]$, если выполняется (21).

Утверждение 8. Пусть семейство множеств $\{X_1, X_2, \dots, X_m, Y\}$ образует разбиение множества V . Предикат $P \in \text{Pre}(S)$ удовлетворяет зависимости $\wedge \{X_1 \cup Y, X_2 \cup Y, \dots, X_m \cup Y\}$ тогда и только тогда, когда для

каждого $A \in P_T[Y]$ (P_T – область истинности предиката P) найдутся такие предикаты $P_1^A(X_1), \dots, P_m^A(X_m) \in \text{Pre}(S)$, что подстановку $Y/A(P) = P(X_1, X_2, \dots, X_m, A)$ можно представить конъюнкцией

$$P(X_1, \dots, X_m, A) = P_1^A(X_1) \wedge P_2^A(X_2) \wedge \dots \wedge P_m^A(X_m). \quad (23)$$

Чтобы с помощью утверждения 8 проверить, удовлетворяет ли предикат зависимости $\wedge \{X_1 \cup Y, X_2 \cup Y, \dots, X_m \cup Y\}$, выполним дизъюнктивное разложение предиката P по переменным Y :

$$P(X_1, \dots, X_m, A) = \bigvee_{A \in P_T[Y]} Y^A P(X_1, \dots, X_m, A). \quad (24)$$

Формула любого предиката $P(X_1, \dots, X_m, A)$, $A \in P_T[Y]$ обычно намного меньше, чем формула исходного предиката $P(X_1, \dots, X_m, Y)$. Поэтому после дизъюнктивного разложения легко проверить: можно представить каждый предикат $P(X_1, \dots, X_m, A)$ конъюнкцией (23) или нет. Для сравнительно небольших формул это довольно легко определить по их внешнему виду.

Следует заметить, что равенство (24) не является дизъюнктивным разложением предиката в том виде, который предложен в [2. С. 28], и поэтому правомерность такого разложения еще надо доказать.

Утверждение 9. Модифицированная теорема о дизъюнктивном разложении предиката. Любой конечный предикат $P \in \text{Pre}(S)$ может быть представлен в виде

$$P(x_1, x_2, \dots, x_n) = \bigvee_{(a_1, a_2, \dots, a_m) \in N} x_1^{a_1} x_2^{a_2} \dots x_m^{a_m} P(a_1, a_2, \dots, a_m, x_{m+1}, x_{m+2}, \dots, x_n). \quad (25)$$

Таким образом, для правильного представления предиката $P \in \text{Pre}(S)$ с помощью дизъюнктивного разложения логическое суммирование можно проводить не по всей области M , а по ее подмножеству N . В дальнейшем при выполнении дизъюнктивного разложения предикатов всегда будет использоваться выражение (25), а кортежи множества N могут быть проиндексированы (в силу конечности множества $N = P_T[x_1, x_2, \dots, x_m]$, которое вытекает из конечности множества P_T).

Следует заметить, что для конъюнктивного разложения [2. С. 36] подобная модификация будет неверна.

Основным методом познания интеллекта считают алгебраизацию логики, т. е. точное формальное описание той алгебры, которая реализована в интеллекте и, как следствие, проявляет себя в естественном языке. В результате такого описания была разработана алгебра предикатов и алгебра предикатных операций. Первая потенциально может описывать мысли, вторая – действия над мыслями, т. е. мышление. Схемная реализация формул этих алгебр приводит к инженерным решениям, которые называются логическими сетями. Логические сети ориентированы на параллельные вычисления и, по мнению создателей, призваны стать основой мозгоподобных компьютеров.

Однако неадекватность тьюрингоподобной или, другими словами, последовательной природы персональных компьютеров феномену естественного языка была осознана уже давно. В поисках новых аппаратных решений для моделирования естественного языка и других функций интеллекта средствами алгебры конечных предикатов в 1980-е гг. разрабатывались специальные технические средства — переключательные цепи. Эти средства были предложены для эффективного решения уравнений алгебры предикатов. В начале 2000-х гг. на смену переключательным цепям первого и второго рода пришла новая более прогрессивная концепция логических сетей, которая основана на идеях теории категорий [5–7].

Логические сети способны стать той аппаратной базой, которая позволит в реальном режиме времени обрабатывать логические уравнения большого объема, описывающие полноценные модели естественного языка (на что последовательные компьютеры не способны в принципе). Развитие метода декомпозиции предикатов основано на утверждениях о функциональных, многозначных зависимостях и зависимости соединения реляционных отношений, теоремах Хеза и Фейгина и утверждениях о предикатных зависимостях. Метод позволяет обоснованно применить аппарат теории зависимостей для анализа зависимостей в предикатах и существенным образом расширить возможности декомпозиции предикатов. Опираясь

на связь между реляционной алгеброй и алгеброй предикатов, утверждения о зависимостях преобразованы в новые средства декомпозиции предикатов. С помощью полученных средств декомпозиции разработан метод бинарной декомпозиции функциональных предикатов, который является мощным инструментом декомпозиции предикатов. Непосредственными операторами декомпозиции в данном методе являются кванторы существования [5].

Выводы

1. Выявлены основные виды зависимостей атрибутов в реляционных отношениях: функциональные, многозначные и зависимости соединения.
2. Рассмотрены утверждения о зависимостях, позволяющие выполнять декомпозицию реляционных отношений.
3. Сформулированы и доказаны обобщающие утверждения о зависимостях соединений особого вида, которые усовершенствовали теорию нормализации отношений.
4. Разработаны новые критерии проверки предикатов на выполнение зависимости конъюнкции, что позволяет выполнять декомпозицию предикатов.
5. Выполнена модификация теоремы о дизъюнктивном разложении предиката на случай бесконечного предметного пространства.

СПИСОК ЛИТЕРАТУРЫ

1. Цаленко М.Ш. Моделирование семантики в базах данных. — М.: Наука, 1989. — 288 с.
2. Шабанов-Кушнаренко Ю.П. Теория интеллекта. Математические средства. — Харьков: Вища шк. Изд-во при Харьк. ун-те, 1984. — 144 с.
3. Дейт К.Дж. Введение в системы баз данных / пер. с англ. и ред. К.А. Птицына. — М.: Издательский дом «Вильямс», 2008. — 1327 с.
4. Мишенин А.И. Теория экономических информационных систем. — М.: Финансы и статистика, 2008. — 237 с.
5. Рудометкина М.Н. Разработка метода бинарной декомпозиции функциональных предикатов // Известия Томского политехнического университета. — 2012. — Т. 320. — № 5. — С. 5–11.
6. Бондаренко М.Ф., Чикина В.А., Шабанов-Кушнаренко Ю.П. Модели языка // Бионика интеллекта. — 2004. — № 1 (61). — С. 27–37.
7. Bondarenko M.F., Hahanova I.V. Logic networks application for computing process organization // Радиоэлектроника и информатика. — 2003. — № 3. — С. 150–156.

Поступила 31.01.2013 г.

УДК 004.415

МЕТОД КОНТРОЛЯ ПРЯМОГО ДОСТУПА К СЕМАНТИЧЕСКИМ БАЗАМ ДАННЫХ

Хоанг Ван Куэт, А.Ф. Тузовский

Томский политехнический университет
E-mail: student8050@sibmail.com

Введены основные понятия, методы определения и создания меток безопасности триплетов семантических данных. Разработаны алгоритмы для формирования покрытия безопасности семантических данных и обеспечения безопасности прямого доступа пользователей к базам данных.

Ключевые слова:

Метка безопасности, покрытие безопасности, несанкционированный доступ.

Key words:

Security label, security cover, unauthorized access.

В связи с активным развитием такого нового направления информатики, как семантические технологии, создаются большие объёмы семантических данных, которые отличаются от реляционных данных тем, что на их основе могут выполняться логические выводы, позволяющие формировать новую информацию [1]. Использование таких данных в работе информационных систем требует решения следующих задач обеспечения их безопасности:

- 1) контроль прямого доступа пользователей к различным частям данных, определённых политикой доступа к данным для каждого пользователя (управление прямым доступом к данным);
- 2) контроль логических выводов на доступных данных, на основе которых могут быть логически выведены данные, доступ к которым пользователю не разрешён (управление возможными логическими выводами).

В данной статье рассмотрены метод определения меток безопасности RDF-триплетов данных (RDF – *Resource Description Framework*) и метод определения уровней безопасности RDF-экземпляров данных семантических баз данных для решения задачи контроля прямого доступа.

Постановка задачи контроля прямого доступа к семантическим базам данных

Угроза безопасности информации – это потенциально возможное воздействие на информацию, которое прямо или косвенно может нанести урон пользователям или владельцам информации. При классификации угроз выделяют три основных свойства безопасности информации: конфиденциальность, целостность, доступность [2]. В соответствии с этими свойствами безопасности информации различают три классические угрозы безопасности информации:

- Угроза конфиденциальности информации состоит в нарушении установленных ограничений на доступ к информации.
- Угроза целостности информации – это несанкционированное изменение информации, случайное или преднамеренное.

- Угроза доступности информации осуществляется, когда несанкционированно блокируется доступ к информации (блокирование может быть постоянным или на некоторое время, которое достаточно, чтобы информация стала бесполезной).

В целях обеспечения безопасности работы информации определены основные виды политик: дискреционная политика безопасности, политика безопасности информационных потоков, политика ролевого разграничения доступа, мандатная политика безопасности и политика изолированной программной среды [3].

Основная цель мандатной политики безопасности – предотвращение утечки информации от объектов с высоким уровнем доступа к объектам с низким уровнем доступа, т. е. противодействие возникновению в информационной системе неблагоприятных информационных потоков сверху вниз. Чаще всего мандатную политику безопасности описывают в терминах, понятиях и определениях свойств модели Белла Ла Падула. По сравнению с информационными системами, построенными на основе дискреционной политики безопасности, для систем, реализующих мандатную политику, характерна более высокая степень надёжности. Правила мандатной политики безопасности более ясны и просты для понимания разработчиками и пользователями, что также является фактором, положительно влияющим на уровень безопасности системы.

В семантических данных для контроля прямого доступа пользователей к различным частям данных принимается модель системы мандатного разграничения доступа с помощью использования меток безопасности RDF-триплетов и уровней безопасности RDF-экземпляров данных семантических баз данных. Для достижения поставленной цели необходимо описать основные понятия семантических данных.

Определение 1 (язык описания ресурсов). RDF-язык – это разработанная организацией W3C модель для описания метаданных о ресурсах. В основе данной модели лежит идея об использовании

специального вида утверждений, высказываемых о ресурсе. Каждое утверждение имеет вид «субъект–предикат–объект» и в терминологии RDF называется триплетом.

Определение 2 (модель RDF-триплета). Триплет является основным элементом семантических баз данных и состоит из набора троек: субъект (r), предикат (p), объект (v), где каждый компонент является константой, такой как $r \in R$ (набор ресурсов), $p \in PR$ (набор свойств) и $v \in R \cup L$ (набор ресурсов и литералов), или переменной, представленной в виде символов \$ или ?.

Определение 3 (RDF-схема, RDFS). Язык RDFS является семантическим расширением языка RDF и предоставляет средства для описания групп связанных ресурсов и отношений между этими ресурсами. Все определения языка RDFS выражены с помощью средств языка RDF.

Определение 4 (язык описания онтологий – Web Ontology Language (OWL)). OWL – это язык представления онтологий в Web-сети. Фактически это словарь, расширяющий набор терминов, определённых на языке RDFS.

Определение 5 (семантическая база данных). Семантическая база данных состоит из множеств данных, описанных на языках RDF, RDFS, OWL, и представляет собой множество триплетов. Она включает в себя набор ресурсов R , свойств PR , URI ссылок U , пустых узлов B и литералов L . В семантических базах данных выражения $R=U \cup B$, $PR \subset U$ и P , B и L не пересекаются.

Модель контроля прямого доступа пользователей к семантическим данным

Предлагаемая модель построена на основе мандатной политики безопасности, основанной на мандатном разграничении доступа, определяемом следующими четырьмя условиями:

- Все пользователи и триплеты базы данных однозначно идентифицированы.
- Задана решётка уровней безопасности информации.
- Каждому триплету базы данных присвоен уровень безопасности, определяющий ценность содержащейся в нем информации.
- Каждому пользователю присвоен уровень доступа, определяющий уровень доверия к нему в семантической базе данных.

В семантической базе данных каждый триплет имеет свой уровень безопасности AC , указатель которого может иметь значения из множества меток безопасности $L = \{\text{неклассифицированный } (L_U), \text{конфиденциальный } (L_C), \text{секретный } (L_S) \text{ и сверхсекретный } (L_{TS})\}$, где $L_U < L_C < L_S < L_{TS}$. Каждый пользователь имеет уровень доступа AC_s , включающий права выполнения таких операций с данными, как $\text{Rule} = \{\text{read}, \text{write}, \text{append}, \text{execute}\}$, где read – право на чтение триплета, write – права на запись триплета, append – права на запись в конец объекта, execute – права на выполнение добавления или удаления триплета.

В соответствии с мандатной моделью, безопасность базы данных должна обладать следующими свойствами:

- **Свойство простой безопасности («простое» свойство):** Пользователь может иметь право доступа на чтение триплета только в случае, когда уровень доступа пользователя не ниже уровня безопасности триплета.
- **Свойство «звезда» безопасности (свойство «звезда»):** Пользователь может иметь доступ к триплету в случае, когда уровень безопасности триплета не ниже его уровня права доступа. Пользователь может иметь право доступа на запись триплета только в случае, когда его уровень доступа равен уровню безопасности триплета. Он может иметь право доступа на чтение триплета только в случае, когда его уровень доступа не ниже уровня безопасности триплета.
- **Свойство дискреционной безопасности (ds-свойство):** При каждом обращении предоставляется одно право доступа пользователя на триплет.
- **Невозможность полного общения с бездействующим триплетом:** Пользователь не может читать бездействующий триплет.

В связи с этим для семантических баз данных предлагаются следующие основные свойства поддержки безопасности работы с ними:

- **Свойство чтения.** Пользователь может читать триплет только в том случае, когда его уровень доступа не ниже уровня безопасности триплета.
- **Свойство записи.** Только владелец имеет право на запись (модификацию) его данных.
- **Уровень безопасности триплета.** Уровень безопасности триплета должен быть не ниже уровня доступа его владельца.
- **Уровень безопасности элементов триплета.** Уровень безопасности триплета должен быть не ниже уровня безопасности его субъекта, объекта и предиката.
- **Уровень доступа на модификацию триплетов.** Только администратор безопасности базы данных и владелец триплетов имеют право на модификацию уровня безопасности триплетов и уровней доступа пользователей.

Алгоритмы решения задачи

Для решения поставленной задачи разработаны два алгоритма, позволяющие определить уровень безопасности RDF-экземпляров данных и контролировать доступ пользователей к базе данных.

Метка безопасности RDF-триплета данных

В целях безопасности семантических баз данных для каждого компонента триплета могут быть заданы разные права доступа, т. е. они находятся на разных уровнях безопасности согласно своим меткам $(sl_1, sl_2, \dots, sl_n)$, созданным пользователями [4]. Например, sl_1 соответствует открытым данным, sl_2 – конфиденциальным данным, ..., sl_n – сверхсекретным данным.

В триплете (субъект, предикат, объект) каждый из компонентов может иметь одинаковую метку sl

или разные метки безопасности (sl_r, sl_p, sl_v) (где sl_r – метка для субъекта, sl_p – метка для предиката, sl_v – метка для объекта). В том случае, когда три элемента имеют одинаковую метку sl , то sl является меткой безопасности для триплета ($sl=sl_r$, где sl_r – метка для триплета). Если три метки являются разными ($sl_r \neq sl_p \neq sl_v$), то надо образовать наименьшую верхнюю границу значения $sl_{\max}$ этих меток, тогда $sl_{\max}$ является меткой безопасности триплета sl_r ($sl_r=sl_{\max}$).

Один и тот же ресурс может являться субъектом или объектом (иногда является предикатом) в разных ситуациях, следовательно, он может иметь разные метки безопасности ($sl_1, sl_2, \dots, sl_n$) в зависимости от своего положения. Триплет, обладающий ресурсом в конкретной роли (субъект или объект), должен иметь конкретную метку sl , охватывающую соответствующую метку обеспечения ресурса в данной позиции, и обозначаться $sl \geq \max(sl_1, sl_2, \dots, sl_n)$.

Алгоритм 1: создание покрытия безопасности триплетов семантических данных

Определение 6. (модель отображения). Пусть $t=[r,p,v]$ и $t'=[r',p',v']$ являются моделями RDF-триплетов. Модель отображения $f: t \rightarrow t'$, где r, p, v и r', p', v' являются константами или переменными, определяется следующим образом:

- f отображает одну переменную в другую переменную или в константу;
- f отображает константу e , находящуюся в D_T (экземпляр данных), в виде такой же константы;
- f отображает константу e , находящуюся в схеме S_T (схема данных):
 - 1) в такую же схему констант;
 - 2) другие схемы констант e , подчиняющиеся следующим формам:
 - $[e, rdf: type, e]$;
 - $[e, rdfs: subPropertyOf, e]$;
 - $[e, rdfs: subClassOf, e]$;
 - $[f(r), f(x), f(v)]=[r', p', v']$.

Определение 7 (модель категоризации триплетов). Пусть $t=[r,p,v]$ и $t'=[r',p',v']$ являются моделями RDF-триплетов. Выражение « t задаёт категорию (категоризирует) t' » обозначается как $t \leq t'$, если существует функция $f: t \rightarrow t'$. Модель категоризации триплетов является рефлексивной, транзитивной или антисимметричной.

В семантических базах данных каждая модель триплета связана с одной меткой безопасности.

Определение 8 (безопасность RDF-триплета). Пусть S_T является RDF-схемой данных, D_T является RDF-экземплярами и элементы S_T и D_T представляются в виде набора триплетов. Пусть множество всех триплетов в базе данных обозначается как K_T , где $K_T=S_T \cup D_T$. Значение безопасности RDF-триплета определяется как пара $s=(t, sl)$, где $t \in K_T$, $sl \in S_L$, а S_L является множеством меток безопасности семантических баз данных $\{sl_1, sl_2, \dots, sl_n\}$.

Определение 9 (присвоенная метка безопасности). Пусть pt и t являются триплетами. Если существует модель отображения $f: pt \rightarrow t$ и pt имеет мет-

кую безопасность sl , то безопасность RDF-триплета будет задаваться как (t, sl) .

Определение 10 (покрытие безопасности для RDF данных). Предполагается, что S является множеством значений безопасностей триплетов семантических данных, где $S=\{s_1, s_2, \dots, s_n\}$, где $s_i=(t_i, sl_i)$, для любой безопасности $s=(t, sl)$ выполняются условия:

1. Не существуют два значения безопасности триплетов (t, sl) и (t', sl') , удовлетворяющие выражениям: $t=t'$ и $sl > sl'$.
2. Не существует значения безопасности $s=(t, sl)$, в котором sl является пустой.

Определение 11 (политика безопасности для RDF данных). Предполагается, что SP является множеством безопасностей RDF-триплетов вида (pt, sl) , где pt является моделью RDF-триплета, и sl – меткой безопасности. Политикой безопасности для RDF-данных является $SP \cup pt_{def}$, где $pt_{def}=[(?x_1, ?x_2, ?x_3), sl_{def}]$, x_1, x_2, x_3 являются элементами триплета и sl_{def} является меткой безопасности по умолчанию. Для любых двух пар (t, sl) и (t', sl') , где $t' \leq t$, выражение $sl \leq sl'$ выполняется только в случае, когда операция $\leq$ является доминирующим отношением.

Политика безопасности определяет требования безопасности для примитивных RDF/RDFS утверждений в базе данных и наследованных утверждений из RDF/RDFS-базы данных. Кроме того, политика безопасности по умолчанию также гарантирует, что не существует утверждения RDF/RDFS, которое является неклассифицированным [5].

Таблица. Модель контроля доступа пользователей

Модель безопасности	Интерпретация
$[r, p, v]$	Все элементы триплета являются константами
$[r, ?x, v]$	Субъект и объект являются константами, а предикат – переменной
$[r, p, ?x]$	Субъект и предикат являются константами, а объект – переменной
$[?x, p, v]$	Объект и предикат являются константами, а субъект – переменной
$[r, ?x, ?y]$	Объект и предикат являются переменными, а субъект – константой
$[?x, p, ?y]$	Субъект и объект являются переменными, а предикат – константой
$[?x, ?y, v]$	Субъект и предикат являются переменными, а объект – константой
$[?x, ?y, ?z]$	Все элементы являются переменными

Покрытием безопасности S является множество всех значений безопасностей триплетов RDF-данных $S=\{s_1, s_2, \dots, s_n\}$, создающихся путём применения модели отображения и присвоения меток безопасности. На рис. 1 показан алгоритм для создания покрытия безопасности триплетов семантических данных, обладающего следующими свойствами:

- если существует модель отображения от модели к триплету, то метка безопасности триплета доминирует над меткой безопасности модели;
- если триплет имеет метку, не созданную с помощью модели по умолчанию, то должна суще-

ствовать такая модель отображения из модели к триплету, у которой метка безопасности модели доминирует над меткой безопасности триплета.

Данный алгоритм применяется для моделей триплетов, определённых в таблице, где любой элемент из триплета $[r, p, v]$ может быть переменной или константой. В соответствии с данным алгоритмом, входными данными алгоритма являются политика безопасности $SP = \{sp_1, \dots, sp_n\} \cup ([x_1, x_2, x_3], sl_{def})$, где $sp_i = (pt_i, sl_i)$, и RDF база данных $K_r = D_r \cup S_r = \{t_i, \dots, t_k\}$. Выходными данными является покрытие безопасности $S = \{s_1, s_2, \dots, s_k\}$, где $s_i = (t_i, sl_i)$, $i = 1, \dots, k$.

Данный алгоритм гарантирует, что все триплеты семантических баз данных получают свои метки безопасности.

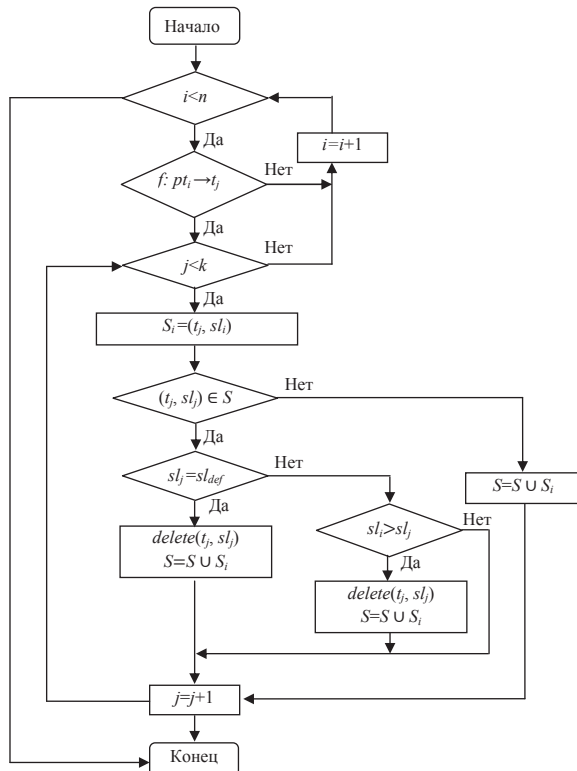


Рис. 1. Алгоритм определения покрытия безопасности триплетов семантических данных

Алгоритм 2: контроль прямого доступа пользователей к базе данных

Под контролем прямого доступа пользователей понимается возможность получать данные в соответствии с заданными для них правами на использование данных.

В семантических базах данных хранятся разные категории информации, такие как экономическая, финансовая и т. п. [6]. Если уровень безопасности информации AC содержит только один указатель с меткой безопасности из множества L , то пользователь, имеющий уровень доступа типа $AC_s = L_c$ в категории M , не может иметь доступ в категории N к данным, имеющим уровень безопасности $AC = L_c$. В

данном случае количество возможных вариантов создания уровней безопасности информации в базе данных будет сильно ограниченным. Для решения данной проблемы в уровни безопасности данных могут быть добавлены другие указатели с метками безопасности из множества L .

Предполагается, что метка безопасности RDF-триплета $AC = (S, P, PS, C)$ (уровень безопасности триплета) включает следующие 4 показателя:

- *чувствительность* S определяет уровень значимости или важности связи (предиката), а также её уязвимости перед несанкционированным лицом;
- *приватность* P определяет права владельца на возможность передачи данной информации другим пользователям;
- *персональная безопасность* PS определяет уровень защиты персональной информации человека или организации;
- *конфиденциальность* C задаёт возможность совместного использования данной информации с другими ресурсами.

Каждый показатель может принимать значения из множества L (в данной работе предлагается, что эти показатели могут принимать значение 0 (unclassified), 1 (confidential)).

Право доступа пользователей к данным $AC_s = (S_s, P_s, PS_s, C_s)$ также включает 4 показателя: чувствительность S , приватность P , персональная безопасность PS и конфиденциальность C . Для контроля доступа пользователя к RDF-триплету необходимо сравнить уровень права доступа пользователей $AC_s = (S_s, P_s, PS_s, C_s)$ с уровнем безопасности триплета $AC = (S, P, PS, C)$.

Считается, что $AC_s = (S_s, P_s, PS_s, C_s)$ не ниже уровня $AC = (S, P, PS, C)$ только в случае, когда $S_s \geq S$, $P_s \geq P$, $PS_s \geq PS$, $C_s \geq C$. Если $AC_s \geq AC$, то пользователи могут получить доступ к этим данным. Если $AC_s < AC$, то пользователи не имеют права доступа к этим данным и данные являются невидимыми для пользователей. На рис. 2 показан алгоритм контроля прямого доступа к RDF-триплету данных.

В соответствии с этим алгоритмом проверка прав доступа пользователя осуществляется на каждом триплете. Входными данными алгоритма являются: уровень права доступа пользователя и множество индексов чувствительности RDF-триплетов данных. Выходными данными является множество всех триплетов, к которым пользователь имеет право доступа. На каждом шаге происходит процесс сравнения каждого критерия индекса чувствительности с уровнем права доступа пользователя.

Данный алгоритм разработан на основе традиционных алгоритмов управления доступом пользователей к данным. Его достоинствами являются простота и небольшое количество операций, а также возможность обеспечения безопасности каждого триплета данных (как столбца в реляционных базах данных), что позволяет контролировать доступ пользователей к различным частям данных.

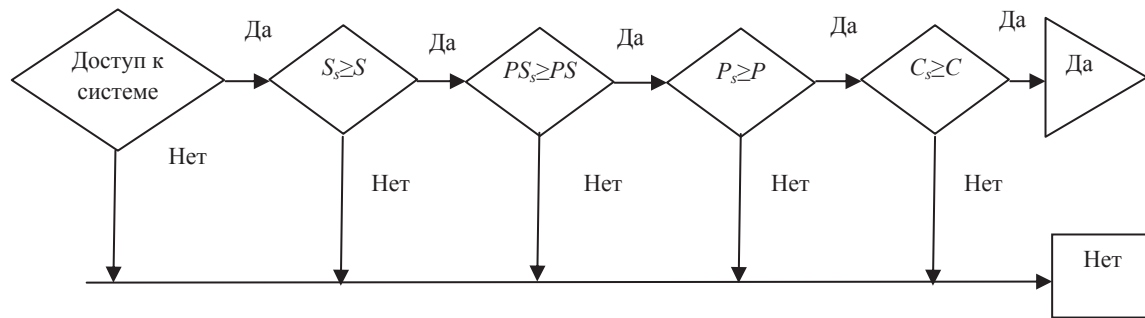


Рис. 2. Алгоритм контроля прямого доступа пользователей к базе данных

Заключение

Задача обеспечения безопасности семантических баз данных является более сложной, чем обеспечение безопасности реляционных баз данных, для которых уже разработано большое количество методов. Сложность обеспечения безопасности семантических баз данных заключается в требовании контроля доступа пользователей к разным частям данных (к каждому триплету).

Данная задача решается путём применения вышеописанной модели мандатного разграничения доступа с использованием алгоритмов определения меток безопасности RDF-триплетов данных и уровней безопасности семантических данных, а также алгоритма контроля доступа пользователя к базе данных.

СПИСОК ЛИТЕРАТУРЫ

1. Reddivari P. Policy based Access Control for a RDF Store // *Proceedings of the Policy Management for the Web Workshop*. – 2005. – V. 120. – № 5. – P. 78–83.
2. Грушо А.А., Тимонина Е.Е. Теоретические основы защиты информации. – М.: Изд-во Агентства «Яхтсмен», 1996. – 250 с.
3. Носов В.А. Основы теории алгоритмов и анализа их сложности. – М.: МГУ, 1992. – 268 с.
4. Stachour P. Design of LDV: A multilevel secure relational database management system *IEEE Trans // Knowledge and Data*. – 1990. – V. 2. – № 2. – P. 190–209.
5. Тузовский А.Ф., Ямпольский С.В. Системы управления знаниями (методы и технологии) / под общ. ред. В.З. Ямпольского. – Томск: Изд-во НТЛ, 2005. – 260 с.
6. Девянин П.Н. Анализ безопасности управления доступом и информационными потоками в компьютерных системах. – М.: Радио и связь, 2006. – 176 с.

Поступила 08.10.2012 г.

УДК 004.912 519.689.3

РАЗРАБОТКА ЛИНГВИСТИЧЕСКОГО ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ НА ОСНОВЕ ОНТОЛОГИЧЕСКИХ МОДЕЛЕЙ ЗНАНИЙ

Е.А. Сидорова

Институт систем информатики им. А.П. Ершова СО РАН, г. Новосибирск
Новосибирский государственный университет
E-mail: lena@iis.nsk.su

Описывается онтологическая модель знаний, ориентированная на автоматический анализ текстов в ограниченной предметной области. Рассматриваются лингвистические ресурсы и инструменты, необходимые для разработки лингвистического обеспечения информационных систем.

Ключевые слова:

Компьютерная лингвистика, онтология предметной области, лингвистическая онтология, тезаурус, лингвистические ресурсы, анализ текста.

Key words:

Computational linguistics, ontology of subject domain, linguistic ontology, thesaurus, linguistic resources, text analysis.

Введение

Разработка новых методов и моделей представления знаний является основным направлением развития современных информационных систем (ИС). С этой точки зрения можно выделить две типовые модели знаний.

Онтология информационной системы [1, 2] в явном виде описывает понятия и отношения предметной области (ПО) и позволяет не только повысить уровень представления и обработки данных в самой системе, но и создавать дополнительные предметно-ориентированные сервисы, в том числе и лингвистические. Модель хорошо поддерживает навигацию в соответствии с онтологией и «структурный» поиск, т. е. поиск в терминах онтологии. Однако для создания полноценных сервисов, ориентированных на автоматическую обработку текста, этих знаний недостаточно.

Наиболее распространенной моделью второго типа, поддерживающей работу с текстом, является тезаурус [3]. Тезаурус — это знания о языке в проекции на конкретную сферу деятельности, которые включают чисто лингвистическую информацию о «взаимоотношениях» лексики в данной области и, при необходимости, соотнесение с универсальной моделью языка. Такая модель хорошо поддерживает анализ и семантическую индексацию текстов, естественно-языковой поиск релевантных документов, создание аннотации и т. п. К недостаткам можно отнести то, что результаты запроса представлены текстами, а не конкретной информацией, как правило, присутствует много дублей и нерелевантных текстов. В таких системах возникает потребность в структуризации информации в соответствии с интересами пользователя.

В работе А.С. Нариньяни [4] предложена концепция ТЕОН, которая описывает системы, использующие при своем создании и функционировании органично взаимодействующие онтологию и тезаурус предметной области. Сложность организации такого взаимодействия связана с тем, что

не всегда в языке есть однозначное соответствие между понятием и термином, часто понятия выражаются более сложным образом: словосочетаниями или фразами, части которых могут быть разнесены по тексту. Это связано с проблемой лексической многозначности слова [5] и требует для уточнения исследования контекста. Наличие определенных правил формирования текста не позволяет в полной мере отражать принципы преобразования текста с помощью тезауруса, требуются дополнительные лингвистические модели и ресурсы.

В работе будет рассмотрена расширенная модель представления знаний — *лингвистическая онтология*, ориентированная на разработку естественно-языковых сервисов, необходимых для создания и наполнения базы ИС, т. е. в первую очередь на автоматический анализ текстов в ограниченной предметной области. Данная модель достаточно полно отражает процесс связывания тезауруса и онтологии, при этом выделяются максимально независимые компоненты модели, что с технологической точки зрения дает все преимущества модульного подхода создания программных систем: независимая и распределенная реализация, повторное использование, прозрачность, легкость расширения и т. п.

**1. Жизненный цикл базы знаний
информационной системы**

Разработка интеллектуальной ИС начинается с проектирования и наполнения ее базы знаний, которая в дальнейшем должна обновляться и пополняться, знания не должны терять свою актуальность. Таким образом, создание и развитие ИС с точки зрения ее знаний, информационного наполнения и ресурсов — это обязательно итеративный процесс. Мы будем рассматривать жизненный цикл онтологии ИС в контексте использования средств анализа текста для ее развития (рис. 1). Отметим, что задачи, также как и типы документов, обрабатываемые лингвистическими сервисами на разных этапах развития ИС, различны.



Рис. 1. Общая схема разработки базы знаний информационной системы

На начальном этапе разработки системы онтология играет важную роль при анализе требований и концептуальном моделировании. На данном этапе осуществляется проектирование базы знаний системы – формируется онтология верхнего уровня (метаонтология), фиксируются основные термины и понятия предметной области.

1. Онтологический анализ ПО начинается с создания словаря терминов, который используется при обсуждении и исследовании характеристик объектов и процессов, составляющих рассматриваемую ПО, также выделяются основные логические взаимосвязи между понятиями, которые соответствуют введенным терминам. Создание словаря можно автоматизировать, используя методы автоматического обучения на основе корпуса текстов предметной тематики. Задача извлечения предметной терминологии включает поиск как однословных, так и многословных терминов [6–8].
2. Задача автоматизации процесса извлечения экспертных знаний о ПО и ее подязыке эффективно решается методами корпусной лингвистики, то есть путем создания и исследования специализированного корпуса текстов, представляющего собой достаточный объем снабженных экспертной интерпретацией лингвистических данных, который может выступать в роли обучающего корпуса. В состав корпуса текстов отбираются фрагменты из справочной и учебной литературы, научные статьи и рефераты, посвященные определенной тематике. Процессу семантической разметки специализированного корпуса текстов предшествует достаточно длительный предварительный этап совместной работы экспертов, лингвистов и разработчиков системы, в рамках которого происходит обмен компетенциями, выработка и согласование признаков и принципов разметки. Наличие семантически размеченного корпуса позволяет автоматизировать создание других лингвистических ресурсов, в первую очередь семантических словарей.
3. На следующем этапе рассматривается задача автоматического добавления справочной информа-

ции в базу знаний системы. Знания, хранимые в справочниках, должны иметь естественную проекцию в онтологии ПО. Справочные ресурсы – это, как правило, хорошо структурированные тексты, поэтому использование формальной жанровой модели таких ресурсов может значительно упростить процесс анализа текста и его настройку.

4. Накопление информации в ИС может осуществляться путем семантического индексирования потока документов. Онтология ПО определяет формат данных, хранимых в ИС, и, следовательно, определяет, какую именно информацию необходимо извлекать из текста документа, а какую можно проигнорировать. Семантический индекс документа представляется в виде сети объектов, являющихся экземплярами понятий и отношений онтологии. Данная семантическая сеть добавляется в базу данных ИС и преобразуется в знания, которыми в дальнейшем может оперировать система.
5. Задача поддержки актуальности индекса документов в таком представлении перестает зависеть от устаревания и изменения терминологии (от этого, конечно, зависит корректная обработка новых документов). Но задача устаревания информации в ИС остается актуальной и тесно связана с разрешением противоречий, возникающих при поступлении (в результате автоматического анализа новых документов) и идентификации новых данных, не согласующихся с информацией, уже присутствующей в системе.

Внесение изменений в онтологию предметной области (а также в онтологию верхнего уровня или метаонтологию) возможно либо при изменении требований к системе со стороны пользователя, либо при накоплении достаточного количества фактов, сигнализирующих о наличии неполноты в системе описания онтологии. Данные факты могут извлекаться из текста по специальным правилам с обязательным требованием высокой точности.

Для доступа к информации ИС разрабатываются пользовательские сервисы, такие как информационный поиск фактов или документов, содержащих определенные факты, представление кратких

рефератов просматриваемых документов, структурирование информации, полученной по поисковому запросу пользователя (рубрикация, кластеризация), и т. п. Сам запрос пользователь может оформлять либо на естественном языке в виде вопроса (вопросно-ответное взаимодействие), либо по ключевой фразе, либо заполняя определенную форму (формируя тем самым структурированный в соответствии с ПО запрос), либо используя навигационные средства, представляемые ИС.

2. Лингвистические потребности информационных систем

Таким образом, в процессе развития информационной системы возникают различные лингвистические задачи, решение которых требует разработки различных лингвистических ресурсов, а также средств их создания и применения (рис. 2).

В соответствии с потребностями ИС выделены следующие типы лингвистических ресурсов:

- 1) *Корпус текстов* — подборка текстов определенного жанра, тематика которых соответствует заданной ПО. Корпус может содержать лингвистическую разметку, представляющую собой информацию, полученную автоматически при анализе текстов либо приписанную экспертом вручную. Основное назначение корпуса — автоматизация создания других лингвистических ресурсов.
- 2) Универсальные и *предметные словари*, содержащие перечень минимальных единиц языка, терминов и устойчивых терминологических словосочетаний, используемых при описании значимой для ИС информации, а также жанровую лексику, описываемую лексическими шаблонами, для извлечения нестандартно представленной в тексте лексики. В рамках словарей определяется набор универсальных и/или специ-

фичных для заданного подязыка лингвистических знаний: морфологические классы, правила формирования многословных терминов и т. п.

- 3) Набор описаний жанровых структур текста в совокупности с логическим представлением текста образуют *модели документов*, соотнесенных с тем или иным типом текстовых ресурсов, хранящихся в ИС.
- 4) *Семантический словарь*, формирующий семантические признаки и отношения на лексиконе, включает целевые тезаурусы (например, справочно-информационный тезаурус, тезаурусы для анализа текста, для поддержки информационного поиска, для перевода и т. п.), а также словарь моделей управления, который ограничивает синтаксическую сочетаемость и проверяет согласованность грамматических и семантических признаков терминов (вершин синтаксических групп) в соответствии с правилами согласования и управления [9]. Эти знания могут быть заданы с разной степенью подробности в зависимости от требований и возможностей разработчиков ИС.
- 5) Знания о согласовании имеющихся лингвистических знаний с предметными знаниями, заданными онтологией ИС. С этой целью термины группируются в семантические группы, которые, в свою очередь, также согласуются с элементами онтологии либо непосредственно, либо в соответствии с определенной схемой (*схемой факта*) [10].
- 6) Результаты семантического индексирования всех текстов помещаются в *хранилище документов* и образуют единую информационную сеть объектов [11]. Хранилище должно поддерживать возможность поиска и идентификации объектов, а также придерживаться определенной стратегии для разрешения противоречий.

Задачи	Ресурсы	Средства
Извлечение терминологии	<i>Корпуса текстов</i> <i>Предметные словари</i>	<i>Средства разметки текстов</i> <i>Средства автоматического построения предметных словарей</i>
Обработка структурированных текстов	<i>Лексические шаблоны</i> <i>Тезаурусы</i> <i>Модели документов</i>	<i>Рабочие места лингвиста для формирования ресурсов</i>
Извлечение информации (фактов)	<i>Семантико-синтаксические модели</i> <i>Схемы фактов</i>	<i>Средства описания моделей и схем фактов</i> <i>Средства автоматического извлечения фактов</i>
Поддержка актуальности информации	<i>Семантически-индексированные хранилища документов</i>	<i>Средства идентификации фактов (данных)</i> <i>Стратегии разрешения противоречий на данных</i>

Рис. 2. Лингвистические потребности информационных систем

3. Лингвистическая модель знаний

Лингвистическая модель знаний должна включать всю совокупность лингвистических и экспертных знаний, необходимых для анализа текста на естественном языке. Особенностью предложенной модели является то, что в нее помимо онтологии и тезауруса включаются дополнительные компоненты, предназначенные для распознавания контекста терминов в тексте и последующего связывания найденных элементов с понятиями и отношениями предметной области.

Рассматривались следующие типы контекста.

- Устойчивое словосочетание (словокомплекс), характеризующееся наличием определенной синтаксической связи между контактно расположенными словами и высокой частотностью в анализируемом подязыке.
- Фраза — к данному типу относятся неустойчивые словосочетания, для которых не задан строгий порядок слов, а также фразы, которые формируются нерегулярным образом: могут быть разрывными, используют различные элементы языковой отсылки.
- Множество связанных фраз — данный контекст характеризуется определенным линейным порядком связанных частей, использованием различных видов языковой редукции (сочинение и другие виды эллипсиса, анафора) и позволяет извлекать из текста информационные объекты со сложной структурой и связи.
- Текст — важным свойством данного контекста является жанр текста. Жанр играет важную роль в определении формальной структуры создаваемого автором текста (наличие в тексте таких разделов, как заголовок, резюме, основной текст).
- Информационное наполнение ИС, в том числе знания, извлеченные из ранее обработанных документов.

Формально лингвистическая модель знаний или лингвистическая онтология, для которой задана онтология предметной области O и ее информационное наполнение I_O , определяется пятеркой вида $\langle V, W, T, F, D \rangle$, где V — словарь, включающий минимальные единицы текста — лексемы и лексические конструкции (сокращения, числа, численно-буквенные обозначения и т. п.), W — словарь устойчивых словосочетаний (словокомплексов) и наименований, T — семантический словарь (тезаурус), который устанавливает тезаурусные отношения между элементами словарей V и W (синонимия, родо-видовые связи, ассоциации и т. п.), F — множество упорядоченных наборов схем фактов, D — множество моделей документов, для каждой из которых может быть определен собственный набор схем фактов.

Технология, поддерживающая создание ИС, требует разработки целого комплекса лингвистических инструментов, направленных не только на непосредственную обработку текста, но и на разработку различных предметно-ориентированных лингвистических ресурсов.

4. Лингвистические процессоры

С технологической точки зрения можно выделить два типа лингвистических сервисов: системные сервисы, используемые для автоматического наполнения и изменения содержания контента системы, и пользовательские сервисы, предоставляющие пользователям разнообразный доступ к информации.

Полная цепочка процессов, обеспечивающая системные сервисы, включает графематический, лексический и морфологический анализ, жанровую сегментацию, поиск и извлечение фактов, идентификацию информационных объектов и формирование контента документа.

Процесс обработки текста начинается с предварительного словарного анализа, включающего сегментацию текста, морфологический, лексический и поверхностно-синтаксический анализ. В результате предварительного этапа формируется цепочка лексических объектов, а также сегментное покрытие текста, представленное сегментами разных типов (абзац, предложение, клауза, заголовок, скобки и т. п.), которые необходимо учитывать в условиях схем фактов.

В процессе основного анализа лексические объекты в соответствии со схемами фактов преобразуются в семантические объекты — гипотезы, которые в дальнейшем проверяются контекстом. При этом объект (как сущность предметной области, описывающая интересующую пользователя информацию) возникает на самом раннем этапе на основе семантической информации, представленной в словаре.

В общем виде задача семантического анализа текста, решаемого в рамках нашего подхода, может быть сформулирована следующим образом.

Для заданной пятерки $\langle O, F, T, \Omega, S \rangle$, где O — онтология предметной области, F — множество схем фактов, T — текстовый фрагмент, Ω — терминологическое покрытие T , S — сегментное покрытие T , найти все семантические структуры, соответствующие онтологии O , покрывающие область T , которые можно получить в процессе применения правил из F к Ω с учетом S .

Ключевым понятием разрабатываемого подхода является схема фактов, которая фиксирует структуру языкового высказывания о факте действительности и явным образом связывает его с элементом онтологии ПО. Множество схем фактов описываются экспертами в терминах семантических категорий ПО и автоматически реализуются лингвистическим процессором.

Формально схема фактов — это тройка вида $\langle Arg, C_F, Res \rangle$, где Arg — множество аргументов факта (семантические признаки словаря, понятия и отношения онтологии ПО, вспомогательные классы фактов), C_F — множество грамматических, синтаксических, семантических и структурных ограничений на сочетаемость аргументов, $Res = \langle s_i, P \rangle$ — результат применения схемы факта, где s_i задает класс создаваемого или редактируемого объекта, а P — множество правил для формирования или уточнения значений атрибутов объекта.

Процесс поиска фактов можно рассматривать как с точки зрения продукционного подхода, где каждой схеме сопоставляется продукционное правило, так и с точки зрения мультиагентного подхода [11, 12], где каждому лексическому объекту, порожаемому на основе словарной информации, можно сопоставить *агента*, цель которого — исследовать контекст, представленный терминами и другими агентами, заполнить свои атрибуты и выявить возможные связи с другими агентами. В первом случае анализ текста заключается в последовательном применении правил к цепочке лексических объектов, во втором случае агенты сами ищут подходящие схемы фактов и осуществляют их проверку в параллельном режиме. И тот и другой подход имеет свои достоинства и недостатки, обсуждение которых выходит за рамки данной статьи.

Результатом работы лингвистического процессора является множество информационных объектов. В дальнейшем модуль формирования контента документов идентифицирует и уточняет параметры полученных объектов, сравнивая их с информационными объектами, хранящимися в ИС, формирует множество информационных объектов, представляющее проанализированный документ и его контент, а также устанавливает между ними необходимые связи.

СПИСОК ЛИТЕРАТУРЫ

1. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. — СПб.: Питер, 2001. — 384 с.
2. Загорюлько Ю.А., Боровикова О.И. Подход к построению порталов научных знаний // Автометрия. — 2008. — Т. 44. — № 1. — С. 100–110.
3. Добров Б.В., Лукашевич Н.В. Онтологии для автоматической обработки текстов: описание понятий и лексических значений // Компьютерная лингвистика и интеллектуальные технологии: Труды Междунар. конф. «Диалог». — Бекасово, 2006. — М.: РГГУ, 2006. — С. 138–142.
4. Нариньяни А.С. ТЕОН-2: от Тезауруса к Онтологии и обратно // Компьютерная лингвистика и интеллектуальные технологии: Труды Междунар. семинара Диалог'2002. — Протвино, 2002. — М.: Наука, 2002. — Т. 1. — С. 154–199.
5. Поляков В.Н. Использование технологий, ориентированных на лексическое значение, в задачах поиска и классификации // Проблемы прикладной лингвистики. Вып. 2 / отв. ред. Н.В. Васильева. — М.: Азбуковник, 2004. — С. 101–117.
6. Большаков И.А. Какие словосочетания следует хранить в словарях? // Труды Междунар. семинара Диалог'2002 по компьютерной лингвистике и ее приложениям. — Протвино, 2002. — Т. 2. — С. 61–69.
7. Большакова Е.И., Носков А.А. Система для поиска и выделения конструкций в тексте на естественном языке // КИИ-2010: Труды XII национальной конф. по искусственному интеллекту с международным участием. — Тверь, 2010. — М.: Физматлит, 2010. — Т. 3. — С. 137–145.
8. Сидорова Е.А. Многоцелевая словарная подсистема извлечения предметной лексики // Компьютерная лингвистика и интеллектуальные технологии: Труды Междунар. конф. «Диалог». — Бекасово, 2008. — М.: РГГУ, 2008. — Вып. 7 (14). — С. 475–481.
9. Волкова И.А., Головин И.Г., Кривнова О.Ф. Компьютерный словарь моделей управления русских глаголов (экспериментальный вариант) // Диалог'98: Труды Междунар. семинара по компьютерной лингвистике и ее приложениям. — М., 1998. — С. 448–452.
10. Васильев И.А., Тузовский А.Ф. Структура системы управления знаниями // Информационные и системные технологии в индустрии, образовании и науке: Труды Междунар. симпозиума. — Караганда: КарГТУ, 2003. — С. 286–288.
11. Нариньяни А.С. ИИ и Мультиагентные технологии // Проблемы управления и моделирования в сложных системах: Труды VIII Междунар. конф. — Самара: Самарский научный центр РАН, 2006. — С. 491–497.
12. Вольман С.И., Минаков И.А., Томин М.С. Мультиагентная система интеллектуального анализа содержимого Интернет-страниц // Проблемы управления и моделирования в сложных системах: Труды VII Междунар. конф. — Самара: СНЦ РАН, 2005. — С. 403–408.
13. Загорюлько Ю.А., Кононенко И.С., Костов Ю.В., Сидорова Е.А. Подход к интеллектуализации документооборота // Информационные технологии. — 2004. — № 11. — С. 2–11.
14. Сидорова Е.А., Загорюлько Ю.А., Боровикова О.И. Подход к автоматизации извлечения информации из текстов по археологии // Информационные технологии в гуманитарных исследованиях. Вып. 11. — Новосибирск: Изд-во НГУ, 2006. — С. 1–8.
15. Кононенко И.С., Сидорова Е.А. Подход к извлечению фактов из текста на основе онтологии // Компьютерная лингвистика и интеллектуальные технологии: Матер. ежегодной Междунар. конф. «Диалог». — Бекасово, 2009. — М.: РГГУ, 2009. — Вып. 8 (15). — С. 451–457.

Заключение

Рассмотренная в статье технология разработки лингвистического обеспечения для информационных систем поддерживает всю цепочку создания и использования лингвистических ресурсов для автоматического анализа текста и формирования контента документа. Система включает средства создания лингвистической онтологии, с помощью которых настройка процесса содержательной обработки документов может выполняться непосредственными носителями знаний — экспертами и лингвистами, не имеющими специальных навыков программирования.

Основные компоненты технологии были успешно апробированы в ряде практических приложений, служащих для поддержки научной и производственной деятельности, в частности при разработке интеллектуальной системы документооборота инвестиционной компании [13], портала знаний по археологии [14], архива «Хроники СО РАН» [15].

Работа выполнена при финансовой поддержке РФФИ (проект № 12–07–31216) и Президиума РАН (интеграционный проект СО РАН № 15/10 «Математические и методологические аспекты интеллектуальных информационных систем»).

Поступила 30.10.2012 г.

УДК 004.415

МЕТОДЫ ОПРЕДЕЛЕНИЯ УРОВНЕЙ БЕЗОПАСНОСТИ ЭЛЕМЕНТОВ ОНТОЛОГИИ

Хоанг Ван Куэт, А.Ф. Тузовский

Томский политехнический университет
E-mail: student8050@sibmail.com

Рассматриваются основные проблемы многоуровневой безопасности в семантических данных. Описаны принципы задания уровней безопасности понятий, свойств и индивидуумов онтологии. Предложены алгоритмы, позволяющие определить уровни безопасности основных элементов онтологии и логических выводов, полученных в логических правилах.

Ключевые слова:

Онтология, уровень безопасности, алгоритм, логические выводы.

Key words:

Ontology, level security, algorithm, inference.

Проблема обеспечения безопасности семантических данных является достаточно актуальной в связи с активным ростом использования семантических технологий для хранения и обработки информации в информационных системах различных организаций [1]. Для решения данной проблемы предлагается многоуровневая система обеспечения безопасности семантических данных, заключающаяся в том, что каждому элементу семантической базы данных задаётся собственный уровень безопасности. В семантических данных каждый элемент одновременно может являться субъектом или объектом, а также может содержать или принадлежать другим элементам, имеющим разные уровни безопасности, следовательно, его уровни безопасности не являются постоянными [2].

Кроме этого, следует также учитывать, что при работе с семантическими данными на основе фактов, извлечённых из базы данных, пользователи могут с помощью логических правил получить новую, не разрешённую им информацию (новое утверждение), имеющую неизвестный уровень безопасности [3].

С целью обеспечения работы с семантическими данными необходимо гарантировать, что уровень каждого их элемента является определённым и единственным. Для выполнения этого требования необходимо создать метод, позволяющий определить уровни безопасности элементов данных.

Целью данной статьи является пояснение алгоритмов определения уровней безопасности каждого элемента онтологий и логических выводов, полученных при выполнении логических правил в семантических данных.

Постановка задач и основные понятия

Онтология — это формальная модель некоторой области знаний с помощью концептуальной схемы. Обычно такая схема состоит из структуры данных, содержащей все релевантные классы объектов (понятия), их связи и правила (теоремы, ограничения), принятые в этой предметной области. Онтологии используются в информационных системах как форма представления знаний о реальном мире

или его части. Современные онтологии в основном строятся одинаково, независимо от языка написания, и обычно они состоят из индивидуумов, понятий, атрибутов и отношений.

В базе данных хранятся онтологии и индивидуумы. Под онтологической моделью (онтологией) O понимается знаковая система $\langle C, T, P \rangle$, где $C = \{C_1, \dots, C_n\}$ — множество элементов, которые называются понятиями (классами), где n — количество понятий в онтологии; T — частичный порядок на множестве C , задающий отношения «подкласс» и «суперкласс»; $P = \{P_1, \dots, P_m\}$ — множество элементов, называемых свойствами (атрибутами, отношениями), где m — количество свойств в онтологии (двуместными предикатами) [4]. Множество индивидуумов обозначается как $E = \{E_1, \dots, E_k\}$, где k — количество индивидуумов.

С целью безопасности доступа к семантическим данным элементу в онтологии и утверждению в базе данных задаются уровни безопасности s , значения которых выбираются из множества меток, например таких, как {неклассифицированный ($s=1$), конфиденциальный ($s=2$), секретный ($s=3$), сверхсекретный ($s=4$)}. Таким образом, множеству элементов онтологии O соответствует множество уровней безопасности, которое обозначается, как $\langle S_c, S_p, S_e \rangle$, где S_c — множество уровней безопасности понятий, S_p — множество уровней безопасности свойств и S_e — множество уровней безопасности индивидуумов [5]. На рис. 1 показана онтология, у которой элементы имеют 4 уровня безопасности.

В онтологии класс C_a может являться подклассом класса C_b или суперклассом класса C_c . В процессе создания онтологий для классов C_a , C_b и C_c разработчики могут задать уровни безопасности s_a , s_b и s_c . Таким образом, для класса онтологии может быть задано несколько или ни одного значения уровней безопасности. В этом случае актуальной является задача определения уровня безопасности для каждого элемента онтологии.

Для работы с семантикой информации может быть использован язык Semantic Web Rule Language (SWRL), основанный на объединении языков

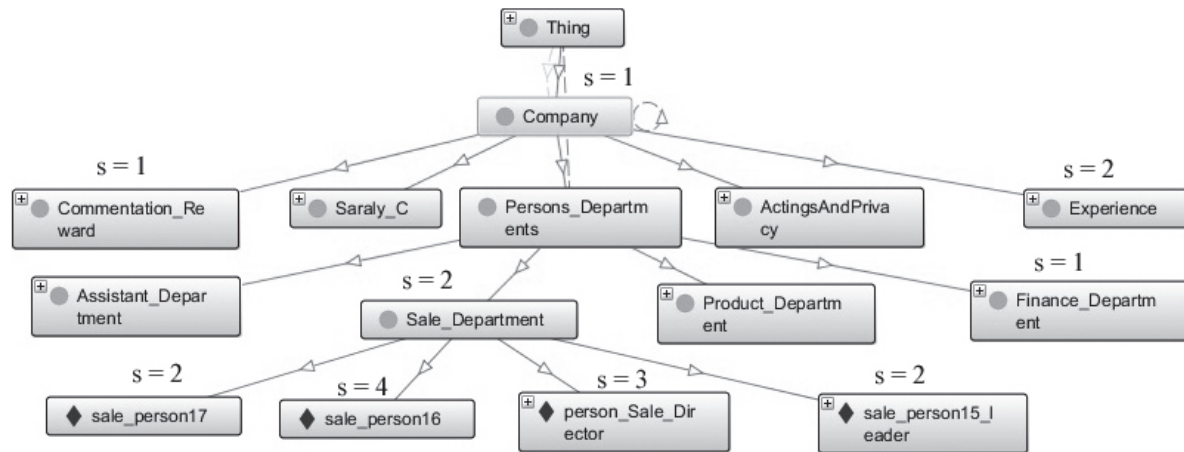


Рис. 1. Пример онтологии с уровнями безопасности

OWL и RuleML, т. е. объединены онтологии (OWL-DL) и правила [6]. Он позволяет формально описать способ работы с объектами предметной области, а так же закономерности предметной области. С помощью данного языка на основе уже известных семантических данных могут быть получены новые утверждения (т. е. сделаны логические выводы). На рис. 2 показан пример логического правила, написанного на языке SWRL.

```
(?a rdf: type owl: NamedIndividual) ∧
(?b rdf: type owl: NamedIndividual) ∧
(?a rdf: type ontology: Finance_Department) ∧
(?b rdf: type ontology: Finance_Department) ∧
(?a ontology: isLeaderDepartmentOf ?b)
→ (?a rdf: type ontology: Leader_Finance)
```

Рис. 2. Пример описания логических выводов на языке SWRL

В данном примере описывается следующее правило: если два человека a и b работают в финансовом отделе, и a руководит b , то следовательно a является начальником данного отдела.

Теперь предположим, что a имеет уровень безопасности $s_i=1$, а классу *Leader-Finance* задан уровень безопасности $s_j=2$, тогда в соответствии с данным логическим правилом a является элементом данного класса и ему должен быть задан уровень безопасности $s_i=s_j=2$. Таким образом, необходимо создать метод для определения уровней безопасности элементов, полученных в результате применения логических правил.

Алгоритмы определения уровней безопасности элементов в онтологии

Для построения алгоритмов, позволяющих определить уровни безопасности элементов онтологии, предположим следующие принципы:

- В онтологиях нет элементов, не имеющих уровней безопасности.
- Если элементу не дан начальный уровень безопасности, то его уровень безопасности равен нулю.

- Уровень безопасности подкласса должен быть больше или равен уровню безопасности суперкласса.
- Уровень безопасности объекта должен быть больше или равен уровням безопасности классов, которым он принадлежит.
- Уровень безопасности свойства должен доминировать над уровнем безопасности других свойств, которым оно принадлежит.
- В зависимости от логических операций, каждый индивидум может принадлежать нескольким классам, следовательно, логические выводы могут обладать некоторыми значениями уровней безопасности.

Алгоритм определения уровней безопасности классов

Понятия или классы – это абстрактные группы, коллекции или наборы объектов. Они могут включать в себя индивидумы, другие классы, либо же сочетания и того, и другого. Классы онтологии составляют таксономию (иерархию понятий) на основе отношения включения (*SubclassOf*). На основе вышеуказанных принципов, алгоритм (алгоритм 1) определения уровней безопасности понятий в онтологии может быть описан следующим образом:

1. Если начальный уровень безопасности s_x понятия C_x не создан, то ему присвоено значение равное нулю ($s_x=0$).
2. Если понятие C_x является подклассом другого понятия C_y , имеющего уровень безопасности s_y , то необходимо сравнить уровень безопасности s_x с уровнем безопасности s_y .
3. Если $s_y > s_x$, то s_x будет присвоено значение s_y , т. е. $s_x=s_y$, в противном случае $s_x=s_x$ и алгоритм заканчивается.
4. Если понятие C_x не является подклассом никакого другого понятия C_y , то алгоритм заканчивается.

Схема алгоритма, позволяющего определять уровень безопасности понятия, показана на рис. 3. В результате выполнения алгоритма уровень безо-

пасности каждого понятия будет определён и будет единственным.

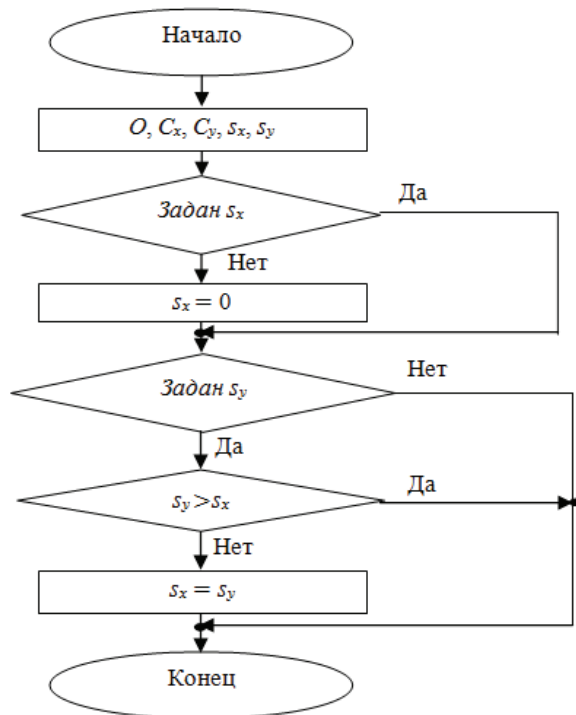


Рис. 3. Алгоритм определения уровня безопасности понятия

В онтологии класс может одновременно являться подклассом одного класса или суперклассом другого, тогда для определения уровней безопасности всех классов необходимо выполнять рекурсивные операции для разработанного алгоритма. На рис. 4 показано описание метода определения уровней безопасности для всех классов в онтологии с помощью языка SWRL.

```

(?C_x rdf: type owl: Class) ∧
noValue (?C_x ontology: Level ?s_x) →
(?C_x ontology: Level 0).
(?C_x ontology: Level ?s_x) ∧
(?C_y ontology: Level ?s_y) ∧
(?C_x rdfs: subClassOf ?C_y) ∧
greaterThan (?s_y, ?s_x) → drop (1) ∧
(?C_x ontology: Level ?s_y).
  
```

Рис. 4. Определение уровня безопасности классов онтологии с помощью языка SWRL

Определение уровней безопасности свойств

Объекты в онтологии могут иметь атрибуты. Каждый атрибут имеет, по крайней мере, имя и значение и используется для хранения информации, которая специфична для объекта и связана с ним. Важная роль атрибутов заключается в том, чтобы определять отношения между элементами онтологии. Обычно под отношением понимается атрибут, значением которого является какой-то объект.

В онтологии между свойствами также могут даваться отношения, например, одно свойство мо-

жет включать в себя другие свойства или принадлежать им. Уровень безопасности свойства может быть описан в процессе создания онтологии или определён с помощью уровней безопасности других элементов. Способ, позволяющий определить уровни безопасности свойств онтологии, аналогичен подходу, описанному в алгоритме 1. Для этого также необходимо рекурсивно выполнять алгоритм, используемый для определения уровня безопасности одного свойства. На рис. 5 показан способ определения уровней безопасности s_x всех свойств P_x в онтологии, описанный с помощью языка SWRL.

```

(?P_x rdf: type owl: ObjectProperty) ∧
noValue (?P_x ontology: Level ?s_x) →
(?P_x ontology: Level 0).
(?P_x ontology: Level ?s_x) ∧
(?P_x rdfs: subPropertyOf ?P_y) ∧
(?P_y ontology: Level ?s_y) ∧
greaterThan (?s_y, ?s_x) → drop (1) ∧
(?P_x ontology: Level ?s_y).
  
```

Рис. 5. Определение уровней безопасности свойств онтологии с помощью языка SWRL

Алгоритм определения уровней безопасности индивидуумов в онтологии

Индивидуумы (экземпляры) — это основные элементы нижнего уровня онтологии. Индивидуумы могут соответствовать как физическим объектам (люди, дома, планеты), так и абстрактным сущностям (числа, слова).

Аналогично алгоритму 1 алгоритм определения уровня безопасности индивидуума создан таким образом:

- если начальный уровень безопасности (s_x) индивидуума (E_x) не задан, то его значение равно нулю ($s_x=0$);
- если индивидуум E_x является элементом класса C_y , имеющего уровень безопасности s_y , то необходимо сравнить s_x с уровнем безопасности s_y ;
- если $s_x < s_y$, то s_x будет присвоено значение s_y , т. е. $s_x=s_y$, в противном случае $s_x=s_x$ и алгоритм заканчивается.

Схема алгоритма (алгоритм 2) для определения уровня безопасности индивидуума показана на рис. 6.

В результате выполнения алгоритма 2 уровень безопасности индивидуума в базе данных будет единственным и будет доминировать над уровнем безопасности класса, которому данный индивидуум принадлежит.

В базе данных хранится много индивидуумов, имеющих свой уровень безопасности. С помощью алгоритма 1 могут быть определены уровни безопасности s_y всех классов онтологии C_y , после этого могут быть определены и уровни безопасности s_x всех индивидуумов E_x онтологии с помощью способа описанного языка SWRL, как показано на рис. 7.

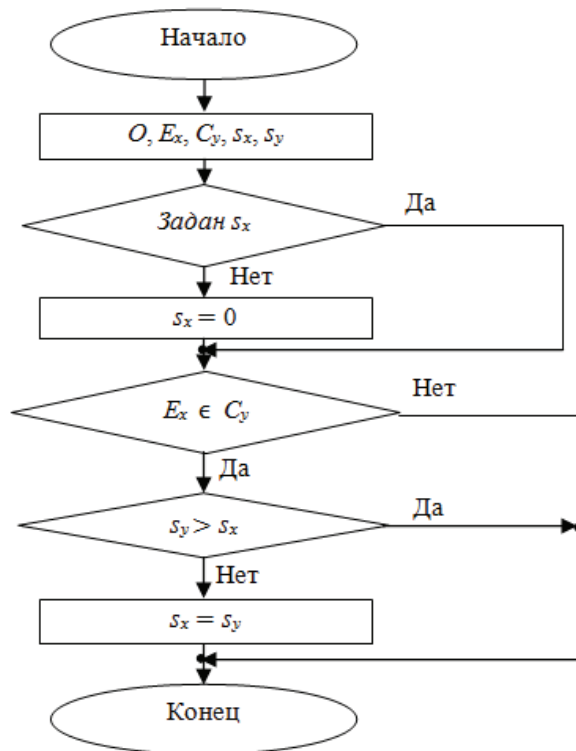


Рис. 6. Алгоритм для определения уровня безопасности индивидуумов онтологии

```
(?Ex rdf: type owl: NamedIndividual) ∧
noValue (?Ex ontology: Level ?sx) →
(?Ex ontology: Level 0).
(?Ex ontology: Level ?sx) ∧
(?Ex rdf: type ?Cy) ∧
(?Cy ontology: Level ?sy) ∧
greaterThan (?sy, ?sx) → drop (1) ∧
(?Ex ontology: Level ?sy).
```

Рис. 7. Определение уровней безопасности индивидуумов с помощью языка SWRL

Уровень безопасности утверждений

Семантические метаданные $M=\{\mu_1, \dots, \mu_m\}$ – это наборы семантических утверждений (триплетов) μ_i , которые имеют вид $\mu_i=(\alpha, \beta, \gamma)$, где α – это субъект утверждения (понятие, или индивидуум – контекстные метаданные некоторого понятия), β – объект (экземпляр – индивидуум, контекстные метаданные некоторого понятия), γ – отношение между субъектом и объектом. При этом понятия и отношения должны быть определены в онтологии O , а индивидуумы описываются контекстными метаданными онтологической базы знаний. Примерами утверждений являются следующие триады $\langle C, P, C \rangle$, $\langle E, P, E \rangle$, $\langle C, P, E \rangle$, $\langle E, P, V \rangle$, $\langle C, P, NULL \rangle$, $\langle C, NULL, NULL \rangle$, $\langle E, P, NULL \rangle$, $\langle E, NULL, NULL \rangle$, где C – понятие; E – индивидуум понятия; P – связь; A – атрибут; V – значение атрибута (текстовое или числовое) [7]. Для контроля доступа пользователей к утверждению необходимо сравнить их уровни доступа с уровнем безопасности утверждения.

С помощью описанных выше алгоритмов могут быть определены уровни безопасности всех элементов (понятий, связей, атрибутов, индивидуумов) семантической базы данных. В этом случае уровень безопасности s_μ утверждения μ_i определяется как максимальное значение из уровней $\{s_\alpha, s_\beta, s_\gamma\}$, где s_α – это уровень безопасности субъекта, s_β – это уровень безопасности отношения, s_γ – это уровень безопасности объекта.

Алгоритм определения уровней безопасности логических выводов

В онтологии логическим правилом R является выражение, обозначающееся как $\forall x_1, \dots, x_m (b_1 \wedge \dots \wedge b_k) \rightarrow q$, где $k \geq 1$ и $x_1, \dots, x_m$ – это свободные переменные в $b_1 \wedge \dots \wedge b_k$. Каждый b_i представляет собой утверждение, имеющее вид $[\alpha, \beta, \gamma]$, где α, β, γ – переменные, константы или OWL аксиомы. Левая часть правила $(b_1 \wedge \dots \wedge b_k)$ называется телом, а правая часть правила (q) – головой правила. При составлении таких правил необходимо, чтобы все переменные, включаемые в голову, содержались в составе тела правила.

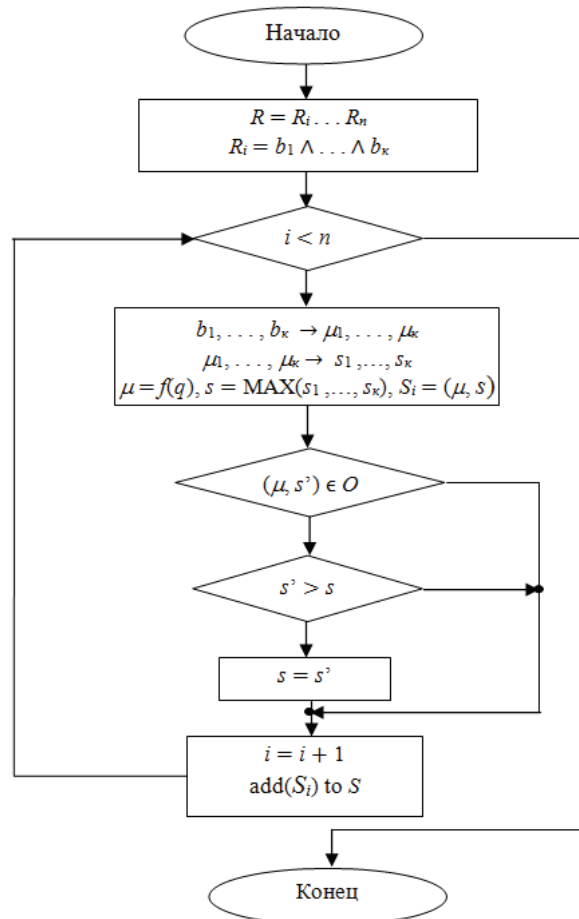


Рис. 8. Алгоритм определения уровней безопасности логических выводов

В семантических технологиях f называется функцией отображения только в случае, когда она удовлетворяется следующими условиями:

- f сохраняет все константы ($f(constant)=constant$);
- Если $f(x_1, y_1, z_1)=[\alpha_1, \beta_1, \gamma_1]$ и $f(x_1, y_2, z_2)=[\alpha_2, \beta_2, \gamma_2]$, то $\alpha_1=\alpha_2$;
- $\forall f(b_i) \in M, (i=1 \dots k)$;

Тогда с помощью функции отображения логическое правило R выводит утверждение $\mu_i=[\alpha, \beta, \gamma](f(q)=[\alpha, \beta, \gamma])$.

Алгоритм определения уровней безопасности логических выводов с использованием правил может быть описан следующим образом:

1. Определяется уровень безопасности каждого логического вывода.

Если существует функция отражения f : $b_1, \dots, b_k \rightarrow \mu_1, \dots, \mu_k$, где $f(b_1)=\mu_1, \dots, f(b_k)=\mu_k$, то:

- генерируется отражение $\mu=f(q)$ (по определению – существуют логические выводы). После отражения получим утверждение μ , которое должно иметь уровень безопасности s ;
- генерируется значение для уровня безопасности логического вывода с использованием выражения $s=\text{MAX}(s_\alpha, s_\beta, s_\gamma)$;
- генерируется пара безопасности $S_i=(\mu, s)$ (утверждение, уровень безопасности).

2. Выполняется контроль значений уровней безопасности.

Если в базе данных уже существует утверждение μ , имеющее уровень безопасности s' , который больше уровня безопасности логического вывода ($s > s'$), то s присваивается значение s' . В противном случае, т. е. если ($s < s'$), то значение

уровня безопасности логического вывода будет равно s .

3. Безопасность (μ, s) добавляется к множеству пар (утверждение, уровень безопасности) безопасности S семантических метаданных M .

На рис. 8 показан алгоритм для определения уровней безопасности логических выводов.

С помощью данного алгоритма, после выполнения логических правил, все уровни безопасности элементов логических выводов определены. В зависимости от своего уровня доступа пользователь может видеть адекватные полученные логические выводы.

Заключение

В работе рассмотрены основные задачи многоуровневой безопасности для семантических данных, которые включают в себя определение уровней безопасности каждого элемента онтологии и логических выводов, выполняемых с помощью правил. Приведены основные принципы задания уровней безопасности для каждого типа элементов онтологии, на основе которых разработаны алгоритмы, использующиеся для определения значения уровней безопасности всех элементов онтологии, а также для каждого утверждения семантических метаданных. Разработан алгоритм определения уровней безопасности элементов новой информации, полученной в результате выполнения логических правил.

СПИСОК ЛИТЕРАТУРЫ

1. Berners T. The Semantic Web // Scientific American. – 2001. – V. 120. – № 3. – P. 220–225.
2. Хоанг Ван Куэт. Многоуровневая безопасность для семантических данных // Вестник науки Сибири. – 2012. – № 5 (6). – С. 93–100.
3. Хоанг Ван Куэт, Тузовский А.Ф. Контроль логических выводов в семантических данных // Известия Томского политехнического университета. – 2012. – Т. 320. – № 5. – С. 148–151.
4. Тузовский А.Ф., Ямпольский С.В. Системы управления знаниями (методы и технологии) / под общ. ред. В.З. Ямпольского. – Томск: Изд-во НТЛ, 2005. – 260 с.
5. Хоанг Ван Куэт, Тузовский А.Ф. Алгоритмы для контроля доступа и модификации семантических данных // Электронные средства и системы управления. – 2012. – Т. 26. – № 2. – С. 41–45.
6. Ian H. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. 2004. URL: <http://www.w3.org/Submission/SWRL/> (дата обращения: 21.05.2012).
7. Тузовский А.Ф. Формирование семантических метаданных для объектов системы управления знаниями // Известия Томского политехнического университета. – 2007. – Т. 310. – № 3. – С. 108–112.

Поступила 24.02.2013 г.

УДК 004.8; 004.4'2

МЕТОД ТРЁХСТОРОННЕГО СЛИЯНИЯ ОНТОЛОГИЙ НА ЯЗЫКЕ OWL 2

И.А. Заикин

Томский политехнический университет

E-mail: zaikin@tpu.ru

Предложен алгоритм трёхстороннего слияния онтологий при многопользовательском редактировании путём попарного сравнения онтологий, основанный на выявлении совпадающих и конфликтующих изменений. Разработана программная реализация алгоритма.

Ключевые слова:

Онтология, OWL 2, трёхстороннее слияние, версия, изменение, утверждение.

Key words:

Ontology, OWL 2, three-way merge, version, change, statement.

Онтология — это формальное описание определённой предметной области на стандартизированном языке, понятном как людям, так и вычислительным машинам. В настоящее время наиболее распространённым из таких языков является язык OWL 2 [1]. С каждым годом возрастает как количество разработанных онтологий, так и их сложность. Разработка сложных онтологий требует участия специалистов по языку онтологий, специалистов в предметной области, а также специалистов по управлению качеством. Совместная разработка сложных онтологий почти невозможна без инструментальных средств поддержки групповой работы, в частности без систем управления версиями, которые дают возможность сохранять журнал изменений, в случае необходимости отменять нежелательные изменения, сравнивать версии, объединять изменения разных членов команды и облегчают разрешение конфликтов. Одним из важнейших компонентов таких систем является программа для слияния изменений, выполненных различными пользователями. На рис. 1 показаны входные и выходные данные программы трёхстороннего слияния. Символом v_0 обозначена исходная версия онтологии, v_1 — версия, изменённая первым пользователем, v_2 — версия, изменённая вторым пользователем, v_3 — итоговая версия, содержащая изменения обоих пользователей.

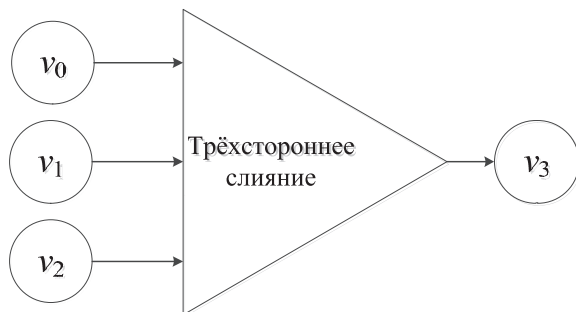


Рис. 1. Схема трёхстороннего слияния версий

Для выполнения слияния изменений в файлах традиционно используется попарное их сравнение и последующий поиск совпадающих, несовпа-

дающих и конфликтующих изменений. В данной работе этот подход применяется к онтологиям на языке OWL 2 (рис. 2). В то время как обычные программы для слияния изменений в текстовых файлах опираются на предположение, что порядок строк в тексте имеет значение, язык OWL 2 не накладывает ограничений на порядок следования аксиом. К тому же, одна и та же онтология может быть сохранена в различных форматах, что не позволяет использовать для слияния изменений в онтологиях обычные программы слияния изменений в текстах.

Язык OWL 2 имеет две семантики: 1) исторически сложившуюся, основанную на RDF, где основными понятиями являются ресурсы и отношения между ними, а сложные конструкции представлены с использованием так называемых «пустых узлов» и 2) прямую семантику, где основными понятиями являются сущность и аксиома. Сравнение RDF-графов — вычислительно сложный процесс [2], а при наличии большого количества пустых узлов невозможно однозначно идентифицировать изменённые триплеты [3]. В данной работе используется метод структурного сравнения онтологий на основе прямой семантики OWL 2. Использование прямой семантики языка OWL 2 даёт возможность не задумываться о пустых узлах, а выполнять сравнение поаксиомно, используя операции над множествами. Предложенный подход также учитывает атрибуты онтологий, не входящие в логическую составляющую, об изменении которых, тем не менее, важно знать при разработке онтологий. Благодаря этому можно, например, выполнять слияние онтологий в различных форматах представления без потери информации об изменении формата.

Постановка задачи

Под онтологией в данной работе понимается совокупность

$$\langle E, A, I, N, r_o, r_v, f \rangle,$$

где E — конечное неупорядоченное множество сущностей (классов, типов данных, индивидов и свойств); A — конечное неупорядоченное множе-

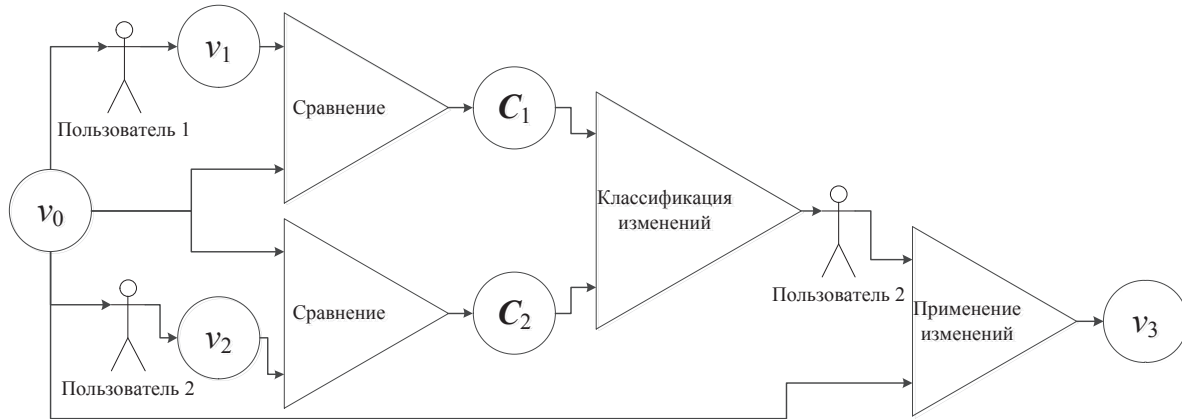


Рис. 2. Поток изменений при трёхстороннем слиянии

ство аксиом (некоторых логических утверждений о сущностях); I – конечное неупорядоченное множество ссылок на другие онтологии (импортов); N – конечное неупорядоченное множество пар вида (p, n) , где p – префикс пространства имён (короткая строка, используемая для сокращения имён сущностей), а n – интернационализированный идентификатор (IRI) пространства имён; r_o – интернационализированный идентификатор онтологии; r_v – интернационализированный идентификатор версии онтологии; f – формат хранения онтологии, может принимать одно из следующих значений: «RDF/XML», «Turtle», «OWL/XML», «OWL Functional Syntax», «OWL Manchester Syntax».

Пусть v_0 – исходная версия онтологии, v_1 – версия, изменённая первым пользователем, v_2 – версия, изменённая вторым пользователем, C_1 – конечное неупорядоченное множество изменений, внесённых в v_0 первым пользователем, C_2 – конечное неупорядоченное множество изменений, внесённых в v_0 вторым пользователем. Эти изменения можно классифицировать следующим образом:

- Совпадающие изменения $C_m = C_1 \cap C_2$;
- Конфликтующие изменения пользователя 1: C_{c1} ;
- Остальные изменения пользователя 1: $C_{o1} = (C_1 \setminus C_m) \setminus C_{c1}$;
- Конфликтующие изменения пользователя 2: C_{c2} ;
- Остальные изменения пользователя 2: $C_{o2} = (C_2 \setminus C_m) \setminus C_{c2}$.

Задача трёхстороннего слияния онтологий v_0 , v_1 и v_2 состоит в получении онтологии v_3 , основанной на v_0 и содержащей изменения C_m , C_{o1} , C_{o2} и часть изменений C_{c1} и C_{c2} , не приводящих к конфликту.

Метод трёхстороннего слияния онтологий

Для выполнения трёхстороннего слияния онтологий с учётом атрибутов, не входящих в логическую составляющую, целесообразно представить онтологию как конечное неупорядоченное множество утверждений S следующих типов:

- **OntologyFormatStatement** – утверждение, определяющее формат онтологии («RDF/XML», «Turtle», «OWL/XML», «OWL Functional Syntax», «OWL Manchester Syntax»);
- **NamespacePrefixStatement** – утверждение о префиксе пространства имён, содержащее строку префикса и строку пространства имён;
- **ImportStatement** – утверждение, определяющее ссылку на импортируемую онтологию;
- **OntologyIRIStatement** – утверждение, определяющее идентификатор онтологии;
- **VersionIRIStatement** – утверждение, определяющее идентификатор версии онтологии;
- **AxiomStatement** – утверждение, описывающее аксиому на языке OWL 2.

Обозначим такое преобразование как τ :

$$S_i = \tau(v_i),$$

а обратное преобразование – как τ^{-1} :

$$v_i = \tau^{-1}(S_i).$$

В этом случае сравнение онтологий сводится к простому сравнению конечных неупорядоченных множеств утверждений $S_i = \tau(v_i)$. Обозначим функцию, вычисляющую изменения между двумя множествами утверждений S_0 и S_i , как $\delta(S_0, S_i)$. Каждое изменение состоит из утверждения и операции («+» или «-»), применённой к нему. Введём дополнительные обозначения:

$\gamma(o, s)$ – изменение, представляющее операцию o над утверждением s ; $\omega(c)$ – операция, соответствующая изменению c : $\omega(\gamma(o, s)) = o$; $\zeta(c)$ – утверждение, соответствующее изменению c : $\zeta(\gamma(o, s)) = s$.

Изменения между S_0 и S_i можно вычислить следующим образом:

$$C_i = \delta(S_0, S_i) = \{\gamma(\langle -, \rangle, s) | s \in S_0 \setminus S_i\} \cup \{\gamma(\langle +, \rangle, s) | s \in S_i \setminus S_0\}.$$

Определим сигнатуру $\sigma(s)$ утверждения s как конечное неупорядоченное множество сущностей, на которые ссылается утверждение s . Сигнатура множества утверждений рассчитывается как объединение сигнатур отдельных утверждений:

$$\sigma(\{s_1, s_2, \dots, s_n\}) = \sigma(s_1) \cup \sigma(s_2) \cup \dots \cup \sigma(s_n).$$

Сигнатура изменения равна сигнатуре соответствующего утверждения:

$$\sigma(\gamma(o,s)) = \sigma(s).$$

Сигнатура множества изменений рассчитывается как объединение сигнатур отдельных изменений:

$$\sigma(\{c_1, c_2, \dots, c_n\}) = \sigma(c_1) \cup \sigma(c_2) \cup \dots \cup \sigma(c_n).$$

Конфликтующие изменения пользователя 1 можно найти по следующей формуле:

$$C_{c1} = \{c \in C_1 \mid \sigma(c) \cap \sigma(C_2 \setminus C_m) \neq \emptyset\}.$$

Согласно этой формуле, конфликтующими изменениями пользователя 1 считаются изменения, сигнатура которых пересекается с сигнатурой изменений пользователя 2, т. е. изменения, затрагивающие те же сущности, что и изменения пользователя 2. Аналогично конфликтующие изменения пользователя 2

$$C_{c2} = \{c \in C_2 \mid \sigma(c) \cap \sigma(C_1 \setminus C_m) \neq \emptyset\}.$$

Конфликтующие изменения не могут быть решены автоматически, поэтому в большинстве случаев при выполнении трёхстороннего слияния требуется участие пользователя. Обычно выполняет слияние пользователь, вносящий изменения последним. От него требуется выбрать из конфликтующих изменений $C_{c1} \cup C_{c2}$ те, которые необходимо сохранить. Обозначим эти изменения как C_s . Функцию, применяющую множество изменений C к множеству утверждений S , обозначим как $\varepsilon(S, C)$. Тогда итоговая версия онтологии, содержащая изменения обоих пользователей,

$$v_3 = \tau^{-1}(\varepsilon(\tau(v_0), C_m \cup C_{c1} \cup C_{c2} \cup C_s)).$$

Программная реализация

На рис. 3 показана диаграмма классов программы. Для работы с онтологиями OWL 2 использова-

лась библиотека OWL API [4]. Класс `OWL ontology` из этой библиотеки содержит объектное представление онтологии $\langle E, A, I, N, r_o, r_v, f \rangle$. Конструктор класса `ComparableOntology` принимает в качестве аргумента экземпляр v класса `OWL ontology` и строит из него множество утверждений $S_v = \tau(v)$, которое можно получить, вызвав метод `getStatements()`.

Алгоритм сравнения

Конструктор `ChangeSet (s1, s2)` выполняет сравнение множеств утверждений по следующему алгоритму.

Входные данные:

- S_1 – исходное множество утверждений;
- S_2 – изменённое множество утверждений.

Выходные данные:

- C – множество изменений.
1. $S \leftarrow S_1 \setminus S_2$
 2. $S^+ \leftarrow S_2 \setminus S_1$
 3. Для каждого $s \in S^-$
 4. $C \leftarrow C \cup \{\gamma(\langle -, s \rangle)\}$
 5. Для каждого $s \in S^+$
 6. $C \leftarrow C \cup \{\gamma(\langle +, s \rangle)\}$

Алгоритм классификации изменений

Классификацию изменений выполняет конструктор класса `ConflictFinder`.

Входные данные:

- $base = \tau(v_0)$ – множество утверждений исходной версии онтологии;
- $remote = \tau(v_1)$ – множество утверждений онтологии, изменённой другими пользователями;
- $local = \tau(v_2)$ – множество утверждений онтологии, изменённой локальным пользователем.

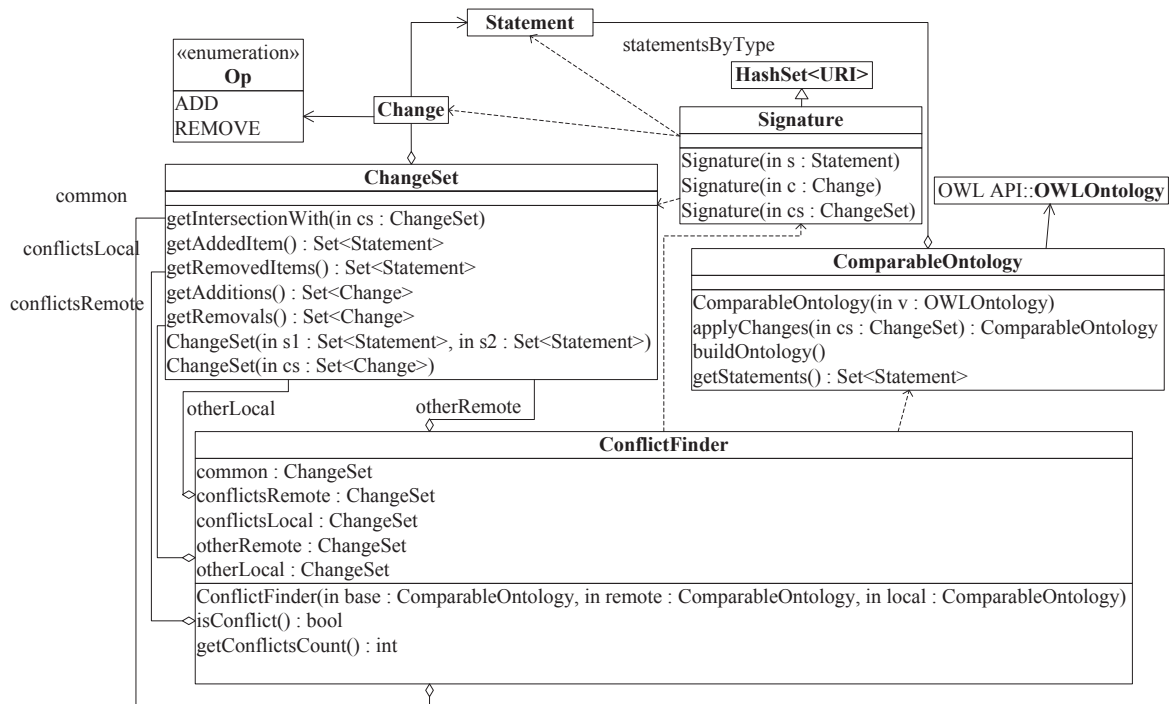


Рис. 3. Основные классы программы

Выходные данные:

- $common = C_m$ – совпадающие изменения;
 - $conflictsRemote = C_{c1}$ – конфликтующие изменения других пользователей;
 - $otherRemote = C_{o1}$ – остальные изменения других пользователей;
 - $conflictsLocal = C_{c2}$ – конфликтующие изменения локального пользователя;
 - $otherLocal = C_{o2}$ – остальные изменения локального пользователя.
1. $remoteChanges \leftarrow \delta(base, remote)$
 2. $localChanges \leftarrow \delta(base, local)$
 3. $common \leftarrow remoteChanges \cap localChanges$
 4. $remoteChanges \leftarrow remoteChanges \setminus common$
 5. $localChanges \leftarrow localChanges \setminus common$
 6. $remoteChangesSig \leftarrow \sigma(remoteChanges)$
 7. $localChangesSig \leftarrow \sigma(localChanges)$
 8. $conflictsRemote \leftarrow \emptyset$
 9. $conflictsLocal \leftarrow \emptyset$
 10. $otherRemote \leftarrow remoteChanges$
 11. $otherLocal \leftarrow localChanges$
 12. Для каждого $c \in remoteChanges$
 13. Если $localChangesSig \cap \sigma(c) \neq \emptyset$
 14. $conflictsRemote \leftarrow conflictsRemote \cup \{c\}$
 15. $otherRemote \leftarrow otherRemote \setminus \{c\}$
 16. Для каждого $c \in localChanges$
 17. Если $remoteChangesSig \cap \sigma(c) \neq \emptyset$
 18. $conflictsLocal \leftarrow conflictsLocal \cup \{c\}$
 19. $otherLocal \leftarrow otherLocal \setminus \{c\}$

Алгоритм применения изменений

Алгоритм применения изменений прост благодаря тому, что онтология, включая её атрибуты, представлена в виде множества утверждений. Алгоритм реализован в методе `applyChanges()` класса `ComparableOntology`.

Входные данные:

- $statements$ – множество утверждений;
- $changes$ – множество изменений.

Выходные данные:

- $statements$ – множество утверждений.
1. Для каждого $c \in changes$
 2. Если $\omega(c) = \langle + \rangle$
 3. $statements \leftarrow statements \cup \{ \zeta(c) \}$
 4. Иначе
 5. $statements \leftarrow statements \setminus \{ \zeta(c) \}$

Тест производительности

Обозначим $m(v_i, v_j) = |\delta(\tau(v_i), \tau(v_j))| / (|\tau(v_i)| + |\tau(v_j)|)$.

Для проверки производительности метода были сгенерированы тройки онтологий (v_0, v_1, v_2) со следующими характеристиками:

- $C_m = C_{c1} = C_{c2} = \emptyset$;
- $C_{o1} = C_1$;
- $C_{o2} = C_2$;
- $|\tau(v_0)| = |\tau(v_1)| = |\tau(v_2)|$;
- $m(v_0, v_1) = m(v_0, v_2) = m(v_1, v_2) = 0,02$.

Первые три характеристики обусловлены тем фактом, что выполнить слияние в автоматическом режиме возможно лишь при отсутствии конфликтов, которые потребовалось бы разрешать вручную при их наличии.

На рис. 4 показан график зависимости времени t слияния от количества утверждений N в онтологиях на вычислительной машине с процессором AMD A6–3410MX с тактовой частотой 1,6 ГГц и оперативной памятью типа DDR3 объёмом 8 Гб с пропускной способностью 667 МГц. Видно, что при больших значениях $|\tau(v)|$ время слияния растёт экспоненциально. Однако при очень больших значениях $|\tau(v)|$ объём оперативной памяти, требуемый для размещения четырёх версий онтологий, превышает доступный объём физической оперативной памяти, что приводит к использованию дискового пространства, вследствие чего время слияния значительно увеличивается.

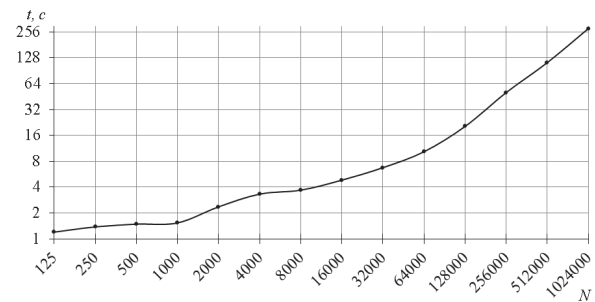


Рис. 4. График зависимости времени слияния от количества утверждений

Графический пользовательский интерфейс

Для разрешения конфликтов необходимо предоставить пользователю возможность выбирать изменения, которые он желает внести в итоговую версию онтологии. На рис. 5 показан графический пользовательский интерфейс программы слияния.

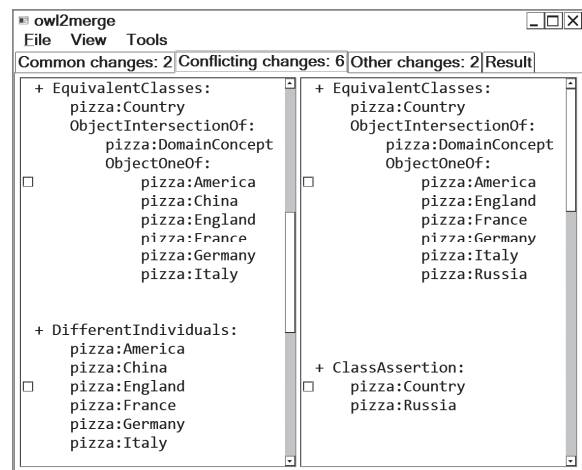


Рис. 5. Вкладка «Конфликтующие изменения» главного окна программы трёхстороннего слияния онтологий

Главное окно программы состоит из четырёх вкладок, отображающих совпадающие изменения, конфликтующие изменения, остальные изменения и результат – изменения, выбранные для внесения в итоговую версию онтологии. По умолчанию выбраны все совпадающие и другие неконфликтующие изменения, а конфликты предлагается разрешить пользователю.

Заключение

На основе проведенных исследований можно сделать следующие выводы:

1. Предложенный метод позволяет выполнять трёхстороннее слияние онтологий с учётом изменений формата онтологии, префиксов пространств имён, импортов и идентификаторов онтологии за приемлемое время.

2. Недостатком программной реализации является отсутствие возможности группировки изменений по сущностям, а также добавления новых изменений (не внесённых ни одним из пользователей). В дальнейшем предполагается усовершенствование программной реализации.

СПИСОК ЛИТЕРАТУРЫ

1. Motik B., Patel P.F., Parsia B. OWL 2 Web Ontology Language structural specification and functional style syntax // World Wide Web Consortium. 2009. URL: <http://www.w3.org/TR/owl2-syntax/> (дата обращения: 23.10.2012).
2. Carroll J. Matching RDF Graphs. 2001. URL: <http://www.hpl.hp.com/techreports/2001/HPL-2001-293.pdf> (дата обращения: 23.10.2012).

3. DeltaView // ESW Wiki. 2007. URL: <http://esw.w3.org/DeltaView> (дата обращения: 23.10.2012).
4. Horridge M., Bechhofer S. The OWL API: A Java API for Working with OWL 2 Ontologies // OWL Experiences and Directions: Proc. of the VI Intern. Workshop. – Chantilly, 2009. – V. 529. – P. 53–62.

Поступила 31.10.2012 г.

УДК 004.02:004.82

СЕМАНТИЧЕСКОЕ АННОТИРОВАНИЕ ДОКУМЕНТОВ В ЭЛЕКТРОННЫХ БИБЛИОТЕКАХ

Ле Хоай, А.Ф. Тузовский

Томский политехнический университет
E-mail: lehotomsk@yahoo.com

Рассматривается задача выполнения аннотирования документов в электронных библиотеках, использующих семантические технологии. Описываются причины и преимущества использования таких аннотаций. Предложен новый метод выполнения полуматематического аннотирования, и показаны результаты тестирования его программной реализации.

Ключевые слова:

Аннотирование документов, семантические технологии, семантическая аннотация, электронная библиотека организации, методы аннотирования.

Key words:

Document annotation, semantic technology, semantic annotation, enterprise digital library, annotation method.

Введение

В настоящее время большинство электронных ресурсов, в особенности веб-страницы, создается только для их использования людьми. Форматы их описания практически не включают формальных описаний знаний, содержащихся в этих документах. Формальное описание основного смысла документа в удобном для программной обработки формате является целью использования активно разрабатываемых в последнее десятилетие семантических технологий, таких как RDF, RDFS, OWL и SPARQL.

Электронные библиотеки представляют собой специализированные информационные системы, которые выполняют управление коллекциями электронных ресурсов (например, таких как текстовые документы, изображения, мультимедиа

файлы) с целью повышения эффективности использования содержащихся в них знаний некоторыми сообществами пользователей. Под семантическими электронными библиотеками (СЭБ) понимаются электронные библиотеки, использующие семантические технологии для организации всех процессов своей работы, таких как описание ресурсов, ведение каталогов, описание профилей пользователей, поиск и рекомендация ресурсов пользователям и т. п. Одной из важных функций семантических электронных библиотек является предоставление возможности аннотирования публикуемых ресурсов. В данной статье рассматривается проблема реализации такого подхода за счет перехода от человеко-читаемых к программно-обрабатываемым описаниям электронных ресурсов.

Аннотирование документов

В современной литературе существует разное понимание термина «аннотация ресурса». Обычно под аннотацией ресурса, понимаются некоторые метаданные (данные о данных) этого ресурса. Выделяют [1] следующие три вида аннотаций: неформальные, формальные и онтологические. Неформальная аннотация описывается на естественном языке и поэтому обычно не обрабатывается с помощью программ. Формальные аннотации составляются с использованием специальных языков (таких как XML и RDF), что позволяет выполнять их программную обработку. Если аннотация составляется на основе некоторой семантической модели (онтологии), описывающей основные понятия и отношения той предметной области, к которой относится описываемый ресурс, то она называется онтологической.

Формирование аннотаций ресурсов может выполняться с применением различных средств, таких как теги, наборы ключевых слов, наборы понятий и наборы триплетов. При использовании тегов аннотирование выполняется путем их добавления в тексты документов. Это делается для того, чтобы обрабатывающая программа могла определить формальный смысл выделенных с их помощью частей. Например, данный способ может использоваться для аннотирования веб-страниц, в которые, кроме HTML тегов, включаются и микроформаты RDFa [2]. Стандарт RDFa позволяет использовать URI идентификаторы для набора атрибутов, а также набор общепринятых словарей, таких как hCalendar, hCard или hAtom.

Другим способом аннотирования ресурсов является использование набора ключевых слов. Такое средство чаще применяется в электронных библиотеках для аннотирования научных статей. Ключевые слова для конкретной статьи выбираются их авторами, а затем хранятся в базе данных для поддержки процесса поиска. Неоднозначность семантики ключевых слов приводит к использованию наборов понятий. Такие понятия могут выбираться из онтологий или тезаурусов, описываемых с помощью языка RDFS или OWL. При этом каждое понятие представляется в виде URI идентификатора [3].

Третьим способом аннотирования ресурса является его описание с помощью наборов утверждений, имеющих следующий формат: субъект–предикат–объект, которые также называются триплетом. Данный способ является самым новым, и его использование еще в достаточной степени не исследовано. В данной статье как раз и рассматривается подход к аннотированию ресурсов с использованием данного способа.

Аннотирование ресурсов представляет собой очень важную, но также и очень трудную и трудоемкую задачу. Следует отметить, что аннотирование является начальным этапом применения семантических технологий, а в настоящее время существует большое количество не аннотированных электронных ресурсов.

Онтологический подход к аннотированию документов

Основная часть информации (более 80 %) современных организаций содержится в виде текстов на естественных языках в бумажном и электронных форматах [4]. В связи с этим одной из наиболее сложных задач в разработке СЭБ является разработка методов и программных средств для составления достаточно точных метаданных для описания содержания текстовых документов.

Описание ресурсов в СЭБ организации выполняется с использованием набора специально разработанных онтологий, описанных на языке RDFS или OWL. В результате выполнения аннотирования для каждого ресурса создается семантическое метаописание, состоящее из набора триплетов, в состав которых могут входить контекстные и контентные семантические метаданные. Контекстные метаданные аннотируемого объекта представляют собой утверждения о его связи с другими объектами, понятиями онтологий, а контентные метаданные – утверждения о знаниях, содержащихся в самом объекте. Кроме этого, контентными метаданными ресурсов, как правило, является набор триплетов, созданный на основе онтологии, которая описывает ту предметную область, с которой связан данный ресурс. Именно контентные метаданные играют важную роль и представляют основную ценность для обработки и повторного использования, а также широко используются в среде организации [5].

Для составления контентных метаданных, содержащих наборы триплетов, используются онтологические модели.

Определение 1 (онтологическая модель). *Под онтологической моделью (онтологией) O понимается знаковая система $\langle C, P, I, L, T \rangle$, где C – множество элементов, которые называются понятиями; P – множество элементов, называемых свойствами (двуместными предикатами); I – множество экземпляров понятий; L – множество текстовых меток или значений понятий и свойств; T – частичный порядок на множестве C и P .*

С помощью онтологий для ресурса могут создаваться их семантические аннотации.

Определение 2 (семантическая аннотация). *Семантической аннотацией ресурса является его формальное описание в виде набора кортежей вида (s, p, o, v) , где s – URI идентификатор субъекта триплета; p – URI идентификатор предиката триплета; o – URI идентификатор некоторого объекта или конкретное значение некоторого типа, а v – весовой коэффициент значимости данного триплета. При этом субъекты, отношения и объекты выбираются из некоторого набора семантических моделей (онтологий). Начальные части таких кортежей, вида (s, p, o) , называются триплетами.*

Онтологический подход к аннотированию заключается в формировании метаданных документа с использованием элементов онтологий [4]. На основе онтологической модели создаются контентные метаданные документа.

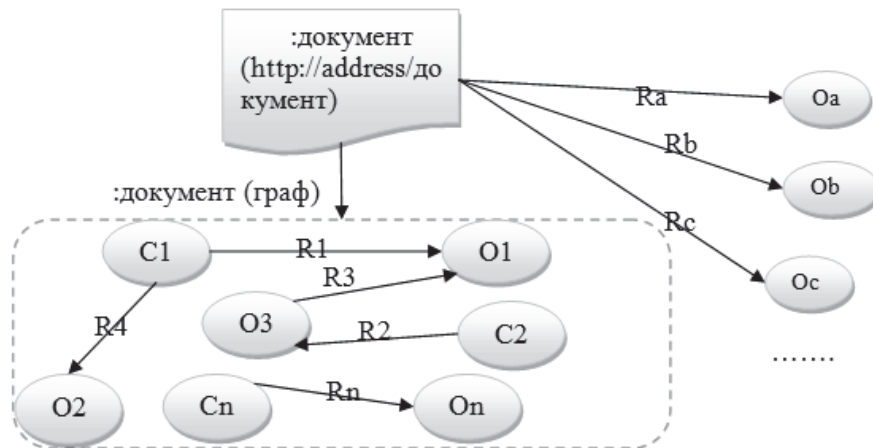


Рис. 1. Пример семантических метаданных документа

Определение 3 (**Контентные метаданные документа**). Контентные метаданные документа – это набор семантических утверждений (триплетов) $M_c = \{t_1, t_2, \dots, t_m\}$, описывающих основные знания, содержащиеся в документе. Каждый триплет t_i имеет вид $t_i = (c, r, o, v)$, где c – это субъект утверждения ($c \in C \cup I$); o – объект утверждения ($o \in C \cup I$); r – отношение между субъектом и объектом ($r \in P$), а v – весовой коэффициент, который оценивает значимость данного утверждения.

Утверждения на основе онтологии O могут иметь следующий вид: $\langle C, P, C \rangle$, $\langle I, P, I \rangle$, $\langle I, P, C \rangle$, $\langle I, P, L \rangle$, где C , P , I – это уникальные URI-идентификаторы понятий, предикатов или экземпляров.

Как уже было отмечено, аннотирование может выполняться для создания контентных и контекстных метаданных. Но контекстные метаданные сохраняются в базе знаний как триплеты, субъектом которых является идентификатор аннотируемого документа, а контентные метаданные – как триплеты, составляющие некоторый подграф, в который входят идентификатор основных понятий, предикатов и экземпляров, являющихся важными для описания смысла данного документа (рис. 1).

Результаты аннотирования документа пополняют базу знаний не только новыми экземплярами, но и контентными метаописаниями документа. Это приводит к новым возможностям обработки документов, таким как: **семантический поиск документа, формирование рекомендаций на основе профиля пользователя, автоматическая категоризация документов по заданной иерархии рубрик.**

Обзор методов семантического аннотирования документов

Аннотирование может выполняться разными способами: вручную, полуавтоматически или автоматически. При ручном способе аннотирования составлением метаописания документа занимается специалист. Однако программная система может оказывать ему помощь, например, показывая онтологии и имеющиеся экземпляры, выполняя проверку их правильности. При полуавтоматическом

методе аннотирования система автоматически составляет начальный вариант аннотации, а затем специалист может проверить и выполнить ее корректировку. Ручной и полуавтоматический способы трудно использовать при большом количестве документов, для документов больших размеров, а также при недостатке специалистов, которые могут своевременного выполнять такую работу. В этих случаях возникает потребность в автоматическом методе аннотирования.

Ручное Аннотирование. В электронной библиотеке организации электронные ресурсы (в основном документы) создаются и формируются специалистами, которые могут создавать для них достаточно качественные аннотации. Для выполнения такой работы хорошо подходят инструменты ручного аннотирования.

При разработке инструментов или платформ ручного аннотирования рассматриваются следующие рекомендации [2]:

1. Поддержка ссылок к экземплярам и понятиям онтологии с помощью URI идентификаторов.
2. Создание инструментов на основе веб-технологий для обеспечения возможности аннотирования ресурсов различными группами пользователей.
3. Разрешение формировать аннотации различных типов (экземпляры, понятия и отношения).
4. Возможность работы с большим количеством онтологий и онтологиями, содержащими большое количество понятий и отношений.
5. Возможность аннотирования различных форматов документов, например таких, как HTML, PDF, XML, изображения и мультимедиа файлы.

Одной из известных комплексных платформ семантического аннотирования является CREAM [6]. Данная платформа работает как отдельная система со своей базой RDF-триплетов (созданной с помощью RDF-робота), которая пополняется информацией, содержащейся в веб-страницах. CREAM позволяет использовать созданные экземпляры для повторного применения, чтобы избе-

жать избыточности, а также предоставляет редактор содержания аннотируемого документа. Ручным редактором аннотаций платформы CREAM является инструмент OntoMat, работающий в веб-браузере. Данный редактор позволяет пользователям расширять онтологию новыми данными с помощью удобного интерфейса и обеспечивает три режима аннотирования: по написанию утверждения (typing statements), по разметке (markup), по управлению версиями (authoring).

Примером другого подхода к ручному аннотированию является система SMORE [7]. Она предоставляет такие же средства аннотирования, как и OntoMat, но работает совместно с редактором онтологий SWOOP [8].

Система Zemanta [9] представляет собой онлайн инструмент аннотирования для блогов и содержания электронной почты. Zemanta позволяет авторам использовать рекомендуемые лексические метки, чтобы пользователям становилась понятной связь между различными сообщениями блогов. Данный инструмент находит все страницы, связанные с заданными метками и предоставляет возможность выбирать ссылки для используемых меток, а также позволяет добавлять ссылки в виде изображений непосредственно в текст документа.

Полуавтоматическое и автоматическое аннотирование. В отличие от ручного метода, полуавтоматический метод создает частичную аннотацию документа с помощью анализа естественного языка документа. Извлечение информации является основной технологией для связывания неструктурированного текста с формальными описаниями, содержащимися в онтологиях. При извлечении информации часто используются такие компоненты обработки естественного языка, как разметчик частей речи (part-of-speech tagger), морфологический анализатор, сканеры именованных сущностей, полный (или поверхностный) синтаксический анализ и семантическая интерпретация.

Существует два подхода к извлечению информации:

1. На основе использования наборов правил, разработанных специалистами по лингвистике, которые создают словари и описывают правила извлечения требуемой информации.
2. На основе использования методов машинного обучения, позволяющих выполнять автоматическое обучение для решения различных задач извлечения информации.

Преимуществом первого подхода является то, что не требуется составлять обучающие выборки данных для создания правил и описания знаний предметной области. Благодаря созданным словарям и правилам системы аннотирования могут работать быстрее систем, разработанных с помощью машинного обучения. Для этого подхода обычно требуется создание коллекции проаннотированных человеком обучающих данных для достижения высокой точности. С одной стороны, на обучение такой системы требуется меньше затрат, чем на соз-

дание словарей и наборов правил извлечения информации из текста, но, с другой стороны, могут возникать проблемы, связанные с низким качеством составления аннотаций. В связи с этим выбор конкретного подхода зависит от характеристики системы электронной библиотеки организации. В [2] описаны и другие системы автоматического аннотирования, такие как AeroDAML, Amilcare и Melita.

Рассмотренные инструменты позволяют аннотировать документ понятиями и экземплярами онтологий, а также создавать новые экземпляры или понятия, содержащиеся в обрабатываемом документе. В отличие от описанных подходов, в данной работе предлагается метод аннотирования документов, который позволяет не только создавать новые экземпляры, но и составлять триплеты, описывающие основные знания, содержащиеся в аннотируемом документе.

Метод полуавтоматического семантического аннотирования

В общем виде для составления триплета аннотирования документа необходимо вручную выбрать субъект, определять его предикат (отношение), на основе его описания в онтологии, а затем выбрать связанный с ним объект. Созданный триплет сохраняется в базе знаний.

Выбор субъектов и объектов триплетов выполняется в ходе решения задач поиска кандидатов и преодоления многозначности.

С учетом изложенного выше, задача семантического аннотирования может быть структурирована следующим образом:

1. Выбор нужных понятий. Аннотирование документа выполняется в соответствии с некоторой онтологией предметной области, и при этом специалист имеет возможность ограничить некоторые категории понятий в онтологии предметной области (желаемые понятия) для преодоления многозначности.
2. Поиска кандидатов. Поиск в базе знаний основных кандидатов (понятий, предикатов или экземпляров). Для этого необходимо в документе находить термины (слова или словосочетания), которые совпадают с метками экземпляров или понятий онтологии или близки им в соответствии с некоторой оценкой семантической близости.
3. Преодоление многозначности. Данный шаг заключается в том, что из аннотации должны быть исключены все нерелевантные кандидаты. Эти кандидаты сходны с терминами документа по текстовым меткам, но в действительности не описывают содержание данного документа.

Решение задачи поиска кандидатов. Обозначим набор понятий и экземпляров онтологии, хранящихся в базе знаний как $M_{CE} = \{ce_1, ce_2, \dots, ce_n\}$. Каждый элемент (ce) может иметь текстовые метки для их обозначения на разных естественных языках. Набор текстовых меток соответствующих элемен-

тов M_{ce} некоторой онтологий O обозначим как $M = \{m_1, m_2, \dots, m_n\}$, каждая метка может быть представлена в виде набора токенов в результате токенизации (tokenization). Под токенизацией понимается процесс разделения текста, содержащего метки экземпляров, понятий или документов, на последовательность токенов, при этом в качестве разделителя может использоваться знак пробела, а программа, выполняющая токенизацию, называется токенизатором.

Таким образом, аннотируемый документ (или набор документов D) может быть представлен в виде множества токенов. Для учета грамматики естественных языков возникает необходимость выполнять нормализацию токенов. Существуют два типа систем, выполняющих нормализацию токенов: лемматизатор (lemmatizer) и стеммер (stemmer). Как правило, лемматизатор токена возвращает его исходную форму, а стеммер – его корень с помощью правил отсечения. При нормализации могут быть удалены стоп-слова (шумовые слова), которые не несут никакой смысловой нагрузки.

Решение задачи поиска кандидатов может быть разделено на следующие шаги:

- преобразование D и M в наборы нормализованных токенов, обычно отсортированных в алфавитном порядке;
- поиск в D набора токенов для каждой метки $m_i \in M$.

В результате выполнения этих шагов будет получен набор кандидатов M_K ($M_K \subset M_{ce}$), лексические метки которых содержатся в документе [10].

Для эффективного решения задачи поиска кандидатов предлагается на основе множества D создать базу индексов $dbindx$, содержащих термины (токены), к которой можно обращаться с запросами для поиска описания любой метки m_i . Способ создания таких индексов, состоящих из терминов документов, можно найти в [11].

С учетом вышесказанного можно с помощью псевдокода составить следующий алгоритм функции поиска кандидатов:

поиск_кандидатов

Вход: M_{ce}, D .

Выход: M_K .

Начало

список $M_K = []$; // пустое множество

база_индексов $dbindx = \text{создание_базы_индексов}(D)$;

Цикл для каждой $ce_i \in M_{ce}$ выполнять

начало цикла

текстовые_метки $t_i = \text{метка}(ce_i)$;

// r – результат поиска – числовая оценка

оценка $r = \text{поиск_метки}(t_i, dbindx)$;

Если $r > 0$, то добавить ce_i к M_K ;

конец цикла.

Конец.

Данный алгоритм создает базу индексов $dbindx$ из аннотируемого документа с помощью функции *создание_базы_индексов()*. После этого для каждого экземпляра или понятия ce_i извлекаются метки

m_i с помощью функции *метка()*. На следующем шаге выполняется поиск m_i в $dbindx$, и если он будет найден (т. е. $r > 0$), то понятие ce_i добавляется к списку кандидатов M_K .

При практической реализации для извлечения текста из аннотируемого документа требуется определить его формат (например, Microsoft Word, PDF или HTML). Также нужно указать естественный язык, на котором написан данный документ и используемые для его аннотирования метки (экземпляры, понятия и отношения). После этого создается база индексов, учитывающая грамматику используемого языка и использующая соответствующие ему токенизаторы.

Решение задачи преодоления многозначности. Существуют два подхода к автоматическому решению данной задачи: с помощью измерения семантической близости и с помощью измерения популярности [12].

Идея подхода измерения семантической близости для решения многозначности заключается в том, что для набора найденных кандидатов с использованием онтологий вычисляются их семантические близости с понятиями, подходящими для аннотирования документа. В результате этого система может выбирать тех кандидатов, близость которых с желаемыми понятиями больше некоторого порогового значения. В данном методе необходимо вычислять близость между понятиями и близость между понятиями и экземплярами. Один из возможных способов оценки такой близости описан в [3].

Допустим, что имеется конечный набор кандидатов M_K из онтологии O . Также имеется конечный набор желаемых понятий для аннотирования документа $M_c = \{c_i \in C \subset O\}$. Тогда релевантными будут считаться кандидаты из M_K , удовлетворяющие следующему условию:

$$\text{Sem}(ce_i, M_c) = \max(\text{Sem}_{c_j \in M_c}(ce_i, c_j)) > \varepsilon \forall \varepsilon > 0,$$

где Sem – семантическая близость и ε – установленное пороговое значение.

В результате объединения описанных выше задач получается следующий псевдокод полного алгоритма для функции поиска кандидатов для заданного документа:

поиск_кандидатов

Вход: M_{ce}, M_c, D .

Выход: M_K .

Начало

список $M_K = []$; // пустое множество

база_индексов $dbindx = \text{создание_базы_индексов}(D)$;

Цикл для каждой $ce_i \in M_{ce}$ выполнять

начало цикла

текстовые_метки $t_i = \text{метка}(ce_i)$;

// r – результат поиска – числовая оценка

оценка $r = \text{поиск_метки}(t_i, dbindx)$;

Если $r > 0$ и $\text{Sem}(ce_i, M_c) > \varepsilon$ то добавить ce_i к M_K ;

конец цикла.

Конец.

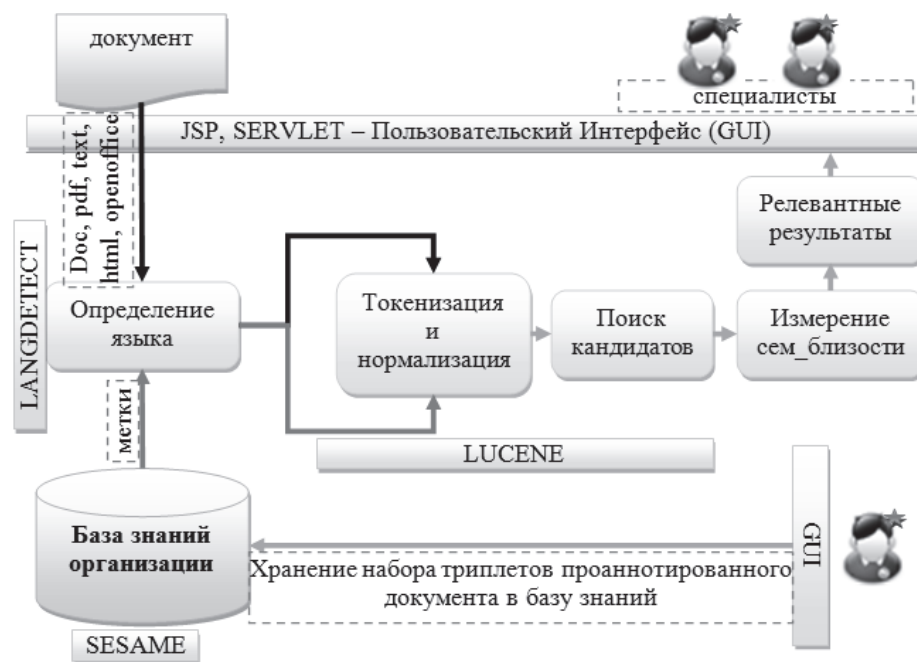


Рис. 2. Структура системы – компонента полуавтоматического аннотирования

Реализация системы полуавтоматического аннотирования

На рис. 2 показана обобщенная структура разработанной программной системы – полуавтоматического инструмента семантического аннотирования, а также другие использованные компоненты. Разработанная программа создана на основе технологии JSP и Servlet. Она может использоваться либо самостоятельно, либо как HTML-тег на странице JSP для аннотирования информационных объектов ЭБ.

В качестве системы управления базой знаний использовалась система **SESAME** [13], представляющая собой веб-сервис с открытым исходным кодом (на языке Java) для хранения триплетов описания всех понятий и их экземпляров онтологий организации.

Для определения языка, используемого в документе, и меток каждого понятия или экземпляра используется пакет **LangDetect** [14]. Данный пакет является библиотекой с открытым исходным кодом (на языке Java), которая позволяет идентифицировать большое количество естественных языков, в том числе и русский язык. Результат работы данного пакета используется для вызова конкретных компонентов анализа текста.

Самой важной частью данной системы является **Lucene** [15] – большая библиотека (на языке Java) полнотекстового поиска. Она позволяет выполнить индексацию документов различных форматов (с помощью специальных модулей). Одной из особенностей библиотеки **Lucene** является использование стеммера для нормализации токенов, написанных на русском языке, а также возможность хранения текста без стемминга.

Для поиска кандидатов и решения задачи многозначности выполняются следующие действия:

1. Специалист вначале загружает документ или набор документов, после чего система определяет формат документа для вызова соответствующего пакета с целью извлечения текста документа.
2. На основе извлеченного текста документа система определяет используемый в нем язык для вызова правильного анализатора текста из библиотеки **Lucene**.
3. Текст документа передается в модули **Lucene** для токенизации и нормализации. В результате этого из текста документа создается инвертированный файл индексов, хранящихся в оперативной памяти для повышения быстродействия алгоритма и используемых в качестве базы индексов для дальнейшего поиска.
4. Метки понятий онтологии и их экземпляров обрабатываются так же, как было описано выше (без создания базы индексов). После этого выполняется их поиск в базе индексов документа, созданной на шаге 3. Понятия и экземпляры, метки которых находятся в базе индексов, заносятся в список кандидатов для последующей обработки.
5. В результате поиска кандидатов получается список экземпляров и понятий, для которых выполняется оценка семантической близости со списком желаемых понятий, с целью исключения нерелевантных.
6. Релевантные кандидаты показываются специалисту для выполнения ручного аннотирования. После этого специалист имеет возможность составлять триплеты, включающие найденных кан-

Аннотация

Контентные триплеты

А.Ф. Тузовский → Интерес-КС → Семантические технологии Ле Хоай → Интерес-КС → Семантические технологии

А.Ф. Тузовский → Знакомство → Ле Хоай

Аннотирование

Choose File Аннотировать...final.doc файл загружен! Загрузить

электронная библиотека
type: Concept

Семантические технологии
type: Concept

Семантическое аннотирование
type: Concept

А.Ф. Тузовский
type: Author

Ле Хоай
type: Author

Метаописание
 type: Concept

☐
 Объект

Контекстные триплеты

Семан... → время создания → 2013011116572146 Семан... → путь К.Файлу → missing Семан... → размер файла → 125.0

Семан... → Авторы → А.Ф. Тузовский Семан... → Авторы → Ле Хоай

Семан... → Домены → Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

Семан... → название → Семантическое аннотирование ресурсов Семан... → год издания → 2013

Рис. 3. Интерфейс системы полуавтоматического аннотирования

дидатов, для онтологического аннотирования загруженного документа.

Тестирование разработанного метода. Созданная система полуавтоматического аннотирования используется в составе разрабатываемой семантической электронной библиотеки (SemDL) для аннотирования электронных ресурсов, рубрик и профиля пользователя. Интерфейс данной системы показан в рис. 3.

Для проверки качества разработанного метода и программной системы было выполнено семантическое аннотирование данной статьи. В базе знаний хранятся экземпляры созданных понятий (Author и Concept). Система реализует поиск меток этих объектов в загруженном документе и выдает список рекомендуемых объектов, как показано на рис. 3.

С помощью данного инструмента возможно аннотировать документ набором триплетов с использованием значений (чисел, строк и дат, в качестве значения объекта триплета). Кандидаты экземпляров и понятий выводятся в отдельной области окна формирования триплетов.

Все созданные триплеты показываются в специальном окне программы, и специалист может выполнять их редактирование: удаление, изменение субъектов или объектов.

Выводы

В статье предложен новый подход к составлению семантических аннотаций электронных документов в виде набора триплетов и описаны разработанные алгоритмы для решения задачи поиска кандидатов и преодоления многозадачности при аннотировании. Для выполнения полуавтоматического аннотирования была разработана программная система, которая активно используется в разработке семантической электронной библиотеки. Использование данного подхода позволяет понизить трудоемкость составления семантических аннотаций и повысить их качество. Создание семантических аннотаций позволяет разрабатывать новые эффективные методы решения таких задач, как семантический поиск, автоматическая категоризация ресурсов и формирование рекомендаций.

СПИСОК ЛИТЕРАТУРЫ

1. Oren E., Hinnerk Möller K., Scerri S., Handschuh S., Sintek M. What are Semantic Annotations? URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.97.7985&rep=rep1&type=pdf> (дата обращения: 21.10.2012).
2. Domingue J., Fensel D., Hendler J.A. Handbook of Semantic web Technologies. — Heidelberg; Dordrecht; London; N.Y.: Springer, 2011. — 1077 p.
3. Saša Nešić, Mehdi Jazayeri, Fabio Crestani, Dragan Gašević. Concept-Based Semantic Annotation, Indexing and Retrieval of Office-Like Document Units. 2010. URL: http://www.old.inf.usi.ch/file/pub/56/tech_rep.pdf (дата обращения: 21.10.2012).
4. Черный А.В., Тузовский А.Ф. Развитие информационной системы организации с использованием семантических технологий // Знания—Онтологии—Теория: Матер. Всеросс. конф. с междунар. участием. — Новосибирск, 20–22 октября 2009. — Новосибирск: ЗАО «РИЦ Прайс-Курьер», 2009. — Т. 2. — С. 52–59.
5. Тузовский А.Ф. Формирование семантических метаданных для объектов системы управления знаниями // Известия Томского политехнического университета. — 2007. — Т. 310. — № 3. — С. 108–112.
6. Handschuh S., Staab S. Authoring and Annotation of Web Pages in CREAM. 2002. URL: <http://www2002.org/CDROM/refereed/506/> (дата обращения: 21.10.2012).
7. SMORE — Create OWL Markup for HTML Web Pages. 2005. URL: <http://www.mindswap.org/2005/SMORE/> (дата обращения: 21.10.2012).
8. SWOOP — A Hypermedia-based Featherweight OWL Ontology Editor. 2004. URL: <http://www.mindswap.org/2004/SWOOP/> (дата обращения: 10.21.2012).
9. Zemanta. URL: <http://www.zemanta.com/> (дата обращения: 21.10.2012).
10. Ле Х.Х. Разработка электронных библиотек на основе семантических технологий // Научно-технический вестник Поволжья. — 2012. — № 3. — С. 138–145.
11. Manning C.D., Raghavan P., Shutze H. An Introduction to information retrieval. — Cambridge: Cambridge University Press, 2009. — 569 p.
12. Navigli R. Word sense disambiguation: a survey // ACM computing surveys. — 2009. — V. 41. — № 2. — P. 1–69.
13. Home of Sesame. URL: <http://www.openrdf.org/index.jsp> (дата обращения: 21.10.2012).
14. Language-detection. URL: <http://code.google.com/p/language-detection/> (дата обращения: 21.10.2012).
15. Paul Th. The Lucene Search Engine. 2004. URL: <http://www.java-ranch.com/journal/2004/04/Lucene.html> (дата обращения: 21.10.2012).

Поступила 16.01.2013 г.

УДК 004.942, 004.652.5

ФОРМАЛЬНАЯ МОДЕЛЬ СТРУКТУРЫ ВЗАИМОСВЯЗЕЙ РАЗНОТИПНЫХ ОБЪЕКТОВ ПРОЕКТИРОВАНИЯ

А.А. Вичугова, В.Н. Вичугов, Г.П. Цапко

Томский политехнический университет
E-mail: anya@aics.ru

В рамках развития существующих теоретических и практических положений технологий информационной поддержки жизненного цикла изделий поставлена задача разработки метода, позволяющего структурировать сущности, создаваемые при проектировании высокотехнологичной продукции на примере радиоэлектроники. Составлены теоретико-множественные модели, описывающие состав и взаимозависимости разнотипных объектов проектирования: изделие, его информационные модели, электронная структура и конструкторская документация. С использованием объектно-ориентированного подхода выполнено концептуальное проектирование базы данных: определен набор атрибутов, характеризующих стадии жизненного цикла рассматриваемых объектов проектирования, и предложена формальная информационная модель структуры их взаимосвязей в виде UML-диаграммы классов.

Ключевые слова:

Структурирование разнотипных взаимосвязанных объектов, объектно-ориентированный подход, теоретико-множественные модели.

Key words:

Structuring different related objects, object-oriented approach, set-theoretic models.

Введение

Проектирование высокотехнологичной продукции с длительным сроком активного существования (глубоководные и космические аппараты, искусственные человеческие органы и т. д.) представляет собой целый комплекс процессов со сложной структурой взаимосвязей и временной длительностью. Эта деятельность сопровождается большим количеством информации в виде инфор-

мационных моделей изделия (ИМИ), конструкторской документации (КД) и электронной структуры изделия (ЭСИ). Готовность изделия к производству определяется состоянием КД на данное изделие. Под качеством КД понимается отсутствие ошибок в описании характеристик изделия по нормативно-техническим стандартам. Первоисточником данных для КД является совокупность элементов ЭСИ, формируемой на основе файлов

ИМИ, разработанных в различных системах автоматизированного проектирования (САПР). Проектирование изделия считается завершенным в случае окончательно сформированной ЭСИ и утвержденной КД.

Для обозначения создаваемых при проектировании сущностей (Изделие, ИМИ, КД, ЭСИ) в работе используется термин «объекты проектирования», а термин «управление жизненным циклом объектов проектирования» — для обозначения следующей совокупности действий:

- структурированное хранение большого объема данных;
- автоматизированное изменение разнотипных объектов от исходного состояния к целевому с учетом их взаимозависимости;
- контроль за действиями участников процессов проектирования и обработкой информации.

Сегодня для решения всех вышеперечисленных задач применяются технологии информационной поддержки жизненного цикла (ЖЦ) изделий, реализованные в системах управления данными (СУД) и САПР. Объединение СУД и САПР в единый программно-аппаратный комплекс представляет собой интегрированную информационную среду (ИИС).

Идея организации ИИС широко исследуется с 2000-х гг. российскими и зарубежными учеными [1]. Однако проведенный анализ существующих стандартов, методов, алгоритмов и программных решений для выполнения проектных работ по разработке изделий в рамках ИИС показал, что они не позволяют в полной мере отразить особенности проектирования высокотехнологичной продукции с длительным сроком активного существования. К этим особенностям относятся:

- итеративность процессов проектирования в соответствии со стадиями испытаний и сложность структуры влияния разнотипных взаимозависимых объектов друг на друга на различных стадиях ЖЦ;
- необходимость интеграции различных САПР, в которых выполняется разработка частей проектируемого изделия, при условии агрегации всех характеристик одного элемента ЭСИ в СУД и отсутствия дублирования данных.

Существующие нормативные документы регламентируют основные термины и определения ЖЦ продукции, но не содержат готовых решений по структуре взаимосвязей разнотипных объектов проектирования и динамике изменения их состояний. Для обеспечения целостности данных об объектах проектирования, структурированного хранения информации и ее автоматизированной обработки необходимо наличие правил управления разнотипными объектами проектирования с учетом их взаимозависимости. В связи с этим возникает необходимость поиска способа, позволяющего получить формализованное описание структуры взаимосвязей исследуемых разнотипных сущностей.

Метод формализации структуры взаимосвязей объектов проектирования

Таким образом, метод формализации структуры взаимосвязей объектов проектирования должен обеспечивать возможность использовать модели, полученные в результате его применения, в качестве алгоритмической основы для разработки программного обеспечения средств управления жизненным циклом объектов проектирования в ИИС. Разрабатываемый метод должен включать математические положения и возможность использования полученных математических выражений в качестве основы для описания предметной области с применением существующих нотаций моделирования. Кроме того, согласно цели настоящей работы, необходимо установить соответствие положений предлагаемого метода и подходов к разработке программного обеспечения. Итак, метод формализации структуры взаимосвязей объектов проектирования должен включать следующие составные части:

- математическая основа;
- концептуальное проектирование структуры программного обеспечения с точки зрения объектно-ориентированной типизации данных;
- формальные модели объектно-ориентированной типизации данных в рамках структуры разрабатываемого программного обеспечения.

Согласно положениям, описанным в [1], СУД выполняет роль инструмента, реализующего автоматизированное управление жизненным циклом изделий. Поэтому возникает задача концептуального проектирования базы данных СУД. Концептуальное проектирование предполагает структуризацию данных в соответствии с базовыми положениями объектно-ориентированного подхода [2]. В свою очередь, в основе объектно-ориентированного подхода лежит теория систем и теория множеств. Таким образом, для реализации цели настоящего исследования, связанной с разработкой моделей и алгоритмов управления высокотехнологичной продукцией, на этапе ее проектирования необходимо формализовать отношения между разнотипными взаимозависимыми объектами с применением теории множеств. С использованием аппарата теории множеств, основные положения которой изложены в работах Б. Больцана, Б. Рассела, Э. Цермело и др., формализованы исследуемые объекты проектирования (Изделие, ИМИ, КД, ЭСИ) и связанные с ними сущности.

Пусть OP — множество объектов проектирования, PRJ — множество проектов, $PROD$ — множество изделий, ITM — множество элементов ЭСИ, DOC — множество документов, DM — множество ИМИ, KD — множество конструкторских документов. ИМИ и КД фактически являются документами: $DOC = \{DM, KD\}$.

Рассматривая высокотехнологичную продукцию на примере отрасли космического приборостроения, следует детализировать множество элементов ЭСИ по видам изделий: детали, сбороч-

ные единицы, комплексы и комплекты. Между этими видами изделий существуют следующие отношения:

- в состав сборочной единицы могут входить детали, комплекты и другие сборочные единицы;
- в состав комплекта могут входить детали, сборочные единицы и другие комплекты;
- в комплекс могут входить детали, сборочные единицы, комплекты, а также другие комплексы.

Кроме того, согласно выделению разделов спецификации, изделие и его составные части также принято делить на:

- стандартные изделия (изделия, примененные по межгосударственным, государственным и отраслевым стандартам, а также стандартам предприятий);
- прочие изделия (изделия, примененные по техническим условиям, и импортные покупные изделия, примененные по сопроводительной технической документации зарубежных изготовителей/поставщиков);
- материалы (все материалы, непосредственно входящие в специфицируемое изделие, иногда их называют «изделия из материалов»).

В рамках данной классификации электрорадиоизделия (ЭРИ), которые являются основой радиоэлектронной аппаратуры (РЭА), относятся к прочим изделиям. Таким образом, множество ЭСИ имеет неоднородный характер, поэтому можно записать

$$ITM = \{ASM, PART, CT, CX, STPT, OTPT, MAT\},$$

где ASM означает множество сборочных единиц; $PART$ – множество деталей; CT – множество комплектов; CX – множество комплексов; $STPT$ – множество стандартных изделий; $OTPT$ – множество прочих изделий; MAT – множество изделий из материала.

Совокупность элементов ЭСИ образуют изделие: одному элементу множества изделий соответствует несколько элементов множества ЭСИ. Поэтому можно записать, что изделие A является сюръективной функцией относящихся к нему элементов множества ЭСИ:

$$f_A : ITM_A \rightarrow \{prod_A\}, \quad (1)$$

где $prod_A \in PROD$ означает изделие A , конкретный элемент множества изделий $PROD = \{prod_A, prod_A, \dots, prod_A\}$, $prod_A \in PROD$, ITM_A – множество элементов электронной структуры изделия A ; $ITM_A \subseteq ITM$, $ITM_A = \{itm_{i,A}\}$, $itm_{i,A} \in ITM_A$ – конкретный элемент множества ITM_A , $i = \overline{1, n}$; n – количество элементов электронной структуры изделия A .

Используемый в настоящей работе термин «объекты проектирования» означает изделие, элементы его электронной структуры и описывающую их информацию в виде файлов информационных моделей и конструкторских документов. Поэтому можно записать

$$OP = \{PROD, ITM, DM, KD\}.$$

В рамках одного проекта $prj \in PRJ$ может создаваться одно или несколько изделий, каждое из которых описывается файлами информационных моделей $dm \in DM$ и конструкторских документов $kd \in KD$. Учитывая (1) и (2), можно записать выражение, описывающее частный случай множества объектов проектирования изделия A в рамках проекта Z :

$$OP_{Z,A} = \{prj_Z, prod_{Z,A}, \{itm_{i,A}\}, \{dm_{j,A}\}, \{kd_{k,A}\}\},$$

где n – количество элементов электронной структуры изделия A , $i = \overline{1, n}$, m – количество информационных моделей изделия A , $j = \overline{1, m}$, p – количество конструкторских документов по изделию A , $k = \overline{1, p}$, $prj_Z \in PRJ$ – проект Z , $prod_{Z,A} \in PROD$ – разрабатываемое в рамках проекта Z изделие A .

Первоисточником данных для одного элемента ЭСИ является один или несколько объектов ИМИ, поэтому можно записать, что множество элементов ЭСИ является сюръективной функцией множества ИМИ:

$$g_A : DM_A \rightarrow \{itm_A\}. \quad (3)$$

В свою очередь, описание состава и иерархии элементов ЭСИ представляет собой КД. Таким образом, множество КД является сюръективной функцией множества ИТМ:

$$y_A : ITM_A \rightarrow \{kd_A\}. \quad (4)$$

где $\{kd_A\} \subseteq KD$ означает конструкторский документ на изделие A .

Выражения (3) и (4) показывают зависимость объектов проектирования одного типа от объектов другого типа. Специфика разработки высокотехнологичной продукции с длительным сроком активного существования, рассматриваемая на примере космического приборостроения, определяет неоднородный состав множества информационных моделей изделия, разработанных в различных САПР. Это множество включает подмножества информационных моделей, разработанных в различных САПР: при электрическом и при механическом проектировании. Таким образом, можно записать

$$DM = \{MCADDM, ECADDM\},$$

где $MCADDM$ – множество файлов ИМИ, разработанных в САПР механического моделирования (MCAD), а $ECADDM$, соответственно, – в САПР электрического моделирования (ECAD).

По технологии проектирования РЭА этап электрического проектирования, решающего задачи схемотехнического и радиотехнического характера, предшествует этапу механического моделирования, рассматривающего конструктивные и теплотехнические вопросы. Но не все составные части изделия подлежат электрическому моделированию, некоторые детали создаются только при механическом конструировании, например, рамка для закрепления электронных схем. В свою очередь, все электронные элементы ЭСИ, помимо своих электрических характеристик, моделирование которых осуществляется с помощью ECAD-САПР, имеют конструктивное воплощение (корпус), пространственные параметры

которого являются объектом моделирования в MCAD-САПР. Таким образом, в рамках разработки одного изделия A множество ИМИ, созданных в MCAD-САПР, превышает множество ИМИ, созданных в ECAD-САПР. Поскольку множества $MCADDM_A \subseteq MCADDM$ и $ECADDM_A \subseteq ECADDM$ являются конечными, можно записать выражение, показывающее сравнение их мощностей:

$$|MCADDM_A| > |ECADDM_A|. \quad (5)$$

Учитывая (5), можно сделать вывод, что полный состав изделия определяется его информационными моделями, разработанными в MCAD-САПР. Поэтому множество элементов ЭСИ изделия A , $ITM_A \subseteq ITM$, является сюръективной функцией множества информационных моделей изделия A , разработанных в MCAD-САПР

$$w_A : MCADDM_A \rightarrow ITM_A. \quad (6)$$

Практическая реализация выражений (1)–(6) будет положена в основу программного обеспечения средств автоматизированного управления жизненным циклом объектов проектирования в рамках ИИС, включая разработку требований к программной интеграции САПР и СУД.

Концептуальное проектирование базы данных системы управления данными

В терминах объектно-ориентированного подхода каждому множеству из выражений (1)–(6) поставлен в соответствие класс, объекты которого будут использоваться для хранения сущностей.

Зависимости между объектами проектирования, формализованные в виде выражений (1)–(6), позволяют выделить типы рассматриваемых сущностей и структурировать их в виде классов в соответствии с положениями объектно-ориентированного подхода [3]. Рассматривая выражение (2) с точки зрения системного подхода, можно сделать вывод, что для реализации автоматизированного управления жизненным циклом объектов проектирования в рамках ИИС в модели данных СУД необходимо предусмотреть следующие классы: «Проекты», «Изделия», «Элемент ЭСИ», «Документы». При этом между данными классами существуют следующие отношения:

- в рамках одного проекта может выполняться разработка одного или более изделий;
- для одного изделия может существовать множество вариантов его электронной структуры;
- к одному проекту может относиться множество документов;
- ЭСИ и ее составные части описывается одним или более документом;
- к одному изделию может относиться один или более документ.

В связи с неоднородным составом множества ЭСИ, класс «элемент ЭСИ» является абстрактным и детализируется на следующие классы, согласно выделению составных частей изделия: «Стандартное изделие», «Комплект», «Деталь», «Сборочная

единица», «Прочее изделие», «Изделие из материала». При этом между данными классами определены следующие отношения:

- к одному экземпляру класса «Комплект» может относиться множество экземпляров классов «Деталь», «Сборочная единица», «Стандартное изделие», «Комплект»;
- к одному экземпляру класса «Сборочная единица» может относиться множество экземпляров классов «Деталь», «Сборочная единица», «Прочее изделие», «Изделие из материала», «Стандартное изделие».

Выражение (5), а также особенности приборного проектирования с электрической и механической точек зрения позволяют сделать вывод о необходимости объявления классов, объекты которых будут использоваться для хранения соответствующих информационных моделей. Кроме того, одной из особенностей механического проектирования является компонентный состав модели и понятие сборки. Сборка представляет собой структурированную совокупность отдельных конструктивных элементов, с учетом их габаритных размеров, положений в пространстве и ограничений подвижности. Фактически сборка представляет собой файл, в котором содержатся ссылки на файлы с описанием отдельных элементов, а также вышеперечисленные параметры зависимостей между ними. Таким образом, сборка состоит из множества подборок и деталей.

Аналогично организован компонентный состав модели изделия с точки зрения электрического проектирования: проект модели представляет собой набор ссылок на структурированную совокупность электрических схем, печатных плат и программного кода. Следовательно, применительно к рассматриваемой специфике проектирования высокотехнологичных изделий на примере отрасли космического приборостроения потомками абстрактного класса «ИМИ» являются абстрактные классы «MCAD» и «ECAD». Они, в свою очередь, детализируются на следующие классы:

- «MCAD-Сборка» – для хранения сборки, с которой ассоциируется множество экземпляров классов сборки «MCAD-Сборка» и деталей «MCAD-Деталь»;
- «ECAD-Проект» – для хранения проекта, с которым ассоциируется множество экземпляров класса входящих в него файлов электрических схем, печатных плат, программного кода «ECAD-Файл».

Различные типы документов (ИМИ, КД) позволяют сделать вывод о необходимости объявить класс «Документы» абстрактным суперклассом, потомками которого будут классы «КД» и «ИМИ».

Учтена возможность компонентного состава файлов информационных моделей, например сборка (класс «MCAD-Сборка») или проект (класс «ECAD-Проект»), представляющий собой файл с содержанием ссылок на файлы с описанием отдельных деталей (класс «MCAD-Деталь») или эл-

мационных и алгоритмических моделей программного обеспечения управления жизненным циклом разнотипных объектов проектирования в интегрированной информационной среде. При этом учте-

ны базовые положения технологий информационной поддержки жизненного цикла изделий об интеграции системы управления данными с системами автоматизированного проектирования.

СПИСОК ЛИТЕРАТУРЫ

1. Михайлов В.Г. Анализ и сравнение существующих PDM / Форум САПР2000. URL: <http://fsapr2000.ru/index.php?showtopic=17636> (дата обращения: 11.01.2013).
2. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений. 3-е изд. — М.: Вильямс, 2010. — 720 с.
3. Дрожин В.В., Зинченко Р.Е. Системный подход к концептуальному моделированию предметной области в самоорганизующейся информационной системе // Программные продукты и системы. — 2009. — № 4. — С. 73–79.
4. Вичугова А.А., Вичугов В.Н., Цапко Г.П., Цапко С.Г., Фень А.М. Анализ характера и состава проектной информации для построения модели данных PLM-системы Enovia SmarTeam применительно к задачам космического приборостроения // Системы проектирования, технологической подготовки производства и управления этапами жизненного цикла промышленного продукта: Труды XII Междунар. конф. — М.: ООО «Аналитик», 2012. — С. 57–60.

Поступила 01.03.2013 г.

Алгоритмическое и программное обеспечение

УДК 681.3.06

ИССЛЕДОВАНИЕ ПРОГРАММНЫХ РЕАЛИЗАЦИЙ АЛГОРИТМОВ ВЫЧИСЛЕНИЯ CRC, СОВМЕСТИМЫХ С PKZIP, WINRAR, ETHERNET

Е.А. Мыцко, А.Н. Мальчуков

Томский политехнический университет
E-mail: EvgenRus70@mail.ru; jgs@tpu.ru

Приведены численные результаты компьютерного эксперимента по определению быстродействия программных реализаций алгоритмов вычисления CRC32. Показано, что быстродействующий четырёхбайтовый матричный алгоритм следует применять во встраиваемых системах, промышленных системах передачи данных.

Ключевые слова:

Контрольная сумма, циклический избыточный код, табличный алгоритм, матричный алгоритм, программная реализация.

Key words:

Check sum, cyclic redundancy code, table-driven algorithm, matrix-driven algorithm, software implementation.

Существуют различные алгоритмы вычисления контрольной суммы CRC и способы их реализации. Стандартный алгоритм подразумевает побитное вычисление контрольной суммы путём деления полинома, представляющего данные, на образующий полином, используемый для вычисления CRC в различных протоколах передачи данных (Ethernet, ZigBee) и архиваторах (Pkgzip, WinRAR). Также существует табличный алгоритм [1], ускоряющий расчёт контрольной суммы за счёт побайтного вычисления. Далее речь пойдёт о программных реализациях алгоритмов вычисления CRC32, используемых в Pkgzip, WinRAR, Ethernet.

Классическая реализация алгоритма вычисления CRC

Основой для данного алгоритма является полиномиальная арифметика [2]. Классическая реализация расчёта контрольной суммы заключается в побитовом делении полинома, представляющего данные (файл) на образующий полином. Образующий полином вычисляется в зависимости от сферы применения алгоритма. В данном случае это бу-

дет полином, используемый в Pkgzip, WinRAR, Ethernet.

Классический алгоритм реализуется с помощью последовательного итерационного сдвига данных в регистре с обратной связью на один бит [3]. В результате проделанных операций в регистре вычисляется контрольная сумма CRC.

После обработки всех данных в регистре будет находиться контрольная сумма CRC32 (рис. 1). Основным недостатком данного алгоритма является то, что в один момент времени происходят вычисления для одного бита данных, что значительно замедляет его работу. В связи с этим для повышения быстродействия используются алгоритмы, выполняющие вычисления побайтно.

Табличный алгоритм

Особенностью табличного алгоритма является то, что данные считываются побайтно, что значительно ускоряет вычисление CRC. При расчёте контрольной суммы используется таблица с предварительно вычисленными значениями на основе образующе-

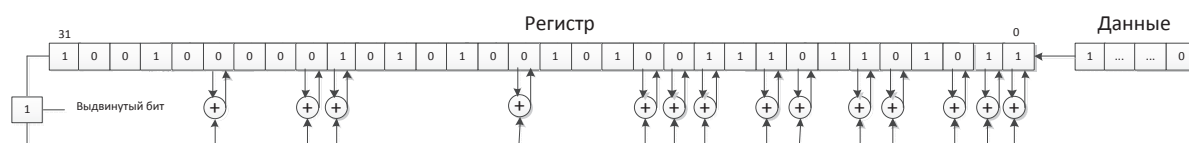


Рис. 1. Схематичное представление реализации классического алгоритма вычисления CRC32

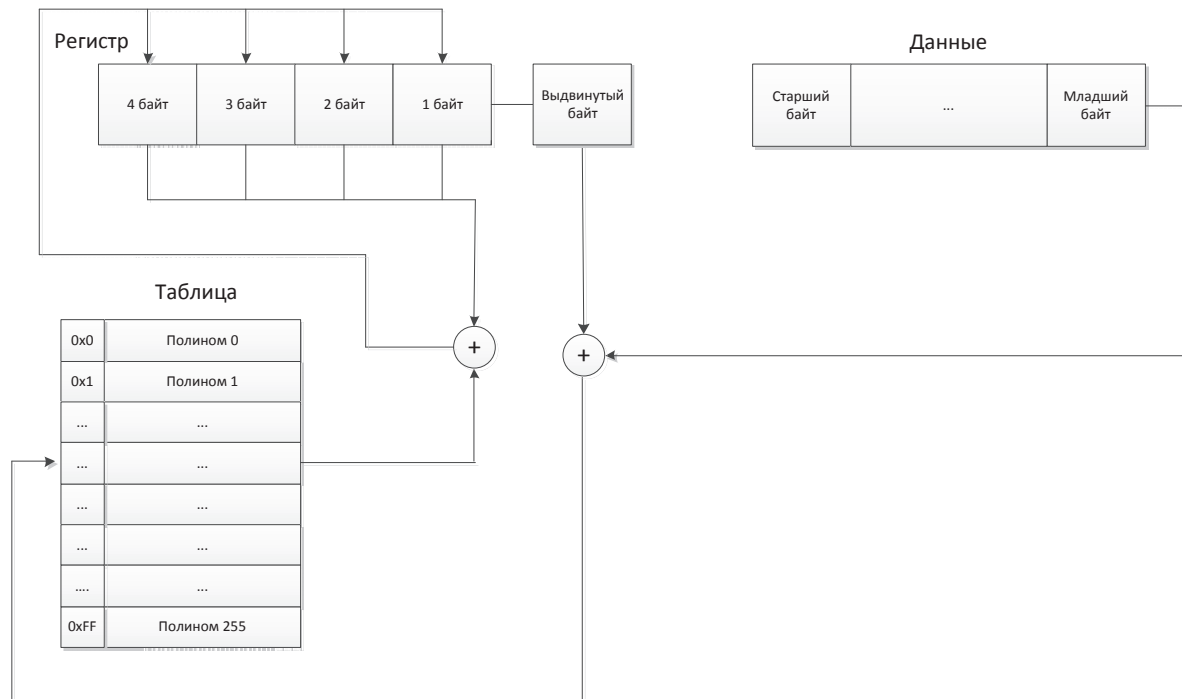


Рис. 2. Схематическое представление табличного алгоритма

го полинома. В алгоритмах Pkzip, WinRAR, Ethernet используется образующий полином 104C11DB7h [4], представленный в шестнадцатеричной системе счисления. На рис. 2 схематично представлен табличный алгоритм.

Матричный алгоритм

Процесс вычисления контрольной суммы CRC32 в матричном алгоритме осуществляется так же, как и в табличном, за исключением того, что вместо таблицы используется операция умножения вектора (выдвинутый байт) на матрицу по модулю 2 (рис. 3, 4).

При программной реализации матричного алгоритма (однобайтовый сдвиг) вычисления контрольной суммы CRC32 потребуются объём памяти равный 32 байтам (для хранения матрицы).

Матричный алгоритм можно ускорить, если вместо однобайтового сдвига использовать сдвиг по 2 или 4 байта. Сдвиг по 3 байта в данном случае не рассматривается, так как при программной реализации отсутствует тип данных размером 3 байта, что вызывает трудности и потерю скорости при реализации алгоритма.

При увеличении числа байт, обрабатываемых за итерацию, будет увеличиваться размер матрицы,

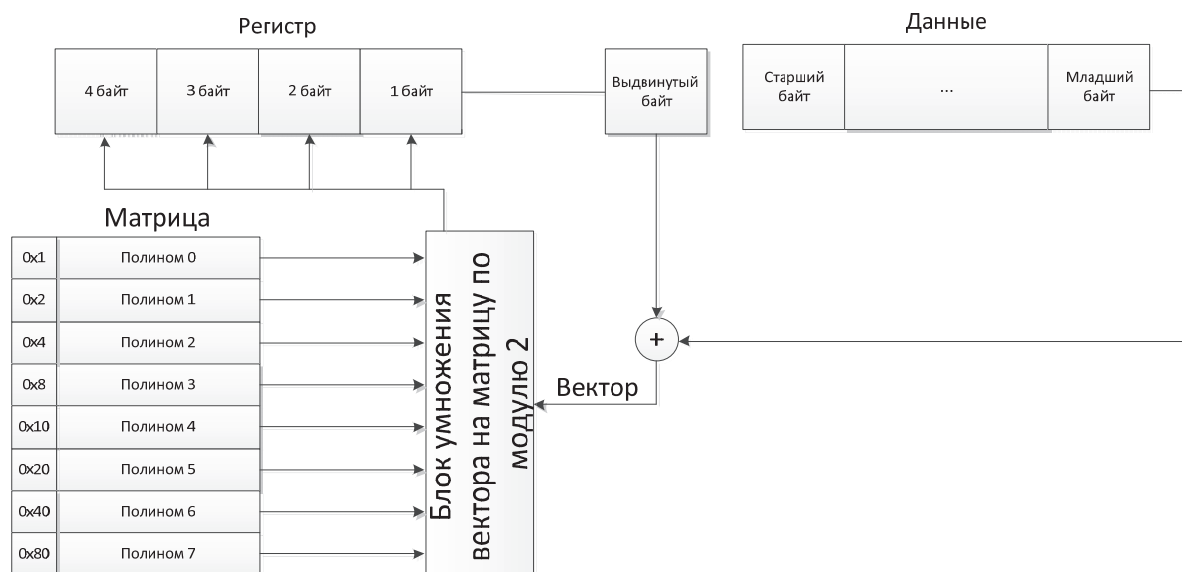


Рис. 3. Схематическое представление матричного алгоритма со сдвигом 1 байт

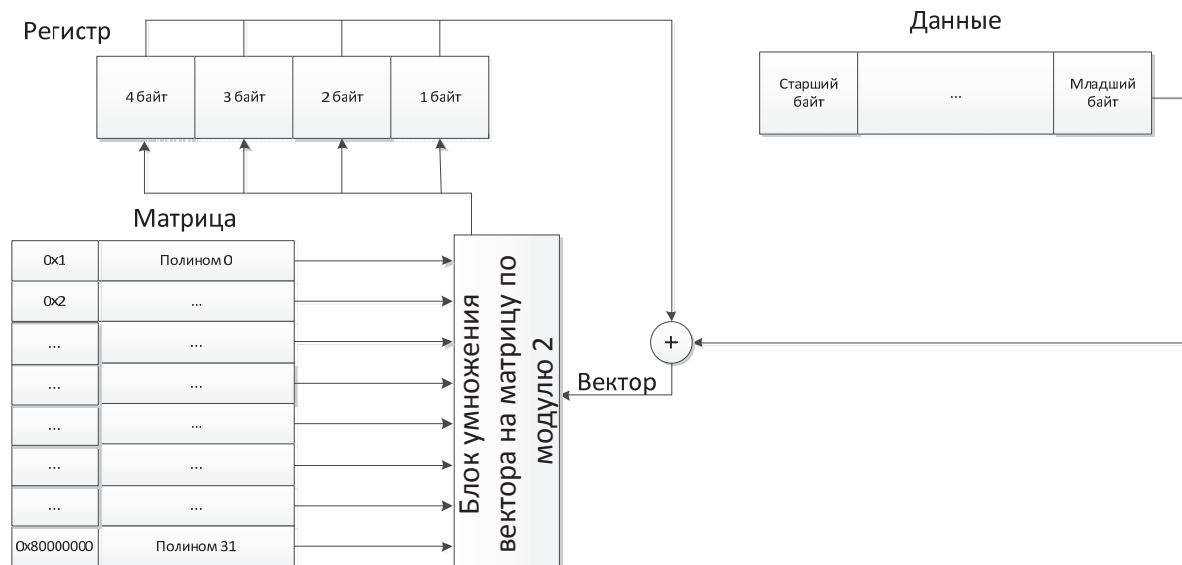


Рис. 4. Схематичное представление матричного алгоритма со сдвигом 4 байта

используемой при вычислениях. Для двухбайтового сдвига матрица расширится до 64-х байт; для четырехбайтового сдвига – до 128 байт [5].

Компьютерный эксперимент

Для постановки компьютерного эксперимента по определению быстродействия программных реализаций различных алгоритмов вычисления контрольной суммы CRC32 использовался суперкомпьютерный кластер «СКИФ-политех1». В табл. 1 представлена конфигурация суперкомпьютерного кластера «СКИФ-политех1» [6]. Для получения временной оценки вычисления контрольной суммы над конкретным файлом конкретной программной реализацией производилось по 50 запусков для файлов объемом от 10 до 1010 мегабайт с шагом 200 мегабайт. На основе данных по 50-ти запускам каждого алгоритма составлены таблицы средних значений по времени расчёта контрольной суммы, по доверительным интервалам и по размерам исполняемых файлов.

Исследование проводилось в 3 этапа: с буфером в 1 Мб, который установлен в исходном коде табличной реализации [7]; с буферами 8 и 1 байт – для моделирования вычисления контрольной суммы для микроконтроллеров.

Таблица 1. Конфигурация «СКИФ-политех1»

Аппаратная конфигурация	Программная конфигурация
Количество вычислительных узлов: 24	Novell SLES 10
Количество процессоров: 48 (Intel XEON 5150)	Microsoft Windows HPC Server 2008
Количество вычислительных ядер: 96 (2,66 Ghz)	Установлена на узлы КВК В том числе в комплекте ОС
Общий объем ОП: 192 Гб	Программное обеспечение трансляции исходных текстов параллельных программ для языков C, C++ и FORTRAN E
Общий объем HDD: 2880 Гб	динная система авторизации пользователей.
Объем СХД: 5 Тб	Поддержка одновременного исполнения команд на всех или выделенных узлах кластера
Системная сеть: Infiniband 4x, 24 порта	
Вспомогательная сеть: Gigabit Etherhet, 48 портов	
Сервисная сеть: ServNet, 25 портов	
Пиковая производительность: 1,02 ТФЛОПС	

Буфер 1 мегабайт. Использование суперкомпьютерного кластера позволило сэкономить время на проведение эксперимента и получить временные оценки с достаточно высокой достоверностью (табл. 2).

Таблица 2. Средние значения времени расчёта CRC32 и доверительные интервалы результата

Алгоритм	Среднее значение времени расчёта CRC32, с (Доверительные интервалы, %)					
	Объём использованных файлов, Мб					
	10	210	410	610	810	1010
Табличный	0,057 (±1,4)	1,194 (±0,217)	2,299 (±0,203)	3,227 (±0,162)	4,334 (±0,211)	6,184 (±0,155)
Матричный (сдвиг 1 байт)	0,125 (±0,732)	2,81 (±0,112)	5,399 (±0,067)	7,602 (±0,166)	10,23 (±0,161)	14,557 (±0,048)
Матричный (сдвиг 2 байта)	0,092 (±3,882)	2,084 (±0,133)	4,02 (±0,09)	5,645 (±0,105)	7,566 (±0,097)	10,835 (±0,072)
Матричный (сдвиг 4 байта)	0,08 (±0,827)	1,729 (±0,172)	3,32 (±0,011)	4,694 (±0,314)	6,272 (±0,188)	8,965 (±0,064)

Таблица 3. Ускорение относительно однобайтового сдвига

Алгоритм	Ускорение относительно матричного алгоритма со сдвигом 1 байт, %						Среднее ускоре- ние, %
	Объём использованных файлов, Мб						
	10	210	410	610	810	1010	
МС 2 байта	26,4	25,836	25,541	25,743	26,041	25,568	25,85
МС 4 байта	36	37,919	38,507	38,253	38,69	38,414	37,96

Для того чтобы выяснить, как влияет на скорость вычисления увеличение числа байт, обрабатываемых за итерацию, в матричном алгоритме была вычислена разница для данных из табл. 2 для матричных алгоритмов с 2- и 4-байтными сдвигами относительно однобайтового (табл. 3).

Как видно из табл. 3, матричный алгоритм с 4-байтным сдвигом вычисляет контрольную сумму на 37,96 % быстрее, чем матричный алгоритм с однобайтовым сдвигом.

Однако даже матричный алгоритм с 4-байтовым сдвигом уступает по скорости табличному алгоритму на 44,11 % (табл. 4).

В связи с увеличением числа байт до 4, обрабатываемых за итерацию, в матричном алгоритме разрыв по быстродействию относительно табличного алгоритма сокращается до ~44 %. Следует помнить, что четырёхбайтовый матричный алгоритм требует в 8 раз меньше памяти (128 байт) для хранения матрицы, чем для реализации табличного алгоритма (1024 байт).

Моделирование вычисления CRC32 для микроконтроллеров

Для моделирования скорости вычисления алгоритмов для микроконтроллеров были исследованы различные варианты программных реализаций с установленными буферами 8 и 1 байт.

Буфер 8 байт. В данном случае размер буфера был установлен равным 8 байт, что соответствует доступному объёму для микроконтроллеров. Все

этапы исследования проводились аналогично исследованиям с буфером 1 мегабайт. Временные затраты на вычисления контрольной суммы файлов разного объёма и доверительные интервалы полученных результатов приведены в табл. 5. Аналогично предыдущему этапу в табл. 6 и 7 приведены результаты ускорений от однобайтового сдвига и от табличного алгоритма.

Как видно из табл. 6 и 7, уменьшение размера буфера до 8 байт изменило картину, полученную ранее.

Таблица 6. Ускорение относительно однобайтового сдвига

Алгоритм	Ускорение относительно матричного алгоритма со сдвигом 1 байт, %						Среднее ускорение, %
	Объём использованных файлов, Мб						
	10	210	410	610	810	1010	
МС 2 байта	31,4	30,91	31,2	30,63	30,62	30,82	30,93
МС 4 байта	47,19	45,9	47	39,82	45,53	45,44	45,14

Таблица 7. Ускорение матричного алгоритма относительно табличного

Алгоритм	Ускорение относительно табличного алгоритма, %						Среднее ускорение, %
	Объём использованных файлов, Мб						
	10	210	410	610	810	1010	
МС 1 байт	-67,92	-58,543	-61,92	-59,82	-59,83	-63,45	-61,91
МС 2 байта	-15,09	-9,533	-11,28	-10,76	-10,88	-13,06	-11,76
МС 4 байта	11,32	14,242	12,29	12,58	12,93	10,78	12,35

Для двухбайтового матричного алгоритма ускорение относительно однобайтового увеличилось в среднем до 30,93 %, в то время как для четырёхбайтового алгоритма – до 45,14 %. Также из табл. 7 видно, что для однобайтового и двухбайтового алгоритма уменьшилось отставание по скорости

Таблица 4. Ускорение матричного алгоритма относительно табличного

Алгоритм	Ускорение относительно табличного алгоритма, %						Среднее ускоре- ние, %
	Объём использованных файлов, Мб						
	10	210	410	610	810	1010	
С 1 байт	-119,298	-135,343	-134,841	-135,575	-136,575	-135,398	-132,83
МС 2 байта	-61,403	-74,539	-74,858	-74,93	-74,573	-75,21	-72,58
МС 4 байта	-40,35	-44,807	-44,41	-45,46	-44,716	-44,97	-44,11

Таблица 5. Средние значения времени расчёта CRC32 и доверительные интервалы результата

Алгоритм	Средние значения времени расчёта CRC32, с (Доверительные интервалы, %)					
	Объём использованных файлов, Мб					
	10	210	410	610	810	1010
Табличный	0,106 (±1,789)	2,183 (±1,966)	4,139 (±0,980)	6,166 (±1,537)	8,239 (±1,0067)	10,003 (±1,082)
Матричный (сдвиг 1 байт)	0,178 (±1,088)	3,461 (±0,521)	6,702 (±0,328)	9,855 (±0,474)	13,169 (±0,404)	16,350 (±0,371)
Матричный (сдвиг 2 байта)	0,122 (±1,447)	2,39112 (±0,516)	4,606 (±0,408)	6,83664 (±0,463)	9,13604 (±0,366)	11,31524 (±0,319)
Матричный (сдвиг 4 байта)	0,094 (±2,580)	1,87208 (±0,386)	3,6392 (±0,464)	5,39032 (±0,267)	7,1732 (±0,308)	8,92452 (±0,337)

Таблица 8. Средние значения времени расчёта CRC32 и доверительные интервалы результата

Алгоритм	Средние значения времени расчёта CRC32, с (Доверительные интервалы, %)					
	Объём использованных файлов, Мб					
	10	210	410	610	810	1010
Табличный	0,315 ($\pm 2,526$)	6,044 ($\pm 0,598$)	11,797 ($\pm 1,36$)	17,452 ($\pm 1,082$)	23,152 ($\pm 0,985$)	28,767 ($\pm 0,896$)
Матричный (сдвиг 1 байт)	0,361 ($\pm 0,637$)	7,152 ($\pm 0,630$)	13,958 ($\pm 0,621$)	20,610 ($\pm 0,6483$)	27,558 ($\pm 0,5331$)	34,129 ($\pm 0,523$)
Матричный (сдвиг 2 байта)	0,216 ($\pm 0,786$)	4,269 ($\pm 0,611$)	8,263 ($\pm 0,614$)	12,243 ($\pm 0,374$)	16,389 ($\pm 1,091$)	20,221 ($\pm 0,432$)
Матричный (сдвиг 4 байта)	0,140 ($\pm 1,092$)	2,744 ($\pm 0,664$)	5,292 ($\pm 0,464$)	7,858 ($\pm 0,423$)	10,546 ($\pm 0,864$)	13,090 ($\pm 0,552$)

вычисления до 61,91 и 11,76 % соответственно; в то время как матричный четырехбайтовый алгоритм уже имеет опережение по скорости в среднем на 12,35 % относительно табличного.

Буфер 1 байт. Для буфера размером 1 байт результаты исследования приведены в табл. 8–10.

Таблица 9. Ускорение относительно однобайтового сдвига

Алгоритм	Ускорение относительно матричного алгоритма со сдвигом 1 байт, %						Среднее ускорение, %
	Объём использованных файлов, Мб						
	10	210	410	610	810	1010	
МС 2 байта	40,166	40,31	40,8	40,596	40,529	40,751	40,52
МС 4 байта	61,218	61,633	62,086	61,872	61,731	61,645	61,69

При исследовании реализации с буфером 1 байт, что по сути можно приравнять к вычислениям без буфера при побайтовой передаче данных, было установлено, что ускорение матричных алгоритмов относительно однобайтового матричного алгоритма увеличилось до 40,52 и 61,69 % соответственно. Исследование же ускорений матричных алгоритмов относительно табличного показало, что уже двухбайтовый матричный алгоритм опережает табличный на 29,91 %, в то время как четырехбайтовый – на 54,86 %, что является значительным приростом быстродействия относительно табличного алгоритма.

Полученные результаты характеризует тот факт, что при уменьшении буфера до 1 байта потери скорости вычислений для табличного алгоритма на-

ного значительней, чем для матричных алгоритмов, в то время как быстродействие обоих алгоритмов с сокращением размера буфера уменьшается.

Заключение

В работе приведены результаты исследования быстродействия различных реализаций вычисления контрольной суммы CRC32. Поставленный компьютерный эксперимент по определению быстродействия программных реализаций алгоритмов вычисления CRC32 показал, что матричный алгоритм хорошо подходит для применения в микроконтроллерах, где доступный объем памяти не превышает 8 байт, в то время как табличный алгоритм следует применять в системах с большим доступным объемом памяти, так как в реализациях для микроконтроллеров быстродействие данного алгоритма существенно падает.

В результате было установлено, что самый быстродействующий из матричных алгоритмов – четырехбайтовый, следует применять во встраиваемых системах, промышленных системах передачи данных, где используются микроконтроллеры. Рекомендация использования матричного алгоритма обусловлена простотой реализацией и значительной экономией памяти, требуемой для управляющей программы микроконтроллера, а также хорошими показателями скорости по сравнению с табличной реализацией для применения в данной сфере.

Исследование выполнено при поддержке Министерства образования и науки Российской Федерации, соглашение 14.B37.21.0457.

Таблица 10. Ускорение матричного алгоритма относительно табличного

Алгоритм	Ускорение относительно табличного алгоритма, %						Среднее ускоре ние, %
	Объём использованных файлов, Мб						
	10	210	410	610	810	1010	
МС 1 байт	-14,603	-18,332	-18,318	-18,095	-19,030	-18,639	-17,83
МС 2 байта	31,428	29,367	29,956	29,847	29,211	29,707	29,91
МС 4 байта	55,555	54,599	55,141	54,973	54,448	54,496	54,86

СПИСОК ЛИТЕРАТУРЫ

1. Ross N.W. A Painless Guide to CRC Error Detection Algorithms // Dr. Ross Williams. 1993. URL: http://www.ross.net/crc/download/crc_v3.txt (дата обращения: 23.10.2012).
2. Буркатовская Ю.Б., Мальчуков А.Н., Осокин А.Н. Быстродействующие алгоритмы деления полиномов в арифметике по модулю два // Известия Томского политехнического университета. — 2006. — Т. 309. — № 1. — С. 19–24.
3. Темников Ф.Е., Афонин В.А., Дмитриев В.И. Теоретические основы информационной техники. 2-е изд., испр. и доп. — М.: Энергия, 1979. — 512 с.
4. Koopman P. 32-Bit Cyclic Redundancy Codes for Internet Applications // Intern. Conf. on Dependable Systems and Networks (DSN). — Washington, July 2002. — P. 459–468.
5. Мальчуков А.Н., Осокин А.Н. Быстрое вычисление контрольной суммы CRC: таблица против матрицы // Прикладная информатика. — 2010. — № 2 (26). — С. 58–63.
6. Центр коллективного пользования «Суперкомпьютерный кластер «СКИФ-политех» // Сайт ЦКП СКК «СКИФ-политех» ТПУ. 2008. URL: <http://cluster.tpu.ru/> (дата обращения: 23.10.2012).
7. 32 bit Cyclic Redundancy Check Source Code for C++ // Create Window Website. 2011. URL: <http://www.createwindow.com/programming/crc32/> (дата обращения: 01.05.2011).

Поступила 25.12.2012 г.

УДК 004.62

ЭФФЕКТИВНОСТЬ РЕЖИМОВ ВНУТРЕННЕГО ПРЕДСКАЗАНИЯ БЛОКОВ В СОВРЕМЕННЫХ СТАНДАРТАХ СЖАТИЯ ВИДЕО

М.П. Шарабайко, Н.Г. Марков

Томский политехнический университет
E-mail: sme_box@tpu.ru

В 2013 г. ожидается финальная версия стандарта видеокompрессии H.265/HEVC. Предполагается, что с его внедрением начнется массовый переход на системы сверхвысокого разрешения. В статье приведены результаты анализа эффективности методов внутреннего предсказания блоков стандарта H.265/HEVC с точки зрения их влияния на степень сжатия опорных кадров цифрового видео. Кроме того, приводятся результаты сравнения эффективности сжатия опорных кадров в стандартах H.265/HEVC и H.264/AVC.

Ключевые слова:

Сжатие видео, стандарт H.265/HEVC, стандарт H.264/AVC, режимы внутреннего предсказания блоков, опорные кадры.

Key words:

Video compression, standard H.265/HEVC, standard H.264/AVC, intra prediction modes, key frames.

Актуальность задачи цифрового сжатия видео-последовательностей растет вместе с развитием сетей передачи данных, в том числе беспроводных, а также возможностей и мощностей персональных и мобильных компьютерных устройств. При этом требования к степени сжатия видео непрерывно возрастают, стимулируя, в свою очередь, развитие стандартов видеокodирования.

Первый общепринятый стандарт видеокompрессии MPEG2 [1] был окончательно принят в 1996 г., после чего началось быстрое развитие цифрового спутникового телевидения. Следующим стандартом стал MPEG4 part 10 (H.264/AVC) [2], принятый в 2003 г., что стало толчком к развитию систем DVB-T/C, Интернет ТВ и привело к появлению разнообразных сервисов видеобмена и видеосвязи.

В 2013 г. выходит стандарт цифрового сжатия видео-последовательностей H.265/HEVC [3], являющийся стандартом нового поколения и обеспечивающий повышение степени сжатия видео до двух раз по сравнению со степенью сжатия видео с помощью текущего индустриального стандарта H.264/AVC.

В статье исследуется эффективность методов, положенных в основу стандартов H.265/HEVC и H.264/AVC, касающихся предсказания и сжатия опорных видеокадров.

Задача сжатия видео-последовательностей

В цифровом представлении видео-последовательность является потоком видеоданных, состоящим из последовательности видеокадров. Каждый кадр видео-последовательности представляет определенный дискретный момент видеосигнала. Физически же кадр является матрицей пикселей изображения.

Большинство современных стандартов сжатия видео-последовательностей реализуют блочную модель кодирования, суть которой заключается в разбиении кадра видео-последовательности на множество непересекающихся блоков и выполнении алгоритма сжатия на каждом полученном блоке. В стандарте H.265/HEVC структура разбиения на блоки представляет собой квадродерево, подобное применяемому при фрактальном сжатии [4].

Сжатие видеоинформации основано на двух важных принципах. Во-первых, каждый кадр видеопоследовательности обладает пространственной избыточностью вследствие корреляции значений пикселей внутри кадра. Во-вторых, большинство кадров видеопоследовательности похожи на предыдущие, а значит обладают временной избыточностью [5]. Алгоритмы сжатия видеопоследовательностей основаны на устранении этих двух видов избыточности.

В сжатой видеопоследовательности обычно присутствуют как кадры с устранением только пространственной избыточности, так и кадры с устранением обоих видов избыточности. Кадры первого типа называются ключевыми (опорными) и обозначаются символом *I* (сокр. от англ. *intra* – внутренний). Такие сжатые кадры позволяют начать их декодирование сразу, поскольку они не ссылаются на другие кадры данной видеопоследовательности.

Кадр, ссылающийся на один или несколько предыдущих, называется разностным и обозначается символом *P* (сокр. от англ. *predicted* – предсказанный). Кадр, для сжатия которого используются как предыдущие, так и последующие кадры видеопоследовательности называется двунаправленным и обозначается символом *B* (сокр. от англ. *bidirectional* – двунаправленный).

В статье исследования сосредоточены на сжатии опорных видео кадров. При их сжатии в стандарте *H.265/HEVC* используются только внутренние предсказания на основе уже сжатой части этого же кадра. В этом стандарте определено 35 режимов внутреннего предсказания: *DC*, плоский и 33 угловых предсказания (рис. 1).

В стандарте предыдущего поколения *H.264/AVC* определено всего 10 режимов внутреннего предсказания блоков: *DC*, плоский и 8 угловых (направления 6, 10, 14, 18, 22, 26, 30, 34 на рис. 1) [6].

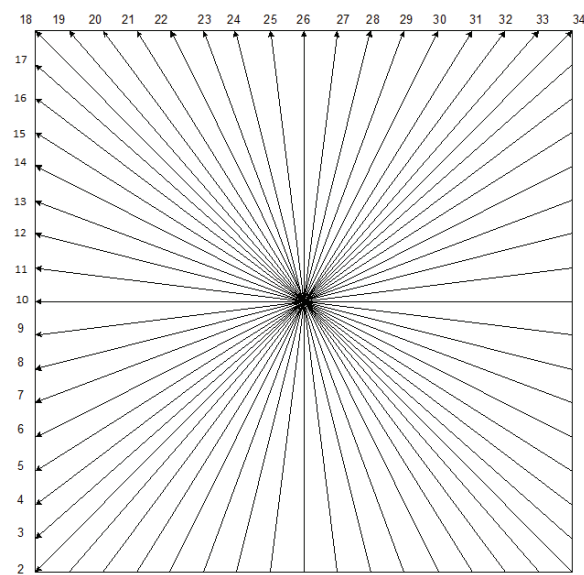


Рис. 1. Угловые внутренние предсказания в стандарте *H.265/HEVC*

При режиме *DC*-предсказания вычисляется среднее значение граничных пикселей, расположенных сверху и слева от предсказываемого блока (рис. 2), которым заполняется весь шаблон предсказания.



Рис. 2. *DC* предсказание в стандартах *H.265/HEVC* и *H.264/AVC*

При режиме плоского предсказания используется линейная функция от левых и верхних граничных пикселей предсказываемого блока (рис. 3).

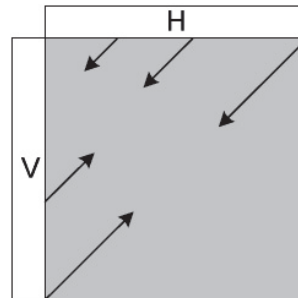


Рис. 3. Плоское предсказание в стандартах *H.265/HEVC* и *H.264/AVC*

В режимах угловых предсказаний шаблон предсказания формируется экстраполяцией значений граничных пикселей, попадающих под направление предсказания. Например, на рис. 4 схематично представлено вертикальное внутреннее предсказание, при котором используются пиксели *A–D*.

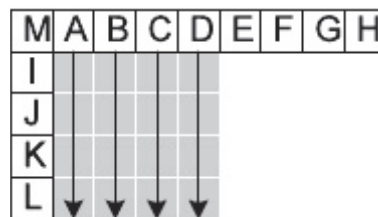


Рис. 4. Вертикальное предсказание в стандартах *H.265/HEVC* и *H.264/AVC*

Помимо введения дополнительных режимов внутреннего предсказания в стандарте *H.265/HEVC*, по сравнению со стандартом *H.264/AVC*, модифицировано энтропийное кодирование. Так, в стандарте *H.264/AVC* используется контекстно-адаптивное двоичное арифметическое кодирование *CABAC* (сокр. от англ. *Context Adaptive Binary Arithmetic Coding*), включающее 15 контекстных моделей для кодирования синтаксических элементов.

В стандарте *H.265/HEVC* применяется синтаксически адаптивное двоичное арифметическое кодирование *SBAC* (сокр. от англ. *Syntax-Based Context-Adaptive Binary Arithmetic Coding*), использующее 31 контекстную модель.

Исследование эффективности методов внутреннего предсказания блоков

Ставилась задача исследования эффективности режимов внутреннего предсказания блоков в стандартах *H.264/AVC* и *H.265/HEVC* в целом на видеопоследовательности и на участках видеокадра, а также изучения выгоды от использования 24 новых режимов внутреннего предсказания блоков в стандарте *H.265/HEVC*.

Исследования проводятся на известном наборе видеопоследовательностей университета Ганновера [7], представленном в таблице.

Оценка эффективности сжатия видеопоследовательности осуществляется на основе общепринятого в области сжатия видеоинформации отношения объективно оцениваемого искажения сжатой видеопоследовательности к скорости потока данных (размеру) сжатой видеопоследовательности. Для возможности соотнесения полученных в данной работе результатов с результатами других исследователей в качестве оценки искажения используется метрика пикового соотношения сигнала к шуму *PSNR* (сокр. от англ. *Peak Signal to Noise Ratio*), вычисляемого по формуле:

$$PSNR = 10 \cdot \lg \left(\frac{255^2}{MSE} \right), \quad (1)$$

при этом

$$MSE = \frac{1}{m \cdot n} \cdot \sum_{i=0}^m \sum_{j=0}^n |S(i, j) - D(i, j)|^2,$$

где $S(i, j)$ и $D(i, j)$ — i и j пиксели исходного и сжатого видео кадров размером $m \times n$ соответственно. Для оценки искажений при сжатии видеопоследовательностей применяется метрика *OPSNR* (сокр. от англ. *Overall Peak Signal to Noise Ratio*), в которой значения *MSE* (сокр. от англ. *Mean Squared Error* — среднеквадратическая ошибка) для каждого сжатого кадра видеопоследовательности суммируются и подставляются в формулу (1).

Выбор режима внутреннего предсказания и процесс сжатия видеопоследовательности опорными кадрами осуществляется с помощью авторской реализации видеоэнкодера, совместимого с проектом стандарта *H.265/HEVC* версии № 9 [1]. В данной реализации выбор режима внутреннего предсказания выполняется по минимальному значению суммы абсолютных значений матрицы коэффициентов, получаемых преобразованием Адамара–Уолша матрицы остаточных коэффициентов для каждого режима предсказания. Преобразование Адамара–Уолша, именуемое как *SATD* (сокр. от англ. *Sum of Absolute Transformed Differences* — сумма абсолютных значений преобразованных разностей), является наиболее часто используемым

методом принятия решения при сжатии видео [4], так как является достаточно хорошим приближением к дискретно-косинусному преобразованию (ДКП), выполняемому при непосредственном сжатии остаточных коэффициентов блоков видео кадра. Преобразование Адамара–Уолша применяется при переборе режимов предсказания для ускорения процесса перебора, т. к. его вычисление на ПЭВМ происходит намного быстрее вычисления ДКП.

Таблица. Тестовый набор видеопоследовательностей

Название	<i>BasketballDrill</i>	<i>PeopleOnStreet</i>
Разрешение	832×480	2560×1600
Частота (кадр/с)	50	30
Всего кадров	501	150
Битрейт (Мбит/с)	228	1406,25

На рис. 5 приведен первый кадр тестовой видеопоследовательности *BasketballDrill*, разбитый сеткой на блоки размером 16×16 пикселей. Это максимальный размер макроблока в стандарте *H.264/AVC*, и один из допустимых размеров блока кодирования (*CU — coding unit*) в стандарте *H.265/HEVC*. *CU* может иметь размер 64×64, 32×32, 16×16 и 8×8 [1]. На таком разбиении интересно сравнить режимы внутреннего предсказания, т. к. чем ближе к исходному будет предсказан блок большего размера, тем меньше необходимость дальнейшего разбиения, и меньше информации требуется закодировать.

При любом виде кодирования чем больше предсказание подобно текущему кодируемому блоку, тем меньше разность (остаток) этих двух блоков (кодируемого и предсказанного). Чем меньше разность, тем меньше высоких частот, а значит, в результате дискретно-косинусного преобразования получится меньше коэффициентов, которые необходимо закодировать и передать в декодер. Таким образом, чем лучше предсказание, тем выше эффективность сжатия.

На рис. 5 приведена исследуемая область видеокadra. Область подобрана специально из-за наличия угловых шаблонов (контуры номера игрока, футболки, руки, линии разметки идут под разными углами). На таком участке интересно оценить эффективность и пользу от новых режимов внутреннего предсказания.

На рис. 6–10 показана эффективность сжатия блоков при различных режимах внутреннего предсказания. Эффективность предсказания определяется получаемым размером сжатого блока, так как это непосредственная характеристика степени сжатия, не отражающая, тем не менее, напрямую качество предсказания и потери в декодированном блоке.

Рис. 6 показывает эффективность сжатия при использовании только *DC* режима предсказания. Каждый предсказанный блок однороден, так как полностью заполняется вычисленным средним

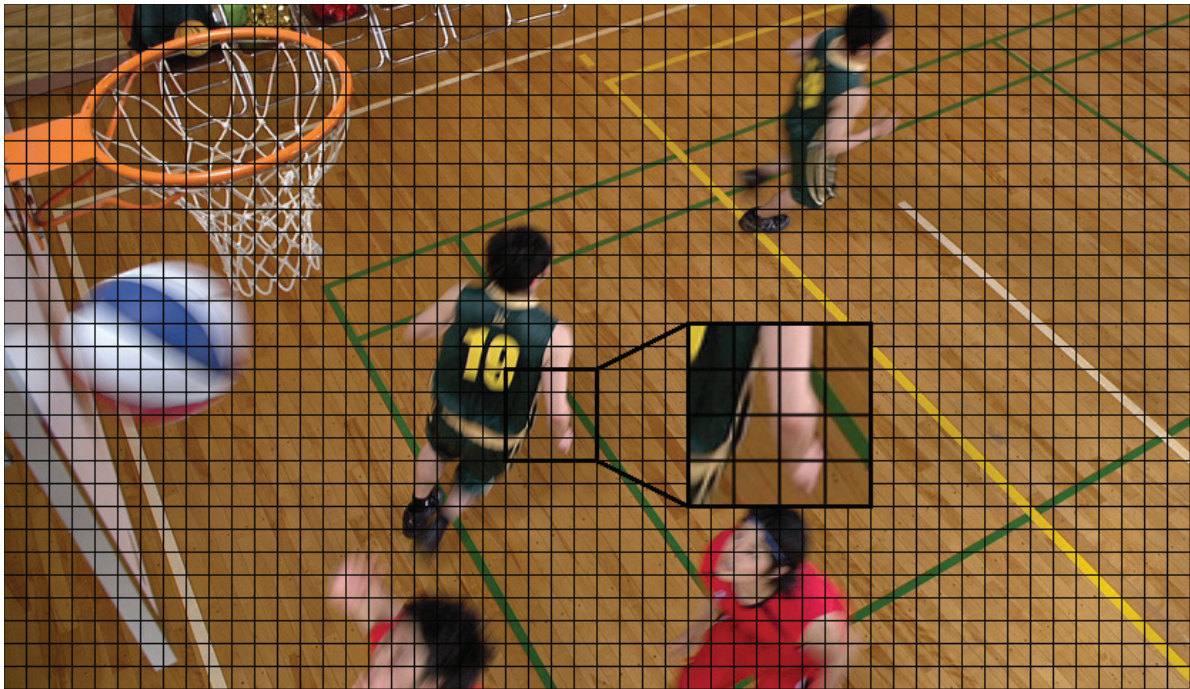


Рис. 5. Разбиение кадра видео последовательности BasketballDrill на блоки размером 16×16 с выделением исследуемого участка

значением соседних пикселей. Поскольку сам участок кадра неоднороден, такое предсказание является крайне неточным и достаточно близко предскажет лишь однородные области: блоки (3;0), (0;1), (3;3).

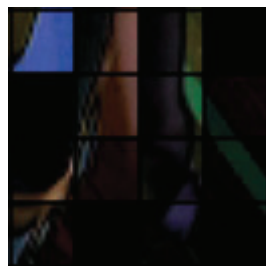
Рис. 7 показывает эффективность сжатия при использовании только плоского режима предсказания. Каждый блок экстраполируется линейной функцией от левой и верхней границ блока. В результате предсказание получается не такое однородное, как при DC режиме, и некоторые блоки предсказываются лучше, например, блоки (0;0) и (0;1) имеют заметно меньший размер сжатия.

При использовании всего доступного набора режимов внутреннего предсказания стандарта H.265/HEVC (рис. 8) удастся повысить эффективность сжатия блоков (1;0), (3;1), (3;2). Для последних двух блоков выбран режим внутреннего предсказания 20, соответствующий примерному углу интерполяции 56° . Линия разметки баскетбольной

площадки, попадающая в этот участок области, очень хорошо предсказывается с помощью такого режима углового предсказания (рис. 6, б, в), угол направления которого максимально близок к углу разметки на данном участке. Размеры сжатых коэффициентов для этого участка на 48 и на 25 % меньше, чем при предсказании от среднего и плоском предсказании. Режим внутреннего предсказания 20 нет в стандарте H.264/AVC. При использовании только режимов внутреннего предсказания, доступных в этом стандарте, наиболее эффективными угловыми режимами сжатия блоков (3;1), (3;2) становятся 22 и 18 (рис. 9). При этом их эффективность заметно уступает режимам DC и плоскому режиму (рис. 1, 2). Видно, что предсказание разметки баскетбольной площадки режимом внутреннего предсказания 18 с углом в 45° (блок (3;2) на рис. 9) дает большую разность, более высокую ошибку предсказания (рис. 9, в) и, как следствие, больший размер сжатия (рис. 9, г).



а



б

1231	947	690	486
378	1056	843	785
1210	1013	712	896
1218	353	730	702

в

Рис. 6. Иллюстрация эффективности сжатия блоков при DC режиме внутреннего предсказания: а) блоки, предсказанные режимом DC; б) блоки после вычитания предсказываемого и предсказанного блоков (остатки); в) соответствующие размеры сжатых блоков, бит

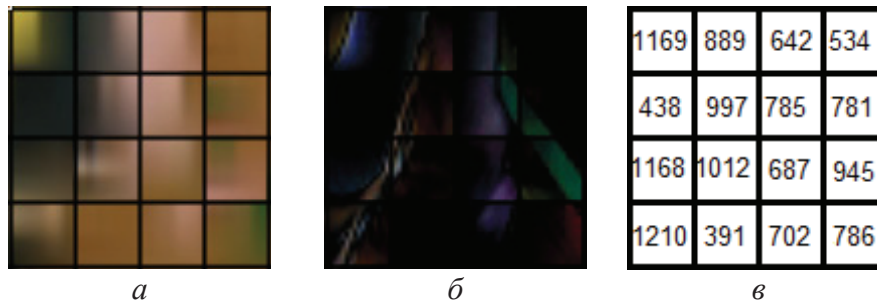


Рис. 7. Иллюстрация эффективности сжатия блоков при плоском режиме внутреннего предсказания: а) блоки, предсказанные плоским режимом; б) блоки после вычитания предсказываемого и предсказанного блоков (остатки); в) соответствующие размеры сжатых блоков, бит

Наконец, если оставить только режимы, доступные в стандарте H.264/AVC для блоков размером 16×16 , то можно заметить снижение эффективности сжатия блоков, например, блоки (1;2) и (1;3) теперь имеют размер 845 и 783 бит соответственно (рис. 10), а эффективность снизилась на 9 и на 48 % соответственно по сравнению с тем, когда были доступны все режимы предсказания.

В стандарте H.264/AVC существуют 3 размера блоков предсказания: 16×16 , 8×8 и 4×4 . Для блоков размером 8×8 и 4×4 допустимо использовать режимы предсказания 1, 6, 10, 14, 18, 22, 26, 30, 34. Таким образом, плоский режим предсказания для блоков такого размера не может использоваться. Для блоков размером 16×16 доступно всего 4 режима внутреннего предсказания: 0 (плоский), 1 (от среднего), 10 (горизонтальный) и 26 (вертикальный).

Добавленные режимы внутреннего предсказания эффективны не на всех блоках предсказания, а скорее на участках с отчетливым шаблоном, идущим под соответствующим углом, как, например, линии разметки площадки в рассмотренном примере. Однако эффективность и частота выбора новых режимов внутреннего предсказания зависит от характеристик самого видео: чем больше угловых шаблонов внутри кадра, тем более эффективны новые режимы предсказания, а значит, тем более эффективно сжатие внутренних блоков в стандарте H.265/HEVC по сравнению со стандартом H.264/AVC, в котором этих предсказаний нет.

Следующей задачей является оценка эффективности дополнительных режимов предсказания при сжатии видео последовательностей. Для этого используется тестовая модель HM 9.0 – общедоступная программная реализация стандарта

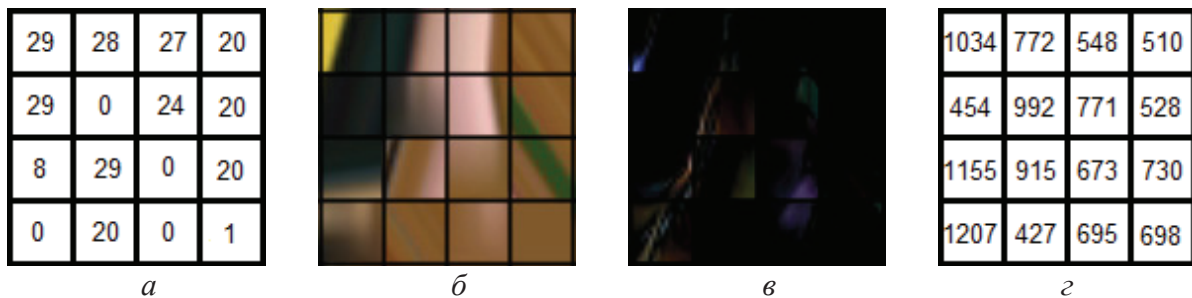


Рис. 8. Иллюстрация эффективности сжатия блоков при выборе из всех режимов внутреннего предсказания: а) режим внутреннего предсказания для данного блока; б) предсказанные блоки; в) блоки после вычитания предсказываемого и предсказанного блоков (остатки); г) соответствующие размеры сжатых блоков, бит

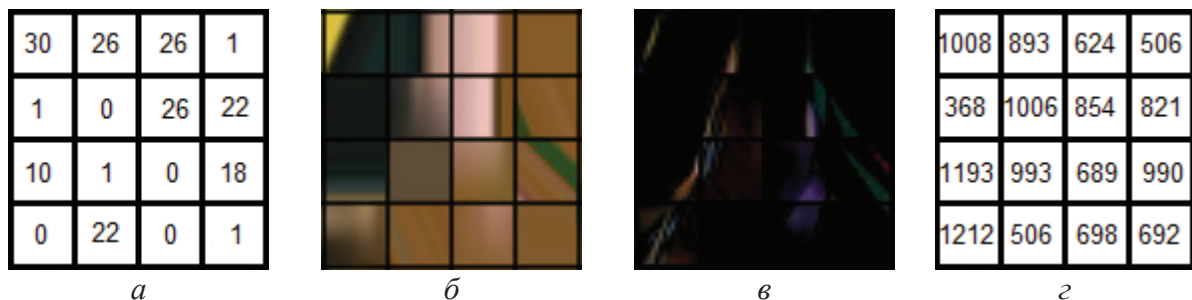


Рис. 9. Иллюстрация эффективности сжатия блоков при выборе из всех режимов внутреннего предсказания, которые были в H.265/AVC: а) режим внутреннего предсказания для данного блока; б) предсказанные блоки; в) блоки после вычитания предсказываемого и предсказанного блоков (остатки); г) соответствующие размеры сжатых блоков, бит

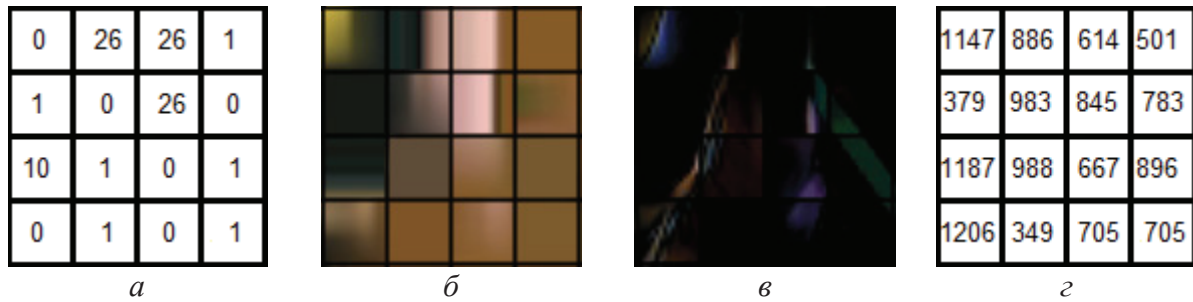


Рис. 10. Иллюстрация эффективности сжатия блоков при выборе из всех режимов внутреннего предсказания, доступных в H.265/AVC для блоков размером 16×16: а) режим внутреннего предсказания для данного блока; б) предсказанные блоки; в) блоки после вычитания предсказываемого и предсказанного блоков (остатки); г) соответствующие размеры сжатых блоков, бит

H.265/HEVC, созданная разработчиками стандарта JCT-VC. Эта реализация выполняет выбор режима предсказания на основе полного перебора всех вариантов предсказания с вычислением *SATD* и модели *RDO* (от англ. *rate-distortion optimization* – оптимизация соотношения степени сжатия к шуму). При использовании полной модели *RDO* выполняется кодирование нескольких наиболее удачных вариантов предсказаний, вычисляются размеры сжатия и потери в декодированных блоках. Полученные метрики используются при окончательном выборе режима предсказания, что позволяет определить наиболее эффективный вариант.

Используется три конфигурации сжатия по стандарту H.265/HEVC: первая с максимальным размером блока кодирования 64×64 пикселей, глубина разбиения 4 (обозн. *HM 64-4* на рис. 11, 12); вторая с максимальным размером блока кодирования 32×32 пикселей, глубина разбиения 3; третья с максимальным размером блока кодирования 16×16 пикселей, глубина разбиения 2, подобно разбиению кадра на блоки в стандарте H.264/AVC (обозн. *HM 16-2* на рис. 1, 12). В первом случае

блок предсказания может быть выбран размером 64×64, 32×32, 16×16, 8×8 или 4×4; во втором – размером 32×32, 16×16, 8×8 или 4×4, в третьем – 16×16, 8×8 или 4×4. Дополнительно используется конфигурация с отключенными внутренними предсказаниями, отсутствующими в стандарте H.264/AVC (*HM 16-2-AVC*). Для сравнения используется тестовая модель *JM 18.3* – общедоступная программная реализация стандарта H.264/AVC комитетом JCT-VC. Исследования проводятся на наборе видеопоследовательностей, приведенном в таблице.

Графики отношения скорости видеопотока (битрейта, кбит/с) к качеству сжатия (*PSNR*, дБ) приведены на рис. 11, 12. На используемых тестовых видеопоследовательностях применение блоков размером 64×64 пикселей с глубиной разбиения 4 дает практически идентичный результат сжатия, что и при использовании блоков размером 32×32 пикселей с глубиной 3 и 16×16 пикселей с глубиной 2. Графики *HM 64-4*, *HM 16-2* на рис. 11, 12 совпадают. Для удобства восприятия график *HM 32-3* не отображен на рис. 11, 12, т. к. он совпадает с графиками *HM 64-4* и *HM 16-2*.

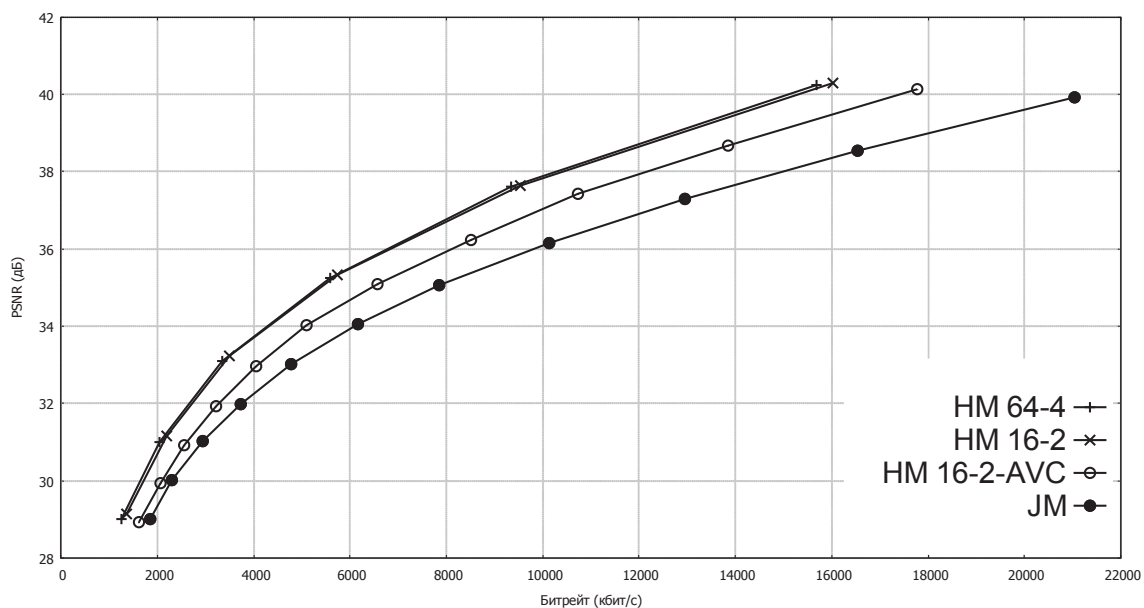


Рис. 11. Сравнение эффективности сжатия, последовательность *BasketballDrill*

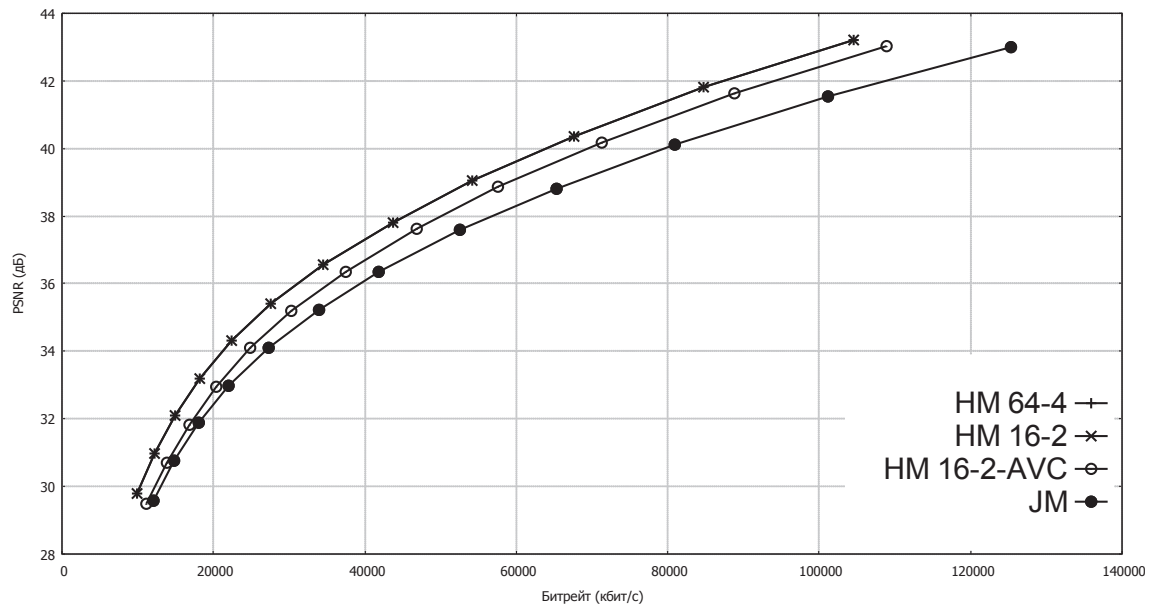


Рис. 12. Сравнение эффективности сжатия, последовательность *PeopleOnStreet*

Последовательности *BasketballDrill* (рис. 11) и *PeopleOnStreet* (рис. 12) имеют разное разрешение, при этом исследуемые конфигурации сжатия дают практически одинаковые результаты относительно разных конфигураций.

Скорости потока несжатых данных у исследуемых последовательностей отличаются в 6 раз. Степень сжатия при одинаковой степени потерь в декодированном видео отличается в 4,5 раза. При качестве 36 дБ эффективность сжатия последовательности *BasketballDrill* стандартом *H.265/HEVC* на 42 % выше, чем при использовании стандарта *H.264/AVC*. Аналогичная разница для последовательности *PeopleOnStreet* составляет 28 %. При этом для обеих последовательностей эффективность от введения дополнительных режимов внутреннего предсказания достигает всего 18 и 14 % соответственно, т. е. 43 и 50 % от общего повышения эффективности.

Видно, что новый стандарт *H.265/HEVC* позволяет на 30...40 % эффективнее стандарта *H.264/AVC* сжимать видеопоследовательности уже с помощью опорных видеокадров (см. графики *HM 16-2* и *JM* на рис. 11, 12). При этом 40...50 % эффективности достигается за счет увеличенного набора внутренних предсказаний (см. графики *HM 16-2* и *HM*

16-2-AVC на рис. 11, 12). Остальное повышение степени сжатия достигается за счет иных улучшений стандарта, в том числе измененной модели энтропийного кодирования.

Выводы

По мнению разработчиков, целью создания стандарта *H.265/HEVC* являлось повышение степени сжатия видеопоследовательностей до двух раз по сравнению с возможностями стандарта *H.264/AVC* при том же качестве. Только на сжатии опорных кадров стандарт *H.265/HEVC* дает до 40 % увеличения степени сжатия при том же качестве. При этом до 50 % повышения эффективности достигается за счет введения дополнительных вариантов внутреннего предсказания, оставшиеся, по-видимому, достигаются за счет модификаций энтропийного кодирования и других усовершенствований стандарта. Такое существенное повышение эффективности сжатия уже на ключевых кадрах позволяет говорить о допустимости двукратного увеличения эффективности сжатия стандарта *H.265/HEVC* по сравнению со стандартом *H.264/AVC*. Для изучения этого вопроса необходимо исследовать возможности стандарта *H.265/HEVC* по сжатию различных кадров.

СПИСОК ЛИТЕРАТУРЫ

1. Generic coding of moving pictures and associated audio information: Video // MPEG-2 ISO/IEC 13818-2. — 1998. — P. 202.
2. Advanced video coding for generic audiovisual services // Recommendation ITU-T H.264. — 2011. — P. 686.
3. Bross B. et al. High-Efficiency Video Coding (HEVC) text specification draft 9 // JCT-VC: 11th meeting. — Shanghai, CN, 10–19 October, 2012. — 390 p. URL: http://phenix.int-evry.fr/jct/doc_end_user/documents/11_Shanghai/wg11/JCTVC-K1003-v13.zip (дата обращения: 16.12.2012).
4. Шарабайко М.П., Осокин А.Н. Быстродействующий алгоритм фрактального сжатия изображений // Известия Томского по-

литехнического университета. — 2011. — Т. 318. — № 5. — С. 52–57.

5. Сэлмон Д. Сжатие данных, изображений и звука. — М.: Техносфера, 2004. — 368 с.
6. Richardson I. White Paper: H.264/AVC Intra Prediction // Vcodex. — 2011. — P. 1–7. URL: http://www.vcodex.com/files/H264_intraped_wp.pdf (дата обращения: 16.12.2012).
7. Bossen F. Common test conditions and software reference configurations // JCT-VC: 6th Meeting. — Torino, IT, 14–22 July 2011. — 5 p. URL: http://phenix.it-sudparis.eu/jct/doc_end_user/documents/6_Torino/wg11/JCTVC-F900-v1.zip (дата обращения: 16.12.2012).

Поступила 16.12.2012 г.

УДК 681.3.06

ИССЛЕДОВАНИЕ АППАРАТНЫХ РЕАЛИЗАЦИЙ ТАБЛИЧНОГО И МАТРИЧНОГО АЛГОРИТМОВ ВЫЧИСЛЕНИЯ CRC32

Е.А. Мыцко, А.Н. Мальчуков

Томский политехнический университет

E-mail: EvgenRus70@mail.ru, jgs@tpu.ru

Приведено описание аппаратных реализаций матричного и табличного алгоритмов вычисления контрольной суммы CRC32 на программируемых логических интегральных схемах Cyclone фирмы Altera макета SDK-6.1. Показаны особенности аппаратной реализации на примере описания блоков вычисления CRC32 и работоспособность спроектированных устройств на конкретных примерах. Проведено исследование алгоритмов на основе сравнения блоков вычисления CRC32 по занимаемым логическим ячейкам и временным задержкам.

Ключевые слова:

Контрольная сумма, табличный алгоритм, матричный алгоритм, CRC32, аппаратная реализация.

Key words:

Check sum, table-driven algorithm, matrix-driven algorithm, CRC32, hardware implementation.

В литературе [1] можно встретить описание алгоритмов вычисления контрольной суммы CRC32, адаптированных для программной реализации. Однако данные алгоритмы можно адаптировать под аппаратные средства для контроля целостности данных при передаче на различные устройства. В данном случае рассматривается аппаратная реализация CRC32 на примере программируемых логических интегральных схем (ПЛИС) Cyclone фирмы Altera [2] макета SDK-6.1 [3].

Особенности аппаратной реализации алгоритмов

В настоящее время для создания цифровых устройств широко используются ПЛИС. Сферами их применения обычно являются цифровая обработка сигналов, цифровая видео-аудио аппаратура, высокоскоростная передача данных, криптография, коммутаторы между системами с различной логикой и напряжением питания, проектирование и прототипирование ASIC, реализация нейронных сетей [4].

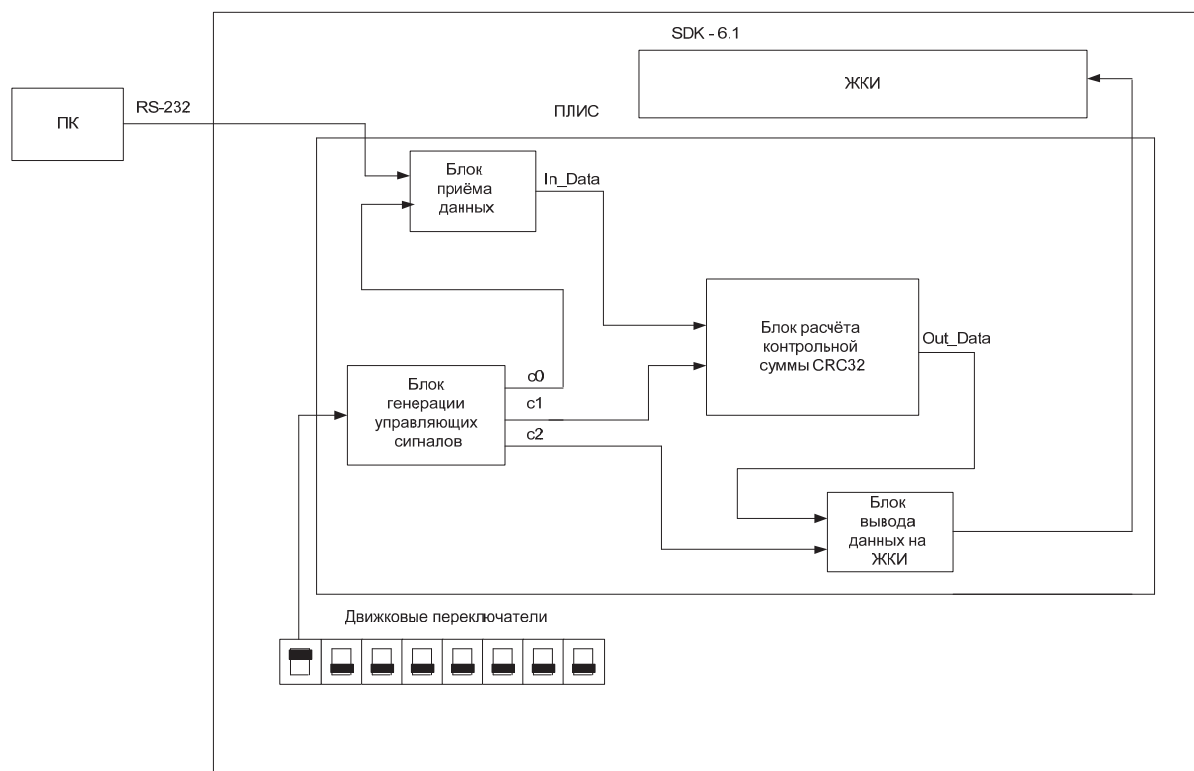


Рис. 1. Структурная схема устройства расчёта контрольной суммы CRC32

На кафедре вычислительной техники Томского политехнического университета студенты обучаются разработке устройств на базе современных ПЛИС фирмы Altera [2] в рамках дисциплины «Схемотехника ЭВМ». Для этих целей используются учебные стенды SDK 6.1 [3] на базе Altera Cyclone EP1C3T144, имеющие движковые переключатели, светодиоды, кнопки, двухстрочный жидкокристаллический индикатор (ЖКИ) и последовательный порт. В связи с этим для аппаратной реализации алгоритмов вычисления контрольной суммы CRC32 выбран макет SDK-6.1.

Ввод данных в макет для расчёта контрольной суммы CRC32 производится с персонального компьютера (ПК). Передача данных с ПК на макет осуществляется по последовательному интерфейсу RS-232, используя терминал для передачи данных (term_1b.exe). CRC32 рассчитывается для блока данных и записывается в регистр. Итоговая рассчитанная контрольная сумма отображается на жидкокристаллической индикации (ЖКИ) макета SDK-6.1 при переводе движкового переключателя № 0 (крайний слева) в верхнее положение. На структурной схеме устройства (рис. 1) 4 основных блока. В блоке приёма данных осуществляется приём последовательности битов данных от ПК по интерфейсу RS-232, определение старт- и стоп-битов, а также контроль единичного уровня сигнала при передаче. Блок расчёта контрольной суммы вычисляет контрольную сумму CRC32 для данных, поступивших с блока приёма, и записывает её в регистр. Блок вывода на ЖКИ осуществляет перевод значения контрольной суммы в шестнадцатеричную форму и выводит её на дисплей SDK-6.1. При этом во время расчёта CRC32 блок генерации управляющих импульсов осуществляет синхронизацию всех блоков схемы.

Табличный алгоритм

На основе структурной схемы, используя блочно-ориентированный подход, в среде Quartus II спроектирована функциональная схема устройства расчёта контрольной суммы CRC32 с использованием языка описания аппаратуры VHDL [5].

Основным блоком в функциональной схеме является блок расчёта CRC32 «CRC32_vhdl» (рис. 2).

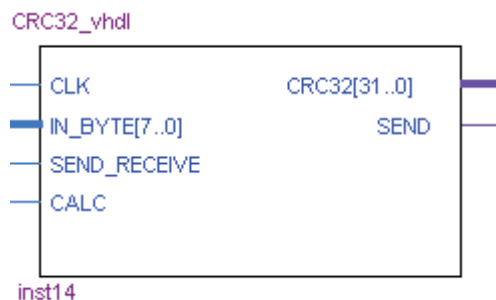


Рис. 2. Блок расчёта CRC32 табличным алгоритмом

Блок расчёта CRC32 (рис. 2) содержит 4 входа и 2 выхода. Вход CLK служит для подачи управляю-

щего синхроимпульса частотой 40 МГц на блок. IN_BYTE [7..0] является входным байтом данных для расчёта контрольной суммы. Управляющий сигнал SEND_RECEIVE устанавливается с помощью движкового переключателя, что позволяет задать режим расчёта CRC32 или выдачи результата на ЖКИ. При нулевом уровне сигнала SEND_RECEIVE (нижнем положении переключателя) осуществляется приём данных по байту с последующим расчётом контрольной суммы и записью её в регистр. При единичном уровне сигнала SEND_RECEIVE (верхнем положении переключателя) рассчитанная контрольная сумма поступает на выход CRC32 [31..0] для выдачи её на ЖКИ SDK-6.1. Вход CALC служит для получения сигнала расчёта CRC от блока «main_vhdl» для принятого байта. Выходной сигнал SEND служит для управления блоком преобразования контрольной суммы из символического представления в шестнадцатеричное.

Матричный алгоритм

Функциональная схема и блок расчёта CRC32 для однобайтового матричного алгоритма выглядят аналогично табличному алгоритму. Отличия заключаются только в описании работы блока вычисления CRC32 на языке VHDL.

Матричный двухбайтовый алгоритм является модификацией однобайтового матричного алгоритма. Для расчёта CRC32 необходимо формировать данные по 2 байта и передавать их на блок расчёта контрольной суммы. Соответственно, блок расчёта CRC32 при аппаратной реализации будет отличаться визуально (рис. 3) и по функциональному описанию.

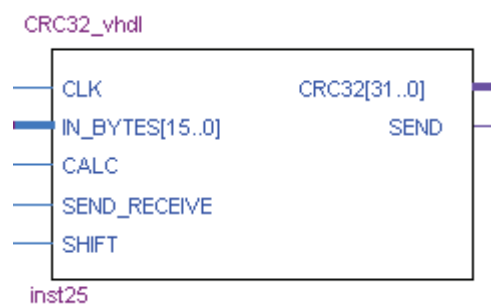


Рис. 3. Блок расчёта CRC32 матричным двухбайтовым алгоритмом

Данный блок имеет шестнадцатиразрядный информационный вход IN_BYTES [15..0] и в отличие от приведённого выше блока для табличного и матричного однобайтового алгоритма имеет дополнительный вход SHIFT, который служит для задания режима однобайтового (при единичном уровне сигнала) или двухбайтового (при нулевом уровне) расчёта CRC32. Это связано с тем, что объём данных в байтах не всегда кратен 2, поэтому для оставшихся байтов нужно применять однобайтовую схему расчёта.

В программной реализации особенность матричного четырёхбайтового алгоритма заключа-

лась в том, что сдвиг данных осуществлялся блоками по 4 байта. Поэтому функциональная схема для аппаратной реализации будет отличаться от описанных ранее алгоритмов. Изменения заключаются в том, что из последовательной передачи данных по порту RS-232 нужно формировать блоки данных по 4 байта, для которых будет рассчитываться контрольная сумма. Таким образом, была спроектирована функциональная схема, реализующая расчёт CRC32 данным алгоритмом.

Блок расчёта CRC32 четырёхбайтовым матричным алгоритмом (рис. 4) содержит 5 входов и 2 выхода. Вход CLK, как и ранее, служит для подачи на блок управляющего синхроимпульса частотой 40 МГц. IN_BYTES [31..0] является информационным тридцатидвухразрядным входом для расчёта CRC32. Двухразрядный вход SHIFT [1..0] выполняет функцию выбора способа расчёта CRC32 в зависимости от кратности набора данных. Пока на информационный вход поступают блоки по 4 байта, данный входной сигнал имеет значение «00». Если же на информационный вход поступает неполный блок данных 1, 2 или 3 байта, то на входе формируется соответствующий двухразрядный сигнал расчёта по однобайтовой («01»), двухбайтовой («10») или трёхбайтовой («11») схеме, что необходимо для расчёта CRC32 при наборах данных не кратных четырём.

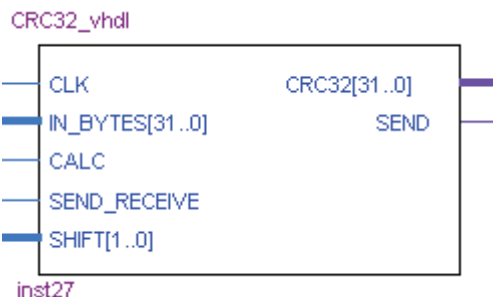


Рис. 4. Блок расчёта CRC32 четырёхбайтовым матричным алгоритмом

Примеры расчёта контрольной суммы CRC32

Далее приведены некоторые примеры расчёта CRC32 с применением SDK-6.1. Для расчёта CRC32 был выбран файл CRCfile.exe, который передавался на макет через последовательный порт (рис. 5). Результат расчёта контрольной суммы CRC32 для файла CRCfile.exe на макете SDK-6.1 приведён на рис. 6.

На примере архиватора WinRAR, в котором используется алгоритм CRC32, можно удостовериться в правильности расчёта контрольной суммы (рис. 7).

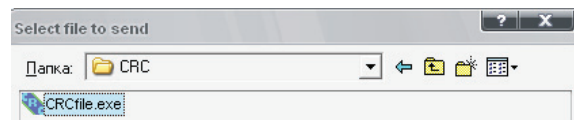


Рис. 5. Выбор файла для передачи на SDK-6.1

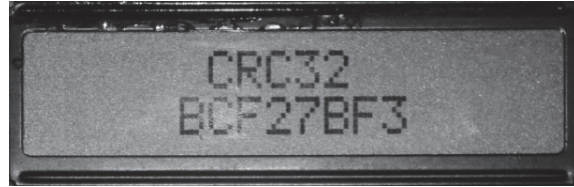


Рис. 6. ЖКИ SDK-6.1 с контрольной суммой CRC32 для файла «CRCfile.exe»

Исследование аппаратной реализации алгоритмов

Для проектирования схем в среде Quartus II фирмы Altera, используя блочно-ориентированный подход, применяют два типа логики: комбинаторную и регистровую. Логическая ячейка (Logic Cell) — общий термин для основных блоков микросхем, на которых реализуется проект. Логическая ячейка состоит из элемента регистровой логики (триггера) и универсального четырехвходового логического элемента LUT (Look Up Table), который может быть запрограммирован на реализацию любой четырехвходовой логической функции. Именно на этих элементах реализуется вся комбинаторная логика проекта в ПЛИС. Если нужна функция больше 4-х входов, то используется два таких элемента, больше 7 входов — три ячейки. Регистр (триггер) может быть сконфигурирован как latch (защёлка), D, T, RS, JK триггер или зашунтирован для реализации только комбинаторной логики [6].

В среде проектирования Quartus II для каждого блока функциональной схемы приводится ряд значений, таких как Logic Cells (общее количество ячеек для элемента), LC Registers (общее количество регистров, входящих в ячейку), LUT-Only LCs (количество элементов только комбинаторной логики, входящих в ячейку), Register-Only LCs (количество элементов только регистровой логики, входящих в ячейку), LUT/Register LCs (количество элементов, использующих как комбинаторную, так и регистровую логику).

Для исследования аппаратной реализации на основе полученных данных была составлена сводная таблица занимаемых логических ячеек для блока вычисления контрольной суммы «CRC32_vhdl» в зависимости от алгоритма (табл. 1).

Как видно из табл. 1, значение Logic Cells состоит из суммы значений LC Registers и LUT-Only LCs, а значение LC Registers, в свою очередь, состоит из суммы значений Register-Only LCs и LUT/Register LCs.

Имя	Размер	Сжат	Тип	Изменён	CRC32
Папка					
CRCfile.exe	151 552	60 772	Приложение	15.01.2000 12:20	BCF27BF3

Рис. 7. CRC32 для файла CRCfile.exe в архиваторе WinRAR

Таблица 1. Количество занимаемых логических ячеек для блока «CRC32_vhdl»

Алгоритм/лог. ячейки	Количество занимаемых ячеек, шт.				
	Logic Cells	LC Registers	LUT-Only LCs	Register-Only LCs	LUT/Register LCs
Табличный	125	65	60	0	65
МС 1 байт	125	65	60	0	65
МС 2 байта	169	65	105	0	65
МС 4 байта	280	66	214	0	66

По полученным данным была составлена таблица изменений количества занимаемых логических ячеек в процентах для аппаратных реализаций различных вариантов матричного алгоритма относительно табличного (табл. 2).

Таблица 2. Изменение количества логических ячеек относительно табличного алгоритма

Алгоритм/лог. ячейки	Изменение количества логических ячеек, %				
	Logic Cells	LC Registers	LUT-Only LCs	Register-Only LCs	LUT/Register LCs
МС 1 байт	0	0	0	0	0
МС 2 байта	+35	0	+73	0	0
МС 4 байта	+124	+1,5	+256	0	+1,5

Из табл. 2 видно, что реализация матричного однобайтового алгоритма в данном случае требует столько же логических ячеек, как и для реализации табличного алгоритма.

При реализации двухбайтового матричного алгоритма общее количество логических ячеек увеличивается на 35 %, а количество элементов комбинаторной логики – на 73 %. Для реализации четырёхбайтового матричного алгоритма потребовалось на 256 % больше элементов комбинаторной логики. Данные изменения связаны с особенностью аппаратной реализации матричных алгоритмов. При реализации двухбайтового и четырёхбайтового алгоритмов увеличивается разрядность блока (с 8 до 16 и 32 бит соответственно), который вычисляет контрольную сумму CRC32. Также на блоке расчёта CRC32 появляется дополнительный вход для вычисления контрольной суммы в случаях, когда объём данных не кратен требуемому при реализации определённого варианта матричного алгоритма. Данные факторы приводят к увеличению количества LUT элементов, требуемых для построения блока вычисления CRC32.

Преимущество многобайтовых матричных алгоритмов заключается в уменьшении числа тактов, требуемых для расчёта CRC32. Так, если при передаче данных доступны блоки по 4 байта, то для расчёта CRC32 четырёхбайтовым матричным алгоритмом требуется в 4 раза меньше тактов, чем по табличному алгоритму или матричному однобайтовому алгоритму.

Помимо количества логических ячеек, занимаемых блоком вычисления CRC32, также исследованы временные задержки вычислительных блоков для каждого алгоритма.

На основе полученных данных составлена таблица временных задержек блоков вычисления CRC32 для каждого алгоритма (табл. 3). При этом учитывалось, что если в системе доступно сразу по 2 байта для обработки за такт, то задержка блоков однобайтовых алгоритмов увеличивается в 2 раза. При доступных 4-х байтах для обработки за такт задержка блока двухбайтового алгоритма увеличивается в 2 раза, а для однобайтовых – в 4 раза.

Таблица 3. Временные задержки для блока вычисления CRC32

Алгоритм	Задержка при передаче, нс		
	последовательной побайтной	по 2 байта	по 4 байта
Табличный	7,308	14,616	29,232
Матричный 1 байт	7,309	14,618	29,236
Матричный 2 байта	8,627	8,627	17,254
Матричный 4 байта	11,025	11,025	11,025

На основе табл. 3 была составлена таблица ускорений блоков вычисления CRC32 матричного алгоритма относительно табличного (табл. 4).

Таблица 4. Ускорение блока вычисления CRC32 матричного алгоритма относительно табличного

Матричный алгоритм	Ускорение при передаче, %		
	последовательной побайтной	по 2 байта	по 4 байта
1 байт	0	0	0
2 байта	-18	+69	+69
4 байта	-50	+32	+165

Как видно из табл. 4, блок вычисления CRC32 матричного однобайтового алгоритма имеет такую же задержку данных, как и для табличного алгоритма, в то время как блок вычисления матричного четырёхбайтового алгоритма отстаёт по скорости от табличного на 50 % при последовательной побайтной передаче данных. В случаях, когда при передаче за цикл доступно больше одного байта, многобайтовые алгоритмы опережают по скорости однобайтовые.

Таблица 5. Значения времени расчёта CRC32 при аппаратной реализации

Алгоритм	Значения времени расчёта CRC32 при аппаратной реализации, с					
	Объём использованных файлов, Мб					
	10	210	410	610	810	1010
Табличный	0,076	1,609	3,141	4,674	6,207	7,739
Матричный (сдвиг 1 байт)	0,076	1,609	3,141	4,674	6,207	7,739
Матричный (сдвиг 2 байта)	0,045	0,949	1,854	2,759	3,663	4,568
Матричный (сдвиг 4 байта)	0,028	0,606	1,184	1,762	2,341	2,919

В табл. 5 представлены значения времени расчёта для файлов различного объема при асинхронной передаче, исходя из задержки блока вычисления.

Таблица 6. Значения времени расчёта CRC32 при аппаратной реализации

Алгоритм	Среднее значение времени расчёта CRC32, с (Доверительные интервалы, %)					
	Объём использованных файлов, Мб					
	10	210	410	610	810	1010
Табличный	0,057 ($\pm 1,4$)	1,194 ($\pm 0,217$)	2,299 ($\pm 0,203$)	3,227 ($\pm 0,162$)	4,334 ($\pm 0,211$)	6,184 ($\pm 0,155$)
Матричный (сдвиг 1 байт)	0,125 ($\pm 0,732$)	2,81 ($\pm 0,112$)	5,399 ($\pm 0,067$)	7,602 ($\pm 0,166$)	10,23 ($\pm 0,161$)	14,557 ($\pm 0,048$)
Матричный (сдвиг 2 байта)	0,092 ($\pm 3,882$)	2,084 ($\pm 0,133$)	4,02 ($\pm 0,09$)	5,645 ($\pm 0,105$)	7,566 ($\pm 0,097$)	10,835 ($\pm 0,072$)
Матричный (сдвиг 4 байта)	0,08 ($\pm 0,827$)	1,729 ($\pm 0,172$)	3,32 ($\pm 0,011$)	4,694 ($\pm 0,314$)	6,272 ($\pm 0,188$)	8,965 ($\pm 0,064$)

На основании значений времени расчета контрольной суммы при аппаратной реализации и значений, полученных при исследовании программных реализаций (табл. 6) [7], была составлена табл. 7, в которой отражена разница по времени расчета CRC32 для программной и аппаратной реализаций.

Таблица 7. Разница по времени расчета CRC32 для программной и аппаратной реализаций

Алгоритм	Разница по времени расчета CRC32 для программной и аппаратной реализаций, с					
	Объём использованных файлов, Мб					
	10	210	410	610	810	1010
Табличный	-0,019	-0,415	-0,842	-1,447	-1,873	-1,555
Матричный (сдвиг 1 байт)	0,049	1,201	2,258	2,928	4,023	6,818
Матричный (сдвиг 2 байта)	0,047	1,135	2,166	2,886	3,903	6,267
Матричный (сдвиг 4 байта)	0,052	1,123	2,136	2,932	3,931	6,046

Из табл. 7 видно, что для табличного алгоритма время расчета в программной реализации меньше, чем в аппаратной. Для остальных вариантов алгоритма аппаратная реализации опережает по скорости вычисления программную приблизительно на одну и ту же величину, о чем говорит положительная разница по времени вычисления.

Заключение

На основе алгоритмов расчёта контрольной суммы CRC32, описанных в литературе [1], были

спроектированы устройства для аппаратных реализаций алгоритмов на ПЛИС Cyclone. При проектировании функциональной схемы использовался блочно-ориентированный подход (BBD). Было установлено, что для однобайтовых алгоритмов (табличный и матричный) количество задействованных логических ячеек для блока вычисления CRC32 одинаково. Для двухбайтового и четырёхбайтового алгоритмов блок вычисления CRC32 имеет существенные отличия от блоков однобайтовых алгоритмов. Для их реализации потребовалось значительно больше логических ячеек (на 35 % для двухбайтового и на 124 % для четырёхбайтового алгоритма).

По результатам анализа временных задержек блоков вычисления CRC32 можно сказать, что при аппаратной реализации однобайтовые алгоритмы имеют одинаковую скорость вычисления, в то время как у двухбайтового матричного алгоритма задержка блока вычисления на 18 % больше, чем у табличного, а четырёхбайтовый алгоритм отстаёт по скорости на 50 %. Однако при доступном блоке данных, который можно обрабатывать за такт объёмом в 4 байта, матричный четырёхбайтовый алгоритм значительно опережает по скорости (на 165 %) однобайтовые алгоритмы.

Данные аппаратные средства с использованием полученных конфигураций с реализованными алгоритмами расчёта CRC32 позволяют осуществлять контроль целостности данных при передаче по последовательному порту.

Исследование выполнено при поддержке Министерства образования и науки Российской Федерации, соглашение 14.B37.21.0457.

СПИСОК ЛИТЕРАТУРЫ

1. Ross N.W. A Painless Guide to CRC Error Detection Algorithms. 1993. URL: http://www.ross.net/crc/download/crc_v3.txt (дата обращения: 01.05.2011).
2. EPIC3T144C8 datasheet // DATASHEET.SU. 2007. URL: <http://pdf3.datasheet.su/Altera/EPIC3T144C8.pdf> (дата обращения: 06.08.2012).
3. Учебный лабораторный стенд SDK-6.1 // Embedded systems. URL: <http://embedded.ifmo.ru/index.php/support/sdk-61> (дата обращения: 06.08.2012).
4. Логовский А. Технология ПЛИС и ее применение для создания нейрочипов // Издательство «Открытые системы». 2012.

URL: <http://www.osp.ru/os/2000/10/178242/> (дата обращения: 06.08.2012).

5. Библио П.Н. Основы языка VHDL. 3-е изд., доп. — М.: Изд-во ЛКИ, 2007. — 328 с.
6. Мьяльк Р., Шестаков В. Технология проектирования цифровых устройств на ПЛИС. — СПб.: Изд-во ГУАП, 2005. — 75 с.
7. Мычко Е.А., Мальчуков А.Н. Исследование программных реализаций алгоритмов вычисления CRC совместимых с PKZip, WinZip, Ethernet // Известия Томского политехнического университета. — 2013. — Т. 322. — № 5. — С. 170–175.

Поступила 09.01.2013 г.

ОЦЕНИВАНИЕ ЧИСЛА КЛЕТОК НА ИЗОБРАЖЕНИЯХ ЦИТОЛОГИЧЕСКИХ РАСТИТЕЛЬНЫХ ПРЕПАРАТОВ

М.С. Тарков, М.И. Осипов*

Институт физики полупроводников СО РАН им. А.В. Ржанова, г. Новосибирск

E-mail: tarkov@isp.nsc.ru

*Новосибирский государственный университет

E-mail: inq1201@mail.ru

Предложен алгоритм для подсчета вакуолей (растительных клеток) на изображении. Алгоритм позволяет выделить контуры клеток в массивах слипшихся клеток путем разбиения диапазона яркости изображения на слои с последующим применением в каждом слое полутоновой эрозии и дилатации, преобразования «верх шляпы» и пороговой сегментации. При подсчете клеток используется аппроксимация их контуров эллипсами. Алгоритм может быть использован в различных приложениях, например при определении скорости распада клеточных структур по серии изображений.

Ключевые слова:

Оценивание числа клеток, цитологические растительные препараты, анализ изображений.

Key words:

Estimating the number of cells, cytological herbal preparations, image analysis.

Введение

В настоящее время анализ изображений цитологических препаратов осуществляется преимущественно вручную, что сильно замедляет процесс, а плохое качество изображения или наличие на изображении большого числа клеток увеличивает вероятность ошибки при выяснении тех или иных характеристик препарата. В частности, при большом объеме (10^2 – 10^3) выборки подсчет клеток вручную может занять много времени, а также возможен некорректный подсчет.

Использование автоматических методов анализа изображений цитологических препаратов может значительно улучшить качество анализа и ускорить его, однако универсальный алгоритм разработать трудно в силу разнородности изображений. Клеточные структуры могут выглядеть совершенно по-разному, иметь прозрачную, полупрозрачную или непрозрачную структуру, обладать ярко выраженными границами или не обладать таковыми. Поэтому целесообразно разрабатывать алгоритмы под конкретные классы изображений, что повысит их эффективность, но, очевидно, применение их будет возможно только в пределах соответствующих классов.

Разработка алгоритмов для подсчета числа клеток ведется, и для некоторых классов изображений они уже есть [1, 2], но такие алгоритмы обычно основаны на сегментации, то есть целью является сопоставление клетки с конкретным сегментом изображения. Такой подход эффективен в том случае, когда клетки непрозрачны, то есть на изображении цвет такой клетки значительно отличается от цвета фона, что позволяет выяснить, является ли сегмент клеткой. К тому же при наличии шума на изображении, а при съемке цитологических препаратов это не редкость, возможно появление множества мелких сегментов, сходных по цветовым характеристикам с клетками, вследствие чего получится некачественный подсчет. Существуют классы изображений, для которых использование алгоритмов, основанных на сегмента-

ции, неприемлемо. Один из таких классов образуют изображения вакуолей (растительных клеток) [3].

Целью данной работы является разработка алгоритма для подсчета числа вакуолей на изображении. При разработке необходимо учитывать специфику изображений данного класса, поэтому необходимо решить ряд проблем, а именно (рис. 1):

1. Клетки могут слипаться или перекрываться, вследствие чего несколько клеток могут восприниматься как одна, поэтому необходимо решить проблему разделения слипшихся клеток.
2. Клетки полупрозрачны, поэтому цветовые характеристики клеток и фона практически совпадают, что препятствует использованию сегментации изображения классическими способами, так как затруднительно по цветам пикселей некоторого сегмента определить, является ли он клеткой или нет.
3. Необходимо решить проблему отделения клеток от фона, так как в некоторых случаях клетки имеют нечеткую границу.

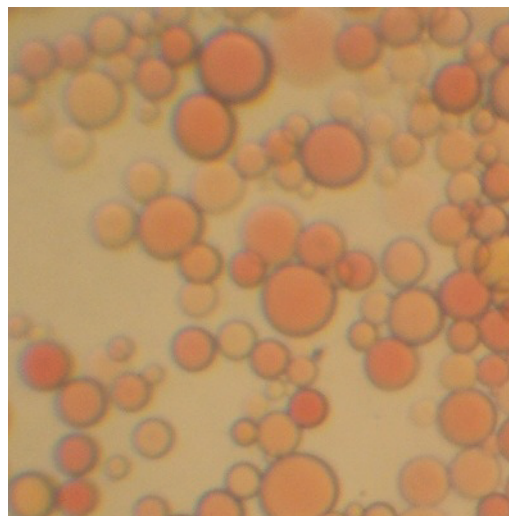


Рис. 1. Фрагмент снимка растительных клеток

Используемый подход

Для решения задачи используется подход, предложенный в [1]. После некоторых модификаций, связанных со спецификой рассматриваемого нами класса изображений, этот подход применим к изображениям растительных клеток (вакуолей). Особенность подхода заключается в том, что после выделения контуров клеток исходное изображение больше не используется. Для дальнейшего анализа используется бинарное изображение контуров. Анализируются геометрические свойства полученных контуров, что позволяет выделять слипшиеся клетки даже в случае, когда граница между ними плохо различима.

В алгоритме можно выделить два этапа:

1. *Выделение и преобработка контуров клеток.* Выделение контуров осуществляется с помощью комбинированного использования морфологических преобразований эрозии и дилатации [4], а также порогового преобразования карты евклидовых расстояний. Контур аппроксимируется многоугольниками: из каждого контура выделяются точки, являющиеся вершинами аппроксимирующего многоугольника. На контурах отыскиваются точки вогнутости. Этими точками контур сегментируется на части.
2. *Выделение клеток с помощью эллипсов.* Строятся эллипсы, аппроксимирующие контуры клеток. Выявляются эллипсы, которые не могут аппроксимировать клетку (слишком узкие или слишком длинные). Объединяются сегменты. Общаются эллипсы, не отобранные на этапе селекции. Они объединяются с другими наиболее подходящими эллипсами. Производится коррекция выделения и подсчет эллипсов.

Алгоритм выделения контуров

Перед тем как приступить непосредственно к нахождению контуров, входное цветное изображе-

ние преобразуется в монохромное. Далее осуществляется следующая последовательность действий:

1. Применение морфологических операций эрозии и дилатации с ядром 7×7 .
2. Выравнивание гистограммы.
3. Сглаживание изображения путем свертки его с гауссовым ядром 3×3 .
4. Вычитание фона.
5. Выделение десяти слоев (интервалов яркости) размером 20 ($[0, 19]$, $[20, 39]$, ..., $[180, 199]$).
6. Каждый слой подвергается последовательному применению морфологических операций дилатации и эрозии для удаления шумовых выбросов. Также производится пороговое преобразование евклидовой карты расстояний для выделения контуров. Выделение контуров для каждого слоя производится независимо.

Вычитание фона является необходимой процедурой. Её задача — не допустить, чтобы в рассматриваемые слои попали фрагменты фона (рис. 2). Рис. 2 показывает, что после вычитания фона значительная его часть не попадает в рассматриваемые далее слои яркости. Чтобы выделить фон на изображениях данного класса необходимо [3]:

1. Преобразовать исходное изображение в монохромное.
2. Осуществить медианную фильтрацию.
3. Выровнять гистограмму.
4. Взять дополнение.
5. Построить размыкание изображения. В качестве структурообразующего элемента используется круг с радиусом 25 пикселей.
6. Вычесть размыкание из изображения, полученного в ходе шагов 1–4 (преобразование «верх шляпы» [4]).
7. Осуществить пороговое преобразование (порог подобран эмпирически и равен 25) (рис. 3).

Пороговое преобразование карты евклидовых расстояний [5] в данном случае применяется к бинарным изображениям (каждый пиксель является либо черным, либо белым). Карта евклидовых рас-

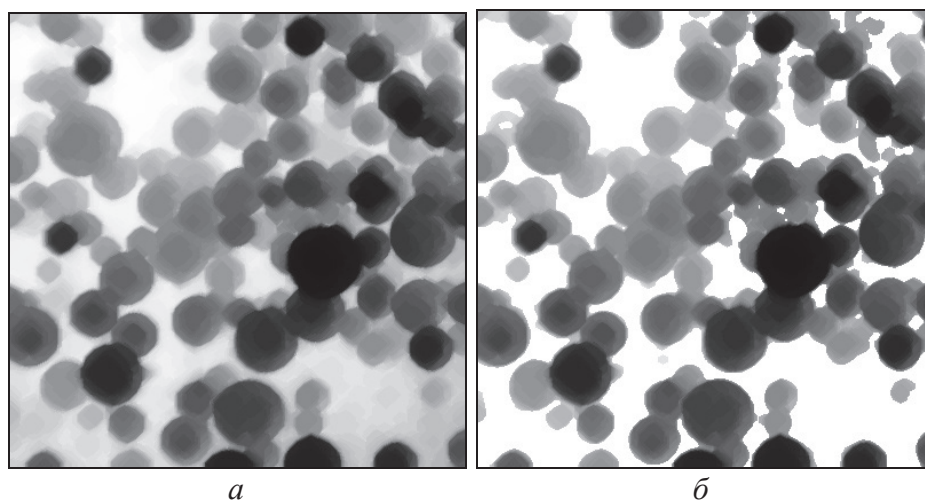


Рис. 2. Изображение рис. 1: а) после действий 1–3; б) после действий 1–4

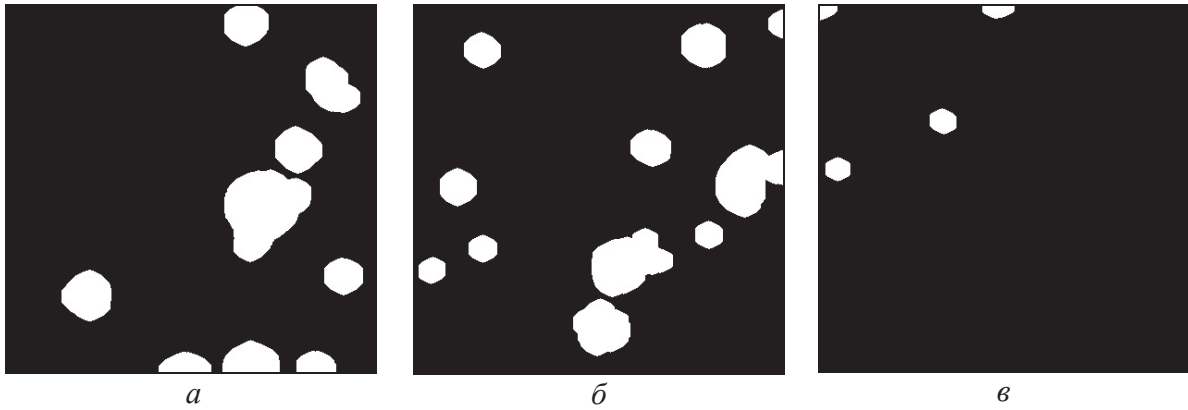


Рис. 3. Слои изображения рис. 2, б: а) слой [0, 19]; б) слой [20, 39]; в) слой [180, 199]

стояний изображения составляется таким образом: для каждого пикселя вычисляется расстояние до ближайшего черного пикселя (если пиксель черный, оно равно нулю). Пороговое преобразование состоит в изменении цвета пикселя с белого на черный, если его расстояние до ближайшего черного пикселя не попадает в заданный промежуток значений.

Аппроксимация контуров многоугольниками

Аппроксимация контуров многоугольниками основана на двух соображениях. Первое – сгладить контуры, чтобы при поиске точек вогнутости не возникало много ложных точек. Второе – уменьшить число точек контуров. После аппроксимации, как правило, количество точек контура уменьшается. Как следствие, уменьшается количество информации о геометрии контура и сокращается объем последующих вычислений.

Для аппроксимации используется алгоритм Дугласа–Пекера [6]. Алгоритм рекурсивно делит линию. Входом алгоритма служат координаты всех точек контура кроме первой и последней, которые сохраняются неизменными. Далее алгоритм находит точку, наиболее удаленную от прямой, проведенной через первую и последнюю точки контура. Если точка находится на расстоянии меньше, чем ε , то все точки, которые ещё не были отмечены к сохранению, могут быть выброшены из набора, и получившаяся прямая сглаживает кривую с точностью не ниже ε . Если же расстояние больше ε , то алгоритм рекурсивно вызывает себя на наборах точек от начальной до данной и от данной до конечной. Это означает, что данная точка будет отмечена к сохранению. По окончании всех рекурсивных вызовов результирующая ломаная строится только из тех точек, что были отмечены к сохранению.

Поиск точек вогнутости и сегментация контуров

Точки вогнутости – это точки, разделяющие части контура, которые принадлежат различным клеткам. Обозначим $p_c(x_c, y_c)$ текущую точку, а $p_{pre}(x_{pre}, y_{pre})$ и $p_{next}(x_{next}, y_{next})$ предыдущую и следующую точки соответственно. Далее определим величину $concavity(p_c)$ в точке p_c [1]:

$$a(p_{pre}, p_c) = \tan^{-1}((y_{pre} - y_c) / (x_{pre} - x_c));$$

$$a(p_{next}, p_c) = \tan^{-1}((y_{next} - y_c) / (x_{next} - x_c));$$

$$concavity(p_c) = \begin{cases} |a(p_{pre}, p_c) - a(p_{next}, p_c)|, \\ \text{если } |a(p_{pre}, p_c) - a(p_{next}, p_c)| < \pi, \\ \pi - |a(p_{pre}, p_c) - a(p_{next}, p_c)|, \text{ иначе.} \end{cases}$$

Условия, при выполнении которых точка p_c считается точкой вогнутости, выглядят так [1]:

1. $a_1 < concavity(p_c) < a_2$;
2. Отрезок $\overline{p_{pre}p_{next}}$ не должен проходить внутри контура.

Значения a_1 и a_2 задаются пользователем.

Сегментация контуров предусматривает разделение контуров на части найденными точками вогнутости (рис. 4). Отдельные части контуров предположительно являются частями контуров отдельных клеток. По этим фрагментам осуществляется построение эллипсов [1]. Однако сегменты контура в ходе дальнейшей работы алгоритма могут объединяться, поэтому их первоначальное количество в общем случае не равно числу клеток, обнаруженных в области, контур которой рассматривается.

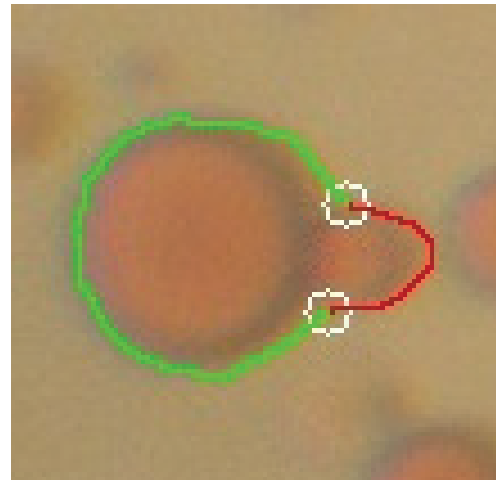


Рис. 4. Пример контура из двух сегментов, разделенных точками вогнутости (точки вогнутости обведены белыми окружностями)

Выделение клеток

Основной целью аппроксимации эллипсами является разделение выделенных областей на отдельные клетки. На данном этапе мы имеем набор контуров, каждый из которых разделен на сегменты точками вогнутости. Изначально для каждого сегмента строится свой эллипс. Далее выполняется анализ геометрических свойств эллипсов и их расположения относительно друг друга. Целью анализа является уточнение результатов, полученных при первоначальном построении.

Построение эллипса производится для каждого фрагмента каждого контура. Сам фрагмент представляет собой набор точек, которые по предположению принадлежат контуру клетки. Однако эти точки образуют только часть реального контура, а построенный эллипс должен дополнить картину (рис. 5).

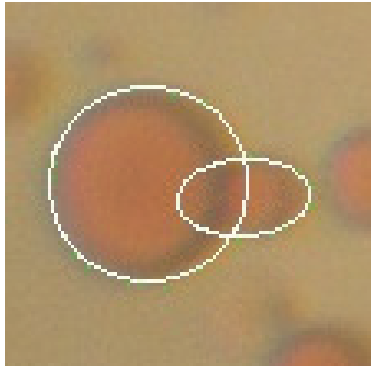


Рис. 5. Пример эллипсов, построенных по точкам сегментов

Рассмотрим уравнение кривой второго порядка:

$$(A, X) = 0,$$

где $A = [a, b, c, d, e, f]$, $X = [x^2, xy, y^2, x, y, 1]$. Если для точки (x_i, y_i) сформировать вектор $X_i = [x_i^2, x_i y_i, y_i^2, x_i, y_i, 1]$, то можно определить алгебраическое расстояние (A, X_i) от точки X_i до кривой $(A, X) = 0$.

Коэффициенты для искомого эллипса находятся методом наименьших квадратов, то есть решается задача условной оптимизации [7]. Минимизируется сумма квадратов алгебраических расстояний при условии, что получившаяся в результате кривая второго порядка – эллипс:

$$\begin{cases} E(A) = \sum_{i=1}^N (A, X_i)^2 \rightarrow \min_A, \\ (a+c) \cdot \begin{vmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{vmatrix} < 0, \\ b^2 - 4ac < 0, \end{cases}$$

где N – количество точек фрагмента контура.

Минимизация проводится по вектору коэффициентов квадратичной формы $A = [a, b, c, d, e, f]$, а ограничения-неравенства гарантируют, что в результате минимизации получится эллипс.

Селекция

Процедура селекции заключается в проверке двух условий для каждого эллипса. Если эти условия выполнены, эллипс участвует в следующем шаге – объединении сегментов. Остальные эллипсы обрабатываются отдельно.

Выбираются эллипсы, которые достаточно точно аппроксимируют контур, и те, которые имеют приемлемую форму для аппроксимации контура клетки, то есть не являются слишком узкими или слишком вытянутыми. Чтобы эллипс участвовал в следующем шаге, необходимо выполнение следующих двух условий [1]:

1. Обозначим сегмент контура $L_i = \{p_1, \dots, p_{i_s}\}$, $p_{ij} = (x_{ij}, y_{ij})$, а соответствующий этому сегменту эллипс $F(A_i, X)$. Определим алгебраическое расстояние $dis(A_i, L_i)$ от точек сегмента L_i до соответствующего ему эллипса:

$$dis(A_i, L_i) = \frac{1}{s} \sum_{j=1}^s |(A_i, X_{ij})|.$$

Первое условие: величина $dis(A_i, L_i)$ должна быть меньше заранее выбранного порога $disTh$.

2. Обозначим $MaxA$ и $MinA$ длины большей и малой полуосей эллипса соответственно. Второе условие: отношение $eRatio = MinA/MaxA$ должно быть не меньше заданного порога eTh , что означает близость эллипса к окружности. Порог eTh необходимо выбирать, исходя из формы рассматриваемых клеток.

Объединение сегментов

На данном шаге объединяются сегменты, эллипсы которых построены по фрагментам контуров одной клетки (рис. 6). Берется два фрагмента, строится новый эллипс по точкам из этих фрагментов, проверяется выполнение условий селекции, и в зависимости от ситуации принимается решение: оставить два изначально построенных эллипса или заменить их новым эллипсом, объединив рассматриваемые сегменты.

Возможны следующие ситуации [1]:

1. Явно соприкасающиеся клетки должны быть разделены, т. е. сегменты объединяться не должны. Для идентификации данной ситуации необходимо пользоваться следующими признаками:
 - а) Расстояние между центром нового эллипса C_{new} и центрами старых эллипсов C_1 и C_2 должно быть больше, чем величина $dMinTh$, то есть

$$\|C_i - C_{new}\| > dMinTh, i = 1, 2.$$

При соблюдении этого условия эллипсы не объединяются. В качестве $dMinTh$ используется величина, близкая к длине меньшей полуоси наименьшей клетки.

- б) Расстояние между центрами старых эллипсов слишком велико для того, чтобы они были частями одной клетки, а именно

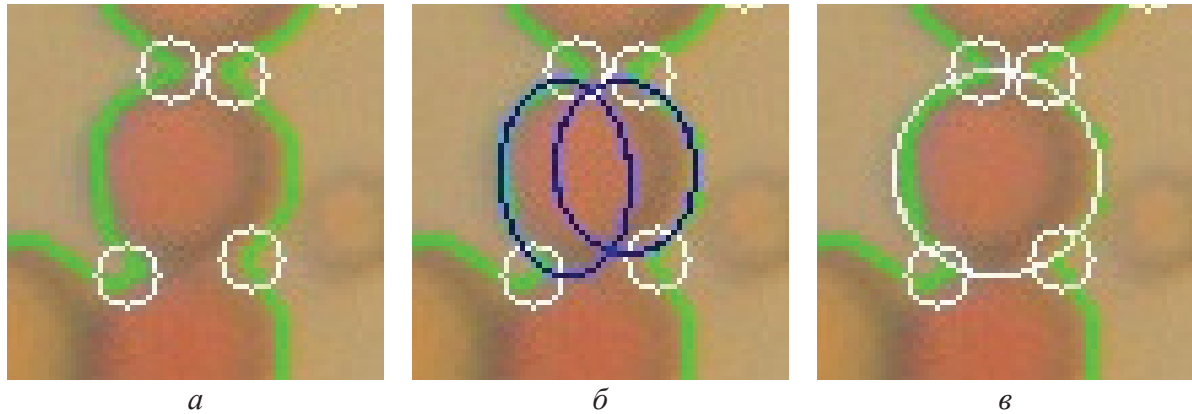


Рис. 6. Пример работы механизма объединения сегментов: а) сегментированный контур; б) эллипсы, построенные первоначально; в) объединение сегментов ввиду того, что они представляют одну клетку

$$\|C_1 - C_2\| > 3 \cdot dMinTh.$$

При выполнении этого условия эллипсы также не объединяются.

2. Два несмежных сегмента должны быть объединены, то есть нужно заменить два старых эллипса новым, построенным по точкам обоих сегментов. Этот случай идентифицируется следующими признаками:

- а) Длины меньших полуосей старых эллипсов меньше величины $dMinTh$, то есть эллипсы слишком малы для аппроксимации контура какой-либо клетки, поэтому их нужно объединить.
- б) Эллипсы, построенные по сегментам контура схожи, то есть длины их больших и малых полуосей соответственно близки:

$$\begin{aligned} |MaxA_1 - MaxA_2| &< dMinTh, \\ |MinA_1 - MinA_2| &< dMinTh / 2. \end{aligned}$$

Такие эллипсы объединяются в один, так как, вероятно, они аппроксимируют одну и ту же клетку.

3. В остальных случаях необходимо пользоваться следующим правилом: если $dis(C_{new}, L_{new}) < dis(C_1, L_1)$ или $dis(C_{new}, L_{new}) < dis(C_2, L_2)$, где $L_{new} = L_1 \cup L_2$, то старые эллипсы заменяются новым.

Обработка эллипсов, не отобранных в ходе селекции

На данном шаге рассматриваются эллипсы, для которых условия селекции не выполнены (необработанные сегменты). Поскольку они являются частями контуров, то соответствующие им сегменты объединяются с наиболее подходящими сегментами эллипсов, прошедших процедуру селекции (т. е. с обработанными сегментами).

Пусть для текущего контура мы имеем m необработанных сегментов $L_1, \dots, L_m$ и n обработанных $E_1, \dots, E_n$. Берется сегмент $L_i, i=1..m$, строится последовательность P_{ij} точек L_i и $E_j, j=1..n$. По последовательности точек P_{ij} строится эллипс T_{ij} и считается величина $dis(T_{ij}, P_{ij})$. Находится сегмент $E_k, k \in \{1, \dots, n\}$, для которого эта величина наименьшая. Объединение L_i с E_k происходит только в том

случае, если эллипс, построенный по точкам двух сегментов, удовлетворяет условиям селекции, иначе сегменты не объединяются, а сегмент L_i исключается из дальнейшего рассмотрения.

Коррекция выделения клеток и подсчет

Растительные клетки (вакуоли) имеют форму круга. Поэтому для лучшего визуального восприятия результирующее выделение нужно сделать при помощи окружностей, а не эллипсов. По полученному набору эллипсов достаточно просто строится набор окружностей следующим образом: по каждому эллипсу строится окружность с центром, совпадающим с центром эллипса, а радиус окружности равен полусумме полуосей эллипса.

Контурные клетки выделяются из различных слоев яркости. При этом возможны попадания разных частей одной и той же клетки в разные слои. В этом случае произойдет наложение окружностей друг на друга. Данный шаг нужен для отслеживания чрезмерного наложения. Рассматриваются все окружности. Если площадь пересечения некоторого круга с другими кругами составляет более 75 % собственной площади, то данный круг удаляется и в дальнейшем не рассматривается. Результирующее число клеток равно числу оставшихся кругов.

Эксперименты

При разработке алгоритма большое внимание уделялось подбору различных параметров, от которых существенно зависит качество работы алгоритма. Подбор осуществлялся, исходя из геометрии клеточных структур и специфики рассматриваемых изображений. Алгоритм программно реализован в среде Visual Studio C++ с использованием процедур обработки изображений библиотеки OpenCV [8]. В табл. 1 приведены значения параметров алгоритма, при которых получены удовлетворительные результаты. Здесь ε – параметр аппроксимации в алгоритме Дугласа–Пекера, a_1 и a_2 – параметры алгоритма поиска точек вогнутости (в градусах), $disTh$ и $dMinTh$ – пороги расстояний, eTh – порог отношения полуосей эллипса.

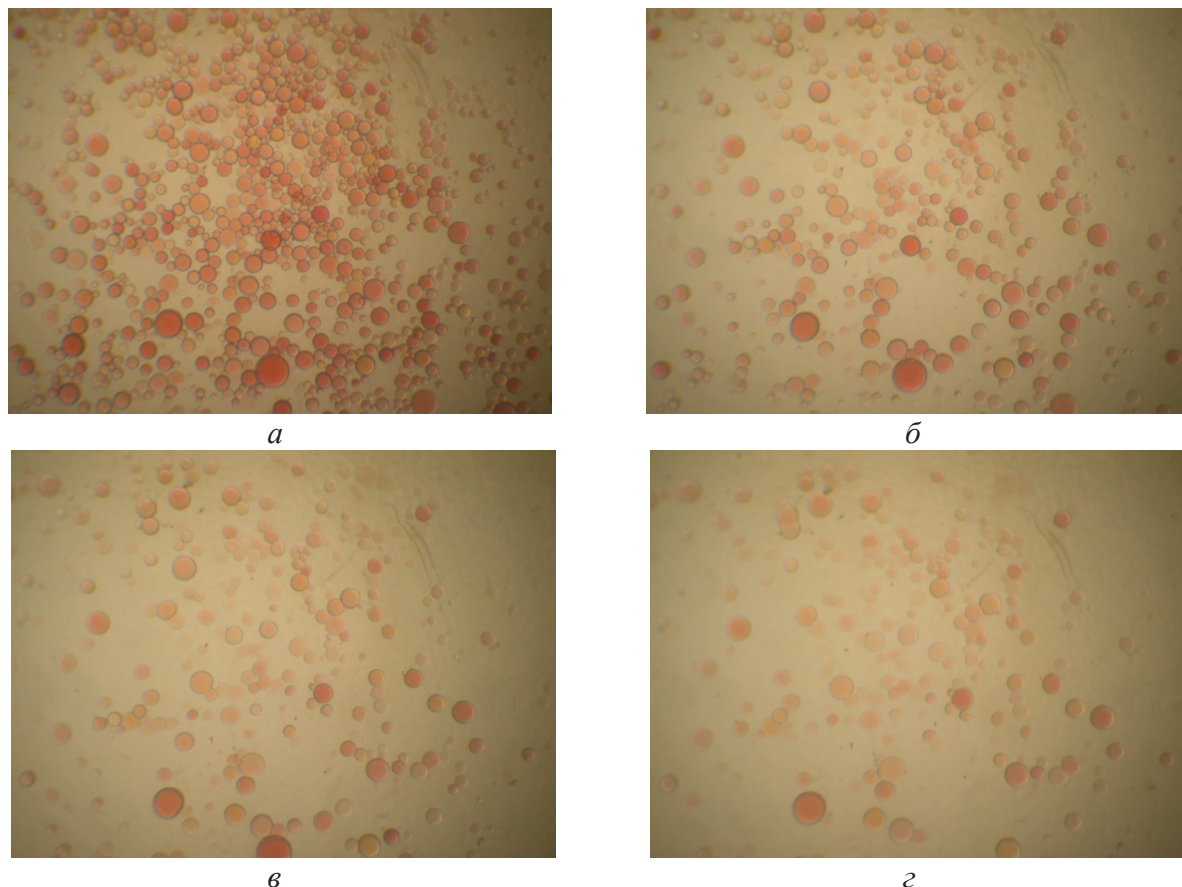


Рис. 7. Тестовые образцы изображений растительных клеток

Таблица 1. Значения параметров алгоритма подсчета клеток

Параметр	ε	a_1	a_2	$disTh$	eTh	$dMinTh$
Значение	2	20	160	0,1	0,1	8

В табл. 2 и 3 приведены результаты работы алгоритма для изображений рис. 7, *a–г* при различных параметрах, используемых в процессе извлечения контуров. В скобках указано отклонение в процен-

тах от результата, полученного подсчетом вручную. На основе экспериментальных данных выбраны оптимальные параметры алгоритма, дающие минимальную по последовательности кадров суммарную ошибку подсчета клеток. При выборе способа разбиения слоев учитывался тот факт, что основные части клеток находятся в диапазоне яркости от 0 до 200. Слой в диапазоне яркости от 201 до 255 содержит фон и мелкие фрагменты клеток.

Таблица 2. Результаты работы алгоритма при различных размерах ядра операторов эрозии и дилатации, применяемых при извлечении контуров

Размер ядра	3×3	5×5	7×7	9×9	Результат подсчета вручную
Изображение (рис. 7)					
а	879 (0,1 %)	817 (7,1 %)	822 (6,5 %)	594 (32,5 %)	880
б	480 (18,2 %)	451 (11,0 %)	391 (3,6 %)	354 (12,8 %)	406
в	370 (37,9 %)	351 (28,1 %)	303 (10,5 %)	291 (6,2 %)	274
г	342 (69,3 %)	324 (60,3 %)	278 (37,6 %)	264 (30,6 %)	202

Таблица 3. Результаты работы алгоритма при различном количестве и толщине слоев яркости, на которые разбивается изображение при извлечении контуров

Кол-во слоев×толщина (пикс.)	2×100	5×40	10×20	12×15	Результат подсчета вручную
Изображение (рис. 7)					
а	508 (42,2 %)	683 (22,3%)	822 (6,5%)	612 (30,4%)	880
б	284 (30,0%)	410 (0,9%)	391 (3,6%)	365 (10,1%)	406
в	367 (33,9%)	322 (17,5%)	303 (10,5%)	287 (4,7%)	274
г	249 (23,2%)	292 (44,5%)	278 (37,6%)	265 (31,1%)	202

В качестве тестовых образцов использовались и изображения размером 1600×1200 пикселей (рис. 7). Алгоритм тестировался на компьютере с процессором Intel Core i3–2310M, 2.1 GHz. Время обработки одного изображения равно 7,3 с. На рис. 8 представлен результат работы алгоритма. Количество обнаруженных клеток – 822.

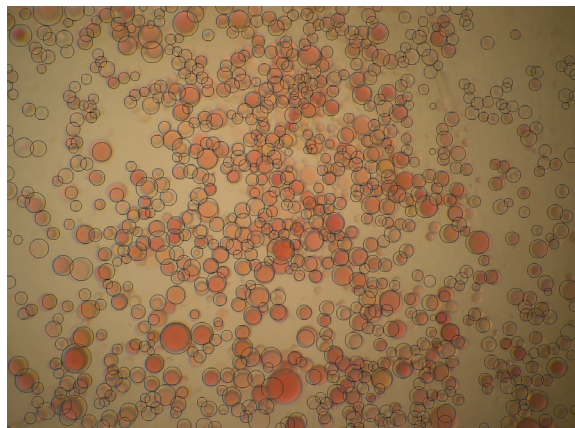


Рис. 8. Пример результата работы алгоритма выделения клеток

Заключение

Разработан алгоритм подсчета растительных клеток (вакуолей) на изображениях клеточных препаратов. Алгоритм основан на идее выделения контуров клеток с последующей аппроксимацией контуров эллипсами (окружностями). Известные подходы, основанные на сегментации изображений, неприменимы к подсчету объектов на изображениях растительных клеток, что подтверждено экспериментами. Реализован метод выделения контуров слипшихся клеток. В алгоритме применен ряд дополнительных операций, таких как вычитание фона и коррекция выделения клеток, что позволило значительно повысить качество работы алгоритма.

Экспериментально подобраны параметры, обеспечивающие приемлемое качество работы алгоритма. Алгоритм реализован в среде Visual Studio C++ с использованием процедур обработки изображений библиотеки OpenCV. Алгоритм примечателен тем, что вся работа с исходным изображением заканчивается после выделения контуров, степень четкости границ клеток (если они слипшиеся) не играет роли, так как дальше анализируется только геометрия контуров.

Авторы выражают благодарность В.Н. Нурминскому (СИФИБР СО РАН) за предоставленные изображения клеточных препаратов.

СПИСОК ЛИТЕРАТУРЫ

1. Bai X., Sun C., Zhou F. Splitting touching cells based on concave points and ellipse fitting // Pattern Recognition. – 2009. – № 42. – P. 2434–2446.
2. Ковригин А.В. Применение принципов построения систем машинного зрения в задаче анализа изображений клеточных структур // Научный журнал КубГАУ. – 2007. – № 29 (5). URL: <http://ej.kubagro.ru/2007/05/pdf/03.pdf> (дата обращения: 12.03.2012).
3. Тарков М.С., Тихонов Н.В., Половинкин В.Г. Оценивание состава изображений клеточных препаратов для медико-биологических исследований // Известия Томского политехнического университета. – 2012. – Т. 320. – № 5. – С. 130–133.
4. Гонсалес Р., Вудс Р. Цифровая обработка изображений. – М.: Техносфера, 2005. – 1072 с.
5. Danielsson P.-E. Euclidean distance mapping // Computer graphics and image processing. – 1980. – № 14. – P. 227–248.
6. Douglas D., Peucker T. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature // The Canadian Cartographer. – 1973. – V. 10. – № 2. – P. 112–122.
7. Fitzgibbon A., Pilu M., Fisher R.B. Direct Least Square Fitting of Ellipses // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 1999. – V. 21. – № 5. – P. 476–480.
8. Bradski G., Kaehler A. Learning OpenCV. – USA: O'Reilly Media, Inc., 2008. – 576 p.

Поступила 03.03.2013 г.

УДК 519.175.1

ЗАДАЧА ОПРЕДЕЛЕНИЯ ОЦЕНОК СХОДСТВА СТРУКТУР ДВУХ ГРАФОВ НА ОСНОВЕ ВЫДЕЛЕНИЯ ОБЩИХ ЧАСТЕЙ

В.К. Погребной

Томский политехнический университет

E-mail: vkp@tpu.ru

Предложен алгоритм решения задачи определения оценок сходства структур, представленных обыкновенными графами, на основе выделения у них общих частей в виде изоморфных подграфов или частичных графов. Алгоритм работает в классе графов с одинаковыми векторами начальных описателей.

Ключевые слова:

Обыкновенный граф, сходство графов, изоморфизм графов, частичный изоморфизм, структурные различия, однородные группы.

Key words:

Ordinary graph, graph similarity, graph isomorphism, partial isomorphism, structural differences, homogeneous groups.

Введение

Решая задачу определения изоморфизма двух графов [1] или гиперграфов [2], мы получаем ответ — являются графы изоморфными или нет. Для изоморфных графов указывается соответствие вершин, при котором их инциденты совпадают. В работе [3] рассматривается более общая проблема описания структур графов и анализа их сходства. Предлагаются методы свободной и зависимой интеграции структурных различий и получения интегральных описателей структур (ISD). При этом изоморфизм двух графов рассматривается как частный случай, соответствующий полному сходству структур графов, когда их ISD совпадают.

Во многих практических задачах, связанных с анализом структур, более важной является возможность оценить частичное сходство двух структур. Данная задача является более сложной и может рассматриваться в разных вариантах:

- выделение в графах общей части в виде подграфа;
- выделение общей части в виде частичного графа;
- определение оценок сходства на основе анализа интегральных описателей.

Методы определения оценок сходства в каждом из вариантов могут различаться в зависимости от параметров сравниваемых графов — имеют они одинаковое или разное число вершин и (или) рёбер, совпадают или нет у них исходные векторы описателей. ISD-технология анализа подобия структур [3] предполагает разработку методов определения указанных видов оценок сходства. Первая из этих оценок отражает наличие у сравниваемых графов общей части, которая для них обоим одновременно является подграфом. Вторая оценка даёт более полное представление о сходстве графов, когда в качестве общей части выступает частичный граф. В этом случае число рёбер, содержащихся в частичном графе, может служить мерой сходства сравниваемых графов. Например, если

сравниваемые графы содержат по 20 рёбер, а общий для них частичный граф — 15 рёбер, то сходство данных графов можно оценить отношением $15/20$ или 75 %.

Статья посвящена разработке метода выделения в двух графах общего для них частичного графа, содержащего наибольшее число рёбер. При этом рассматривается самый простой случай, когда сравниваемые графы имеют равные исходные векторы описателей и, соответственно, одинаковое число вершин и рёбер. Задача выделения частичного графа и получения соответствующей оценки сходства решается в составе ISD-технологии и рассматривается как задача определения наибольшей общей части двух графов.

Метод решения задачи выделения в графах общей части включает последовательное выполнение двух этапов (двух алгоритмов). На первом из них выделяется общая часть в виде изоморфных подграфов. Второй этап посвящён расширению общей части, полученной на первом этапе, до связного частичного графа или частичного подграфа, если некоторые вершины окажутся с нулевой степенью. При решении данной задачи используется метод интеграции структурных различий (метод ISD) [3]. Важное значение имеет также использование результатов, полученных при исследовании свойств однородных групп. Эти исследования, наряду с выделением в общей части изоморфных подграфов, сделали возможным оценивание близости сравниваемых графов на уровне однородных групп.

Однородные группы и отношения между ними

Изучение однородных групп удобнее вести на обыкновенных графах [4]. При этом результаты исследований легко переносятся на графы других типов, в том числе на графовые модели, представленные атрибутными матрицами [5]. Формально однородная группа определяется как совокупность вершин графа, у которых описатели, полученные в результате выполнения очередного шага интеграции структурных различий, совпадают.

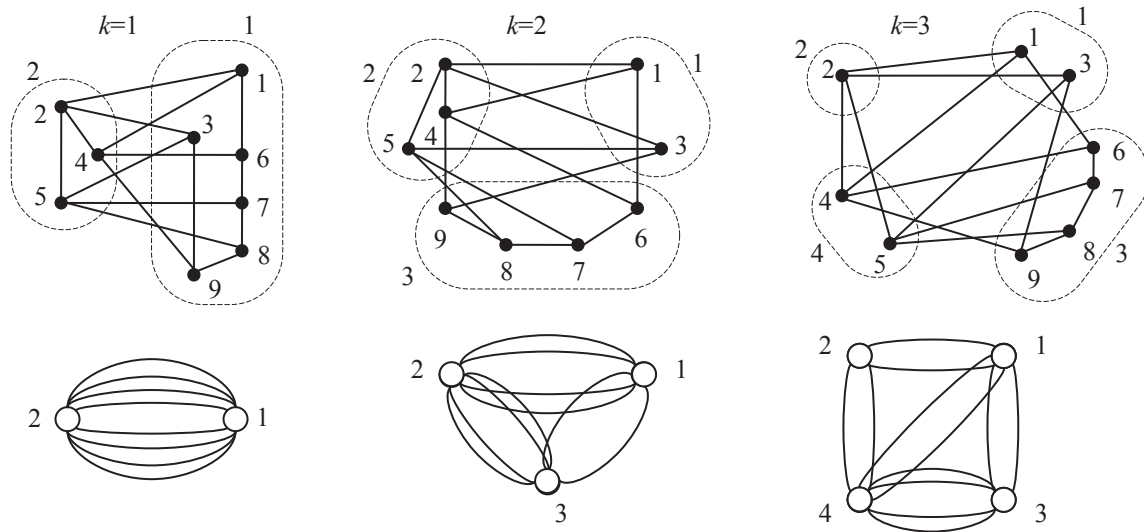


Рис. 1. Примеры однородных групп и результаты дифференциации их вершин

Пример визуального представления трёх шагов свободной интеграции с выделением однородных групп приведён на рис. 1. Однородные группы получены в результате применения формальных правил интеграции структурных различий в условиях, когда атрибуты отсутствуют. В этом случае все вершины $e_i \in E$ и рёбра $r_{ij} \in R$ обыкновенного графа $G=(E, R)$ являются неразличимыми, а их начальные характеристики описываются вектором $D^0=\{d_i^0\}$ с элементами $d_i^0=1$.

Совокупность описателей вершин, соответствующих инциденту вершины e_i на k -м шаге интеграции, обозначим Z_i^k . В нашем примере описатели $d_i^0=1$ для всех вершин $e_i \in E$, поэтому на первом шаге интеграции элементы в совокупностях Z_i^0 также равны 1. Дифференциация вершин на данном шаге происходит по степеням вершин (по числу элементов в Z_i^0). В результате получают группы вершин $E^k(d_i^k)$ с разными описателями d_i^k . В одну группу $E^1(d_i^1=1)=(e_1, e_3, e_6, e_7, e_8, e_9)$ попадают вершины со степенью 3, а в другую $E^1(d_i^1=2)=(e_2, e_4, e_5)$ – со степенью 4. Вершинам со степенью 3 присваивается описатель $d_i^1=1$, а со степенью 4 – описатель $d_i^1=2$. Соответственно, для $k=1$ имеем две группы $E^1(1)$ и $E^1(2)$, которые на рис. 1 обозначены своими описателями.

Несмотря на то, что при $k=0$ в каждой группе вершины имели одинаковые описатели, для $k=1$ легко обнаруживаются их структурные различия. Так, в группе $E^1(1)$ для вершины e_1 совокупность $Z_1^1=(1, 2, 2)$, а для e_6 – $Z_6^1=(1, 1, 2)$. Аналогично различие имеет место и в группе $E^1(2)$. Например, для вершины e_2 совокупность $Z_2^1=(1, 1, 2, 2)$, а для e_3 имеем $Z_3^1=(1, 1, 1, 2)$. Однородную группу, вершины которой имеют разные Z_i^k , будем именовать нестабильной или дифференцируемой. Для $k=1$ обе группы являются нестабильными, и следовательно возможна их дифференциация. Результат дифференциации вершин группы $E^1(1)$ на шаге $k=2$ приведён на рис. 1. Здесь группа $E^2(1)=(e_1, e_3)$ в отличие

от групп $E^2(2)=(e_2, e_4, e_5)$ и $E^2(3)=(e_6, e_7, e_8, e_9)$ является стабильной, т. к. $Z_1^2=Z_3^2=(2, 2, 3)$. Дифференциация группы $E^2(2)=(e_2, e_4, e_5)$ на шаге $k=3$ приводит к выделению вершины e_2 с уникальным описателем $d_2^3=2$, который включается в состав вектора ISD. Вершины (e_4, e_5) образуют стабильную группу $E^3(4)$ с $Z_4^3=Z_5^3=(1, 2, 3, 3)$.

При анализе причин нестабильности однородной группы будем различать неоднородность отношений между вершинами внутри группы и отношений с вершинами внешнего окружения. Соответственно, нестабильные группы будем подразделять на группы с внутренней и (или) внешней нестабильностью. В частности, группа $E^3(3)$ так же, как и группа $E^1(2)$, является одновременно внутренне и внешне нестабильной. Нетрудно убедиться, что после дифференциации вершин группы $E^3(3)$ группа $E^3(4)$ из стабильного состояния переходит во внешне нестабильное. В общем случае возможности перехода однородной группы из одного состояния в другое определяются следующими свойствами:

- стабильная группа в результате дифференциации вершин в группах её окружения может перейти во внешне нестабильное состояние;
- внутренняя нестабильность группы в отличие от внешней не может быть приобретена в ходе дифференциации вершин в группах её окружения;
- внутренняя и внешняя нестабильность группы не могут быть утеряны в ходе дифференциации вершин в группах её окружения;
- дифференциация вершин группы независимо от её состояния может привести к появлению групп в любом состоянии.

Заметим, что перечисленные свойства не совсем очевидны. Они сформулированы исходя из анализа метода интеграции структурных различий. При этом предполагается, что в последующих исследованиях будут более строго обоснованы эти и другие свойства однородных групп.

Приведём еще одно свойство, которое связано со степенями вершин в стабильных группах. Это свойство определяет условие существования таких групп и также является очевидным. В стабильной группе с числом вершин n и суммарным числом рёбер между вершинами равным r степени у всех вершин по данным рёбрам одинаковы и могут принимать значения $s=0,1,2,\dots$. При этом должно соблюдаться условие $s=2r/n$. Из условия следует, что степень $s=0$ возможна при $r=0$ и произвольном $n \geq 2$. Для $s=1$ должно соблюдаться условие $2r=n$, т. е. для $s \geq 1$ число вершин в группе всегда должно быть четным. Выделим аналогичное свойство и относительно внешних рёбер, связывающих вершины стабильной группы с вершинами, окружающими группу. Любая вершина стабильной группы связана с вершинами за пределами группы равным числом рёбер. Таким образом, в стабильной группе степени вершин по внутренним и внешним рёбрам совпадают.

При дифференциации вершин нестабильной группы часть её внутренних рёбер становится внешними. Например, после дифференциации вершин группы $E^1(1)$ (рис. 1) внутренние рёбра $r_{1,6}$ и $r_{3,9}$ для групп $E^2(1)$ и $E^2(3)$ стали внешними. Исследования структуры отношений между однородными группами важны при получении оценок сходства двух графов на уровне однородных групп. Для этого строятся и исследуются графы связности однородных групп. Примеры таких графов, построенных для групп, полученных на соответствующих шагах интеграции, приведены в нижней части рис. 1. Графы связности однородных групп используются при решении задачи определения сходства двух графов на основе анализа интегральных описателей. В частности, о степени сходства двух графов можно судить по анализу изоморфизма соответствующих графов связности однородных групп. Такие исследования отнесены к варианту решения задачи на основе анализа интегральных описателей и в данной работе не рассматриваются.

Алгоритм выделения в графах общей части в виде изоморфных подграфов

Задачу сравнения графов G и H с целью выделения общего для них подграфа и определения на этой основе оценки сходства будем рассматривать как задачу определения их частичного изоморфизма с помощью метода ISD. В этом случае предполагается, что в векторах ISD у графов G и H совпадают не все описатели, а только их часть, которая соответствует общему подграфу. Проблема здесь заключается в том, что по методу ISD описатели вершин общего подграфа определяются на основе интеграции структурных различий относительно всех вершин графа. Это приводит к тому, что вершины неизоморфных частей графов G и H вносят «искажения» в описатели вершин изоморфной части и выделение соответствующего подграфа становится невозможным.

Ниже предлагается алгоритм, который позволяет локализовать неизоморфные части графов, маскируя структурные различия в этих частях, и тем самым исключает искажающее воздействие на описатели вершин изоморфных частей графов. Локализация изоморфных частей и маскировка структурных различий осуществляется с помощью метода ISD на основе выделения и сопоставления однородных групп. В результате для графов G и H получаются векторы интегральных описателей (ISD), в которых каждая вершина изоморфных частей получает уникальное значение описателя, а все вершины неизоморфных частей окажутся сосредоточены в специальных однородных группах с маскирующими значениями описателя.

Дадим некоторые пояснения к таким группам. Однородные группы $E_G^k(d^k)$ и $E_H^k(d^k)$ в графах G и H будем считать эквивалентными, если при равных описателях d^k число вершин в них совпадает, а между вершинами $e_i \in E_G^k(d^k)$ и $e_j \in E_H^k(d^k)$ имеет место взаимно-однозначное соответствие, при котором $Z_i^k(G) = Z_j^k(H)$. Свойство эквивалентности и неэквивалентности рассматривается только относительно групп, принадлежащих разным графам, независимо от стабильности или нестабильности данных групп. Таким образом, при работе алгоритма будем различать стабильные эквивалентные группы (С), нестабильные эквивалентные группы (Н), неэквивалентные стабильные и нестабильные группы (НЭ), которые были отмечены выше как специальные. Заметим также, что группы типа (НЭ) соответствуют не изоморфным частям графов и выполняют роль «убежища» для структурных различий, которые маскируются путём использования одинаковых описателей вершин внутри группы.

Алгоритм включает выполнение следующих операций.

1. Сравняются обыкновенные графы G и H при отсутствии каких-либо атрибутов у вершин и рёбер. В этом случае векторы начальных описателей $D^0(G)$ и $D^0(H)$ принимаются единичными и, следовательно, каждый граф представлен одной однородной группой $E_G^0(1)$ и $E_H^0(1)$. Последующие шаги интеграции в одном из графов, например в G , будут выполняться как свободные, а в другом — как зависимые. После выполнения первого шага интеграции ($k=1$) согласно условиям задачи получим векторы исходных описателей $D^1(G) = D^1(H)$, которые отражают дифференциацию вершин по степеням. Если все вершины графов G и H имеют одинаковые степени, то группы $E_G^0(1)$ и $E_H^0(1)$ относятся к типу (С) и для дифференциации их вершин в векторы $D^1(G)$ и $D^1(H)$ вводятся абстрактные описатели по правилам, изложенным в [3]. Выполняя шаги интеграции относительно новых векторов $D_i^0(G)$ и $D_j^0(H)$, получим векторы $D^1(G) = D^1(H)$, в которых вершины разбиты на две однородные группы.

2. Выполняется анализ векторов $D^k(G)=D^k(H)$ и классификация содержащихся в них однородных групп $E_G^k(d^k)$ и $E_H^k(d^k)$ по типам (С), (Н), (НЭ). Группы типа (НЭ) не подлежат дифференциации и сохраняются до завершения работы алгоритма. Группы типа (С) ожидают возможной дифференциации своих вершин. Среди групп типа (Н) выбирается группа с наибольшим числом вершин. Для вершин этой группы в графах G и H выполняется $(k+1)$ -й шаг интеграции относительно векторов $D^k(G)$ и $D^k(H)$. Полученные при этом описатели d^{k+1} определяют дифференциацию вершин в группе. Если группа типа (Н) отсутствует, то среди групп типа (С) выбирается группа с наибольшим числом вершин и для неё выполняется дифференциация с помощью введения абстрактного описателя. При отсутствии групп типа (С) и (Н) работа алгоритма завершается.
3. Выполняется формирование векторов $D^{k+1}(G)$ и $D^{k+1}(H)$. На шаге интеграции $k=1$ все описатели d^1 векторов $D^1(G)$ и $D^1(H)$ определяются в ходе дифференциации вершин в группах $E_G^0(1)$ и $E_H^0(1)$, выполняемой в первой операции данного алгоритма. В этом случае формирование векторов $D^1(G)$ и $D^1(H)$ происходит автоматически. На шаге $k>1$ после дифференциации в группе типа (Н) или (С) векторы $D^{k+1}(G)$ и $D^{k+1}(H)$ содержат только описатели d^{k+1} вершин этой группы. Для вершин с уникальными описателями и вершин из других групп типа (НЭ), (Н), (С), описатели d^k из векторов $D^k(G)$, $D^k(H)$ переносятся в векторы $D^{k+1}(G)$ и $D^{k+1}(H)$ без изменений. Сформированные таким обра-

зом векторы анализируются во второй операции.

В тех случаях, когда при выполнении операции 2 выделяется несколько групп типа (Н) или (С), операции 2 и 3 выполняются для каждой из этих групп. Повторное выполнение операций 2 и 3 производится также в случаях введения абстрактных описателей при дифференциации вершин в группах типа (С).

В результате выполнения операций данного алгоритма вершины, получившие уникальные описатели, составляют общую часть графов G и H в виде изоморфных подграфов, а вершины, входящие в группы типа (НЭ), представляют неизоморфные части графов.

На рис. 2 приведён пример выполнения алгоритма решения задачи выделения общих частей для графов G и H , содержащих по 9 вершин и по 15 рёбер. Исходные векторы $D^1(G)$ и $D^1(H)$ графов по условиям задачи должны совпадать, что и подтверждается выполнением первой операции.

Во второй операции определяется, что группы $E^1(1)$ и $E^1(2)$ в графах G и H относятся к типу (Н) и группа $E^1(1)$, содержащая больше вершин, дифференцируется в первую очередь. После этого третья операция формирует векторы $D^2(G)$ и $D^2(H)$. Анализ данных векторов (операция 2) указывает на наличие трёх групп $E^2(1)$ типа (С), $E^2(2)$ типа (НЭ) и $E^2(3)$ типа (Н). После дифференциации группы $E^2(3)$ и формирования векторов $D^3(G)$ и $D^3(H)$ в них, помимо группы $E^2(2)$ типа (НЭ), выделяются 3 группы типа (С) равной мощности. Поэтому для дифференциации с помощью введения абстрактного описателя выбирается любая из них.

Граф G

		D^0	D^1	D^2	D^3	D_1^3	D^4	$D^5 Z^5(G)$
	1 2 3 4 5 6 7 8 9							
1		1		1				
2	1		1	1				
3		1			1			1
4	1	1				1		1
5		1	1				1	1
6	1			1				
7				1	1		1	
8					1		1	1
9			1	1				1

Граф H

		D^0	D^1	D^2	D^3	D_6^3	D^4	$D^5 Z^5(H)$
	1 2 3 4 5 6 7 8 9							
1				1	1	1		1
2			1			1	1	1
3		1			1	1		1
4				1		1		1
5	1			1				1
6	1	1	1					
7	1		1	1			1	1
8		1			1		1	
9	1		1	1				

Рис. 2. Пример выделения и сопоставления однородных групп

В примере выбрана группа $E^3(1)$, что приводит к нестабильности группы $E^3(3)$. В последующей итерации дифференцируется оставшаяся группа $E^4(4)$ типа (H). Для вершин группы $E^3(2)$ на рис. 2 приведены совокупности $Z^3(G)$ и $Z^3(H)$, из которых следует, что в графах G и H соответствующие вершины по-разному (неизоморфно) связаны с вершинами окружения. Вершины с уникальными описателями $\{d^3\}=(1,2,...,7)$ в графах G и H соответствуют изоморфным подграфам.

На рис. 3 показано дерево, отражающее последовательность выбора и дифференциации вершин однородных групп в графах G и H , приведённых на рис. 2. Жирными линиями выделен маршрут, который соответствует последовательности дифференциации групп, представленной на рис. 2. Рядом с вершинами дерева указаны соответствующие группы и обозначены их типы. Символом * помечены стабильные группы, для дифференциации которых вводились абстрактные описатели. В дереве группа типа (HЭ) показывается один раз на шаге её выделения. Это обусловлено наличием у групп данного типа очень важного свойства – для групп типа (HЭ) в ходе дифференциации групп её окружения переход в группу типа (H) или (C) невозможен. Для дерева на рис. 3 это означает, что однородная группа на любом k -м шаге, оказавшись неэквивалентной, сохраняет этот тип во всех маршрутах дерева, проходящих через данную группу.

По результатам работы алгоритма сходство графов G и H можно оценить в виде доли вершин, вошедших в изоморфный подграф. В данном примере эта доля составила $2/3$. Неизоморфная часть в примере содержит 3 вершины, которые в данном случае также можно рассматривать как изоморфный подграф с оценкой $1/3$.

Алгоритм выделения в графах общей части в виде частичных графов

Если данный алгоритм рассматривать независимо от результатов, полученных при выполнении первого алгоритма, то он сводится к поиску такого варианта попарного совмещения вершин из графов G и H , при котором число рёбер между совмещёнными вершинами является максимальным. Для решения такой задачи нужно проанализировать все варианты совмещения вершин. Число вариантов определяется числом перестановок вершин в одном из графов, что приводит к недопустимо большим объёмам вычислений.

При использовании результатов первого алгоритма задача сводится к расширению числа вершин и рёбер общей части (изоморфного подграфа) за счёт вершин и рёбер неизоморфной части. Алгоритм использует совмещение вершин, полученное при выделении изоморфного подграфа, и выбирает вариант совмещения вершин в неизоморфной части, доставляющий наибольшее число рёбер в расширенной общей части, т. е. в частичном графе. Если неизоморфная часть представлена несколькими однородными группами типа (HЭ), то для каждой из них алгоритм выполняется автономно.

Ниже предлагаются основные операции алгоритма решения данной задачи.

1. В графах G и H рассмотрению подлежат однородные группы типа (HЭ) $E_G^k(d^k)$ и $E_H^k(d^k)$ с одинаковыми значениями d^k . Для каждой пары совмещаемых вершин (e_i, e_j) , $e_i \in E_G^k(d^k)$, $e_j \in E_H^k(d^k)$ определяется оценка c_{ij} сходства отношений этих вершин с вершинами окружения, $c_{ij} = |Z_i^k(G) \cap Z_j^k(H)|$. Заметим, что в совокупности $Z_i^k(G)$ и $Z_j^k(H)$ могут входить описатели d^k рассматриваемых групп $E_G^k(d^k)$ и $E_H^k(d^k)$ и других групп типа (HЭ) неизоморфных частей. Величины c_{ij} формируются

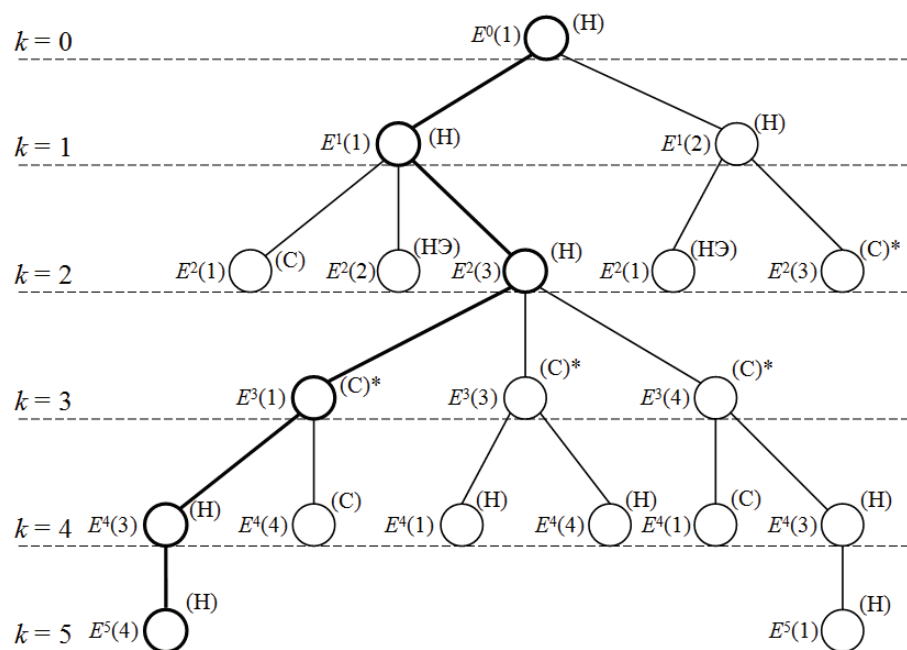


Рис. 3. Дерево выбора и дифференциации вершин однородных групп

- в матрицу $C=\|c_{ij}\|$. Значения элементов c_{ij} оценивают предпочтение совмещения вершин e_i и e_j . Для примера на рис. 2 совмещение (e_4, e_3) с соответствующими $Z_4^3(G)=(5,2,3,6)$ и $Z_3^4(H)=(3,5,2,1)$ даёт оценку $c_{4,3}=3$, а (e_5, e_7) — оценку $c_{5,7}=2$.
- Относительно матрицы C с помощью одного из известных алгоритмов решается задача назначения [6]. В результате решения данной задачи выбирается вариант совмещения вершин с максимальной суммой оценок предпочтения. В примере получены следующие элементы матрицы C : $c_{2,1}=3$, $c_{2,3}=3$, $c_{2,7}=2$, $c_{4,1}=2$, $c_{4,3}=3$, $c_{4,7}=2$, $c_{5,1}=3$, $c_{5,3}=2$, $c_{5,7}=2$. Одно из решений с суммарной оценкой равной 8 соответствует варианту совмещения $c_{2,7}=2$, $c_{4,3}=3$, $c_{5,1}=3$.
 - Для варианта совмещения, полученного во 2-й операции, строится частичный граф, который представляет общую часть графов G и H . С этой целью формируется полный вариант совмещения вершин графов G и H , включающий совмещения вершин согласно изоморфным подграфам и совмещения, полученные во 2-й операции. Совмещаемые пары вершин (e_i, e_j) принимаются в качестве вершин формируемого частичного графа. Вершины частичного графа (e_i, e_j) и (e_q, e_s) связываются ребром, если в графе G есть ребро r_{iq} , а в графе H — ребро r_{js} .

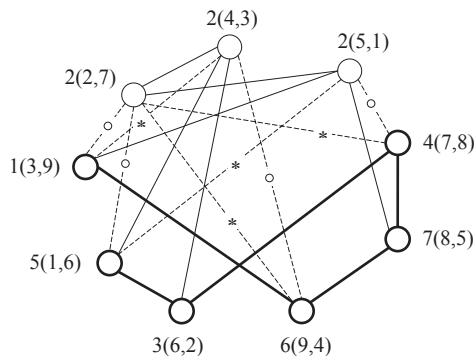


Рис. 4. Общие части графов G и H в виде подграфа и частичного графа

Частичный граф, полученный с помощью 1-го и 2-го алгоритмов для рассматриваемого примера сравнения графов G и H , представлен на рис. 4. Обозначение вершин частичного графа приведено в виде $d^k(e_i, e_j)$. Здесь d^k — описатель вершин совмещения (e_i, e_j) , вершина e_i из графа G , а e_j из графа H . Вершины и ребра подграфа, полученного в первом алгоритме, выделены жирными линиями. Сплошными тонкими линиями показаны вершины и ребра, включённые

в частичный граф во втором алгоритме. Пунктирные линии, помеченные знаком $*$, соответствуют рёбрам графа G , не вошедшим в общую часть. Аналогичные рёбра графа H помечены знаком $*$.

Граф на рис. 4 представляет результат наложения графов G и H в соответствии с полученным вариантом совмещения вершин. При этом 11 рёбер из 15 совместились и вошли в общую часть. Таким образом, сходство графов G и H по их общей части в виде частичного графа можно оценить отношением $11/15$. Данная оценка сходства получилась несколько выше, чем оценка равная $2/3$, полученная на основе изоморфных подграфов. Здесь нельзя говорить о предпочтениях второй оценки над первой или наоборот. При анализе сходства структур в различных предметных областях каждая из этих оценок может интерпретировать определённый содержательный смысл.

Заключение

Задача определения оценки сходства структур двух графов относится к классу неполиномиальной сложности. Поэтому сам факт разработки приемлемого для применения на практике алгоритма её решения на основе выделения у сравниваемых графов общих частей является важным результатом выполненных исследований. В числе достоинств отметим также, что алгоритм решения данной задачи эффективно работает на графах, содержащих десятки вершин, и по объёму вычислений сопоставим с задачей интеграции структурных различий.

Что касается недостатков, то их существенно больше. Основной недостаток заключается в том, что алгоритм может сравнивать только такие графы, у которых векторы начальных описателей совпадают. Это означает, что алгоритм может работать в узком классе графов, определяемом вектором начальных описателей. Алгоритм не гарантирует выделение наибольшего изоморфного подграфа и последующее получение наибольшего частичного графа. В качестве значимого недостатка отметим также следующее. Принимая группу типа (НЭ) в качестве «убежища» для структурных различий, алгоритм не проверяет возможность сокращения размерности этого «убежища». Такие проверки особенно важны для групп, содержащих относительно большое число вершин. Несмотря на указанные недостатки, алгоритм и в предложенном виде является эффективным инструментом анализа сходства структур, относящихся к одному классу, задаваемому вектором начальных описателей.

Работа выполнена в рамках госзадания «Наука».

СПИСОК ЛИТЕРАТУРЫ

- Погребной В.К. Об одном методе определения изоморфизма графов // Кибернетика. — 1982. — № 2. — С. 7–13.
- Погребной В.К. Алгоритм решения задачи определения изоморфизма гиперграфов // Известия Томского политехнического университета. — 2010. — Т. 317. — № 5. — С. 16–21.
- Погребной В.К. Метод интеграции структурных различий в графовых моделях и его применение для описания структур // Известия Томского политехнического университета. — 2011. — Т. 318. — № 5. — С. 10–16.
- Зыков А.А. Основы теории графов. — М.: Изд-во «КомКнига», 2004. — 644 с.
- Погребной В.К. Решение задачи определения изоморфизма графов, представленных атрибутными матрицами // Известия Томского политехнического университета. — 2012. — Т. 321. — № 5. — С. 52–56.
- Гольштейн Е.Г., Юдин Д.Б. Задачи линейного программирования транспортного типа. — М.: Наука, ФИЗМАТЛИТ, 1969. — 384 с.

Поступила 14.01.2013 г.

УДК 519.175.1

ИНВАРИАНТ ГРАФА НА ОСНОВЕ КОМПАКТНЫХ ПОДГРАФОВ И АЛГОРИТМ ЕГО ВЫЧИСЛЕНИЯ

Ан.В. Погребной, В.К. Погребной

Томский политехнический университет

E-mail: avpogrebnoy@gmail.com

Для взвешенных обыкновенных графов введена оценка компактности подграфов. На основе этой оценки определён ряд инвариантов, характеризующих структуру графа с учётом неравномерности распределения значений весов по рёбрам. Основное внимание уделено понятию компактного подграфа. Предложен алгоритм выделения компактных подграфов и вычисления на их основе инвариантов. Подмечено важное свойство компактных подграфов – способность отражать эффект обособления подмножеств вершин с высокой оценкой компактности.

Ключевые слова:

Инвариант графа, взвешенный обыкновенный граф, оценка компактности подграфа, компактный подграф, эффект обособления вершин.

Key words:

Graph invariant, weighted ordinary graph, estimation of subgraph density, dense subgraph, vertex isolation effect.

Введение

Для описания и анализа структуры графовых моделей в теории графов накоплено большое разнообразие количественных показателей (дескрипторов), каждый из которых характеризует некоторое структурное свойство графа [1]. Например, хроматическое число графа, число внутренней устойчивости, вектор упорядоченных значений степеней вершин [2]. Дескрипторы (описатели), которые определяются независимо от нумерации вершин или какого-либо их наименования, т. е. являются инвариантами относительно обозначения вершин, стали называться инвариантами [2]. В частности, матрица смежности вершин графа не является инвариантом, т. к. зависит от нумерации вершин (строк и столбцов матрицы). Напротив, длина кратчайшего маршрута в графе является инвариантом, а непосредственно сам маршрут в виде последовательности вершин не является.

В общем случае можно говорить, что каждый инвариант представляет некоторую характеристику структуры абстрактного графа, т. е. графа, у которого отсутствует обозначение вершин. Эти характеристики интерпретируют различные свойства графовых моделей, разработанных для конкретных приложений.

Инварианту соответствует количественная мера в виде числа или вектора. Поэтому инварианты одного типа, полученные для разных графов, могут сравниваться между собой, определяя степень сходства или различия этих графов. В изоморфных графах инварианты совпадают, т. е. равенство графов является необходимым условием их изоморфизма. К сожалению, обратное утверждение не верно, т. к. равенство инвариантов у двух графов вовсе не гарантирует наличие у них изоморфизма. Объёмы вычислений, связанные с определением инвариантов, зависят от их типов. Некоторые из них, например степени вершин, определяются совсем просто, а такие как хроматическое число графа требуют решения отдельной сложной задачи.

Во многих практических приложениях графовые модели представляются в виде взвешенных обыкновенных графов. В таких моделях рёбрам графа приписываются некоторые значения (веса), отражающие отдельное свойство соответствующей коммуникации, например протяженность, пропускную способность и т. п. Для анализа таких графовых моделей важно иметь инвариант, который характеризует структуру графа с учётом весов рёбер.

В статье вводится инвариант данного типа и предлагается алгоритм его вычисления. Для разработки инварианта вводится новое понятие – *компактность подграфа* и даётся её количественная оценка. Предлагаемый инвариант даёт возможность связать неравномерность распределения значений весов по рёбрам графа с его структурой и оказывается полезным при решении ряда задач анализа структур графов.

Компактность подграфов и инварианты

Рассматривается связный обыкновенный граф $G=(E, U)$ с множеством вершин $E=\{e_i\}$, $i=1, 2, \dots, n$ и множеством рёбер $U=\{u_{ij}\}$. Каждому ребру $u_{ij}=(e_i, e_j)$, связывающему две вершины e_i и e_j , ставится в соответствие вес $r_{ij}>0$. В этом случае граф G может быть представлен матрицей $R=\|r_{ij}\|$, которая совмещает матрицу смежности вершин и матрицу весов рёбер. Граф G может быть также представлен множеством инциденторов $F(e_i)$ вершин $e_i \in E$. Инцидентор $F(e_i)$ представляет совокупность вершин $e_j \in E$, $i \neq j$, которые в графе G связаны рёбрами $u_{ij}=(e_i, e_j)$, инцидентными вершине e_i . Мощностью множества $F(e_i)$ в этом случае определяет степень s_i вершины e_i , т. е. $s_i=|F(e_i)|$.

Если значение веса r_{ij} ребра (e_i, e_j) ассоциировать с силой притяжения (сжатия) вершин e_i и e_j между собой, то можно говорить о компактности совокупности вершин e_i и e_j (от лат. *compactus* – «сжатый»). При этом будем считать, что чем больше значение веса r_{ij} , тем выше значение компактности.

Таким образом, в качестве количественной меры компактности подграфа, содержащего вершины e_i и e_j , может быть принято значение веса r_{ij} . Аналогичные рассуждения можно распространить на подграфы, содержащие больше двух вершин. В этом случае оценка компактности $C(E_i)$ подграфа $G_i=(E_i, U_i)$ с множеством вершин $E_i \subset E$ и множеством рёбер $U_i \subset U$ определяется в виде суммы значений весов рёбер $u_{ij} \in U_i$, связывающих вершины $e_i \in E_i$,

$$C(E_i) = \sum_{u_{ij} \in U_i} r_{ij}.$$

На основе оценок компактности подграфов можно предложить ряд инвариантов:

- 1) сумма весов рёбер графа (компактность графа);
- 2) вектор оценок компактности кратчайших маршрутов в графе;
- 3) вектор упорядоченных значений оценок компактности связных подграфов, содержащих заданное число вершин;
- 4) вектор упорядоченных значений оценок компактности подграфов, построенных для подмножеств вершин, выделенных по определённым правилам.

Инварианты первой группы вычисляются на основе оценок компактности графа в целом. Сюда можно отнести инвариант $C(E)$, равный сумме весов рёбер графа, или инвариант $C(U)=C(E)/|U|$, равный среднему значению весов рёбер. В качестве инварианта этой группы можно рассматривать также гистограмму частот значений весов рёбер в графе.

Во вторую группу отнесены инварианты, характеризующие компактность кратчайших маршрутов между всеми парами вершин или компактность подграфов, содержащих вершины этих маршрутов. В первом случае для каждой пары вершин вычисляется сумма весов рёбер кратчайшего маршрута. Значения этих сумм в векторе инварианта упорядочиваются по возрастанию или убыванию. Во втором случае элемент в векторе инварианта вычисляется как оценка компактности подграфа, содержащего вершины кратчайшего маршрута. Эта оценка суммирует веса всех рёбер, связывающих вершины маршрута в графе.

Векторы инвариантов третьей группы $S^g(E^g)=\{C^g_i(E^g_i)\}$ формируются на основе оценок компактности $C^g_i(E^g_i)$ связных подграфов, содержащих E^g_i вершин. Здесь $t=1,2,\dots,t_g$ обозначает порядковый номер подграфа, оценка компактности которого заняла t -е место в упорядоченной по возрастанию последовательности элементов вектора $S^g(E^g)$, t_g — общее число связных подграфов, содержащих g вершин.

В отличие от третьей группы векторы инвариантов четвертой группы $C(E_i)$ включают оценки компактности подграфов $G_i=(E_i, U_i)$, выделенных на определённых условиях. Например, в качестве G_i могут выступать полные подграфы графа G . В качестве множеств вершин E_i для выделения подграфов G_i могут быть приняты инциденторы $F(e_i)$ или $F(e_i) \cup e_i$.

В составе данной группы особый интерес представляют инварианты, основанные на так называемых *компактных подграфах*. Введению понятия компактного подграфа, алгоритму их выделения в графе и инварианту, полученному на основе этого понятия, посвящено последующее содержание данной статьи.

Компактным подграфом графа G будем называть связный подграф $G^g_i=(E^g_i, U^g_i)$ $i=1,2,\dots,n$ с множеством вершин E^g_i , $|E^g_i|=g$ и множеством рёбер U^g_i , связывающих вершины из множества E^g_i , который в множестве E^g_i содержит вершину e_i и имеет наибольшую оценку компактности C^g_i .

Для выделения компактного подграфа G^g_i в графе G нужно сформировать все связные подграфы с множеством вершин E^g_i , $e_i \in E^g_i$, $|E^g_i|=g$, получить для каждого из них оценку компактности C^g_i и выбрать подграф, соответствующий наибольшей оценке.

Инвариантом графа G на основе компактных подграфов G^g_i будем называть n -мерный вектор S^g с упорядоченными по возрастанию оценками компактности C^g_i подграфов G^g_i .

Алгоритм выделения связных подграфов

Задача выделения компактных подграфов G^g_i может быть решена путём простого перебора всех сочетаний вершин из n по g . Для каждого сочетания анализируется соответствующий подграф, и если он является связным, то для него определяется оценка компактности, а несвязные подграфы исключаются из рассмотрения. Далее в полученном множестве подграфов для каждой вершины e_i выбирается компактный подграф G^g_i , который имеет наибольшую оценку компактности и содержит вершину e_i .

Очевидным недостатком такого алгоритма является необходимость полного перебора всех сочетаний вершин и соответствующих подграфов. При анализе графа G , содержащего несколько десятков вершин, число сочетаний становится большим уже при $g \geq 4$. Поэтому важно, чтобы алгоритм мог формировать сочетания вершин, которые соответствуют только связным подграфам.

Второе требование к алгоритму обусловлено тем, что при анализе графа G как правило требуется последовательно выделять компактные подграфы для $g=2,3,4,\dots$ и формировать соответствующие инварианты S^g . В этом случае при генерации множества сочетаний для очередного значения $(g+1)$ алгоритм не должен выполнять генерацию заново, а всего лишь продолжить её, используя множество связных подграфов, содержащих g вершин. Предлагаемый ниже алгоритм разработан с учётом данных требований.

Множество связных подграфов алгоритм формирует в виде дерева. На уровне $g=1$ в дереве располагаются вершины e_i множества E , упорядоченные по возрастанию номеров. Вершины этого множества представим как условно связанные с нулевой вершиной ($e^g_{i=0}$) на уровне $g=0$ и обозначим его в виде $Z(e^g_0)$. Вершина $e^{g-1}_i \in Z(e^g_0)$ выступает в каче-

стве корневой вершины автономного дерева, ветви которого соответствуют всем связным подграфам, содержащим корневую вершину. Построение такого дерева заключается в формировании множеств $Z(e_i^{g+1})$, вершины которых в дереве связываются с вершинами e_i^g на предыдущем g -м уровне.

Совокупность вершин множества $Z(e_i^{g-1})$ представим в виде упорядоченной по возрастанию номеров последовательности $(e_{i_1}^g, e_{i_2}^g, \dots, e_{i_z}^g)$. Будем считать, что вершина $e_{i_k}^g \in Z(e_i^{g-1})$, для которой формируется множество $Z(e_{i_k}^g)$ вершин $e_{i_k}^{g+1}$, связанных в дереве с вершиной $e_{i_k}^g$, разбивает множество $Z(e_i^{g-1})$ на две части $Z^-(e_{i_k}^g) = (e_{i_1}^g, e_{i_2}^g, \dots, e_{i_k}^g)$ и $Z^+(e_{i_k}^g) = (e_{i_{k+1}}^g, \dots, e_{i_z}^g)$, z – число вершин в множестве $Z(e_i^{g-1})$.

В принятых обозначениях алгоритм формирует множества $Z(e_i^g)$ по следующим правилам:

$$Z(e_0^0) = E = \{e_i^1\}, i = 1, 2, \dots, n; \quad (1)$$

$$Z(e_k^1) = F(e_k^1) \setminus Z^0(e_0^0). \quad (2)$$

В общем случае для вершин e_i^g дерева на уровне $g \geq 2$ множества $Z(e_i^g)$ определяются по выражению:

$$Z(e_i^g) = (F(e_i^g) \cup (Z^1(e_i^{g-1}) \setminus \bigcup_{g=0}^g Z^0(e_i^{g-1}))). \quad (3)$$

Так, например, для вершины e_2^2 на уровне $g=2$ имеем:

$$Z(e_2^2) = (F(e_2^2) \cup (Z^1(e_2^1) \setminus (Z^0(e_0^0) \cup Z^0(e_1^1))). \quad (4)$$

Выражения (1), (2) определяют подграфы, содержащие 1 и 2 вершины соответственно. Все последующие подграфы, содержащие $g+1$ вершину, определяются рекуррентно по выражению (3). В дереве на уровне $g+1$ каждой вершине e_i^{g+1} соответствует подграф, содержащий все вершины ветви от e_i^{g+1} до корневой вершины. Число подграфов на уровне $g+1$ определяется суммарным числом вершин всех множеств $Z(e_i^g)$, сформированных для вершин e_i^g , расположенных на уровне g дерева.

Работу алгоритма покажем на примере графа G и его матрицы R , представленных на рис. 1.

При построении дерева нет необходимости учитывать веса. Поэтому вес $r_{ij} > 0$ рассматривается в матрице R как факт наличия ребра u_{ij} , связывающего вершины e_i и e_j в графе G . Будем также считать, что при построении дерева переход на уровень g осуществляется после формирования всех подграфов на предыдущем уровне. На рис. 2 при-

ведено дерево, построенное по данному алгоритму для $g=1, 2, 3$.

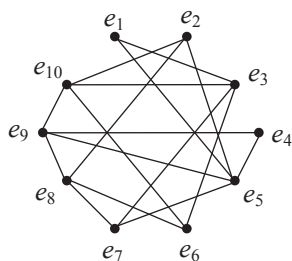
Дерево содержит 17 подграфов по 2 вершины на уровне $g=2$ и 41 подграф по 3 вершины на уровне $g=3$. Подграфы на уровне $g=2$ соответствуют всем рёбрам графа G . На уровнях $g=2$ и $g=3$ у каждой вершины дерева поставлены 2 числа – верхнее обозначает номер вершины e_i в графе G , а нижнее определяет компактность подграфа для соответствующей ветви дерева. Например, ветвь, выделенная жирно, соответствует подграфу с вершинами (e_2, e_8, e_1) и компактностью равной 10. Заметим, что наличие ребра в дереве не означает наличие соответствующего ребра в графе. Например, ветвь (e_2, e_8, e_{10}) содержит ребро (e_8, e_{10}) , а в графе такого ребра нет. Здесь важно, что рёбра ветви (e_2, e_8, e_{10}) отражают связность подграфа.

После построения согласно выражениям (1) и (2) множеств 1-го и 2-го уровня дерева формируются множества $Z(e_k^2)$ для всех вершин e_k^2 2-го уровня. Например, чтобы сформировать множество $Z(e_8^2)$ предва- рительно по ребру (e_2^1, e_8^1) устанавливается множество $Z(e_2^1) = (e_5, e_8, e_{10})$ к которому относится вершина e_8^2 . Далее это множество, как показано на рис. 2, разбивается на $Z^-(e_2^1) = (e_5, e_8)$ и $Z^+(e_2^1) = (e_{10})$. Учитывая, что инцидентор $F(e_8^2) = (e_2, e_6, e_7, e_9)$, согласно (3) или в данном случае (4), имеем:

$$\begin{aligned} Z(e_8^2) &= ((e_2, e_6, e_7, e_9) \cup (e_{10})) \setminus ((e_1, e_2) \cup (e_5, e_8)) = \\ &= (e_6, e_7, e_9, e_{10}). \end{aligned}$$

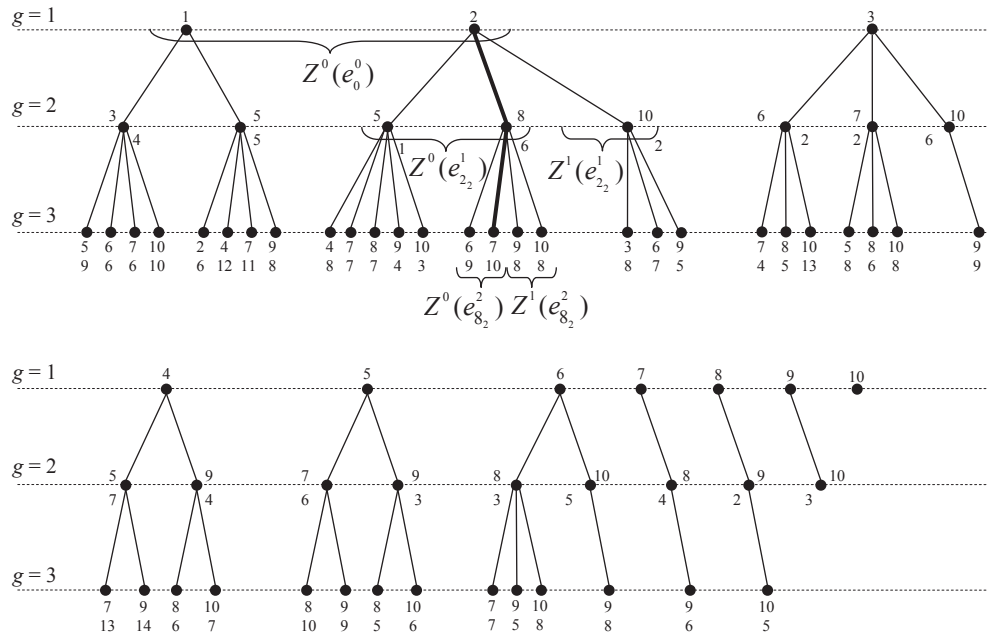
Построенное таким образом дерево удовлетворяет сформулированным ранее требованиям и содержит все связные подграфы, содержащие 3 вершины. При построении дерева алгоритм допускает формирование множеств для вершин e_i^g в любой последовательности. В частности, можно без построения всего дерева автономно сформировать множество для вершины e_{12}^3 , расположенной на ветви (e_2, e_8, e_1) . С этой целью нужно последовательно сформировать множества $Z(e_0^1)$, $Z(e_2^1)$, $Z(e_8^2)$, $Z(e_{12}^3)$. В нашем примере $Z(e_{12}^3)$ в соответствии с (3) определяется следующим образом:

$$\begin{aligned} Z(e_{12}^3) &= \\ &= ((e_3, e_5, e_8) \cup (e_9, e_{10})) / ((e_1, e_2) \cup (e_5, e_8) \cup (e_6, e_7)) = \\ &= (e_3, e_9, e_{10}). \end{aligned}$$



	1	2	3	4	5	6	7	8	9	10
1				4	5					
2					1			6		2
3	4					2	2			6
4					7				4	
5	5	1		7			6		3	
6			2					3		5
7					6			4		
8		6				3	4		2	
9				4	3			2		3
10		2	6			5			3	

Рис. 1. Пример графа G и его матрицы R


 Рис. 2. Дерево связанных подграфов для $g=1,2,3$

Заметим также, что упорядочение вершин в множествах $Z(e_i^g)$ по возрастанию номеров не является обязательным и введено в алгоритм всего лишь для удобства восприятия дерева.

Инварианты на основе компактных подграфов

При наличии дерева связанных подграфов G_i^g с оценками компактности G_i^g выделение компактных подграфов G_i^g осуществляется последовательно для всех вершин $e_i \in E$. С этой целью среди подграфов G_i^g , содержащих вершину e_i , выбирается подграф G_i^g с максимальной оценкой компактности C_i^g . Если таких подграфов окажется несколько, то они все запоминаются. Выбор подграфов в этих случаях производится после выбора компактных подграфов для всех вершин e_i . При этом предпочтение отдаётся тому подграфу, который максимально пересекается с подграфами, выбранными для других вершин.

Результаты выбора компактных подграфов G_i^3 для рассматриваемого примера сведены в таблицу.

Столбец таблицы представляет компактный подграф G_i^3 для вершины e_i . Вершины в множестве E_i^3 графа G_i^3 указываются в последовательности, аналогичной соответствующей ветви дерева. Для вершины e_8 в дереве оказалось два компактных подграфа с оценкой $C_8^3=10$. Предпочтение в данном случае отдаётся подграфу с множеством вершин $E_8^3=(e_2, e_8, e_7)$, которое полностью пересекается с E_2^3 .

Таблица. Компактные подграфы

e_i	1	2	3	4	5	6	7	8		9	10
E_i^3	1	2	3	4	4	3	4	2	5	4	3
	5	8	6	5	5	6	5	8	7	5	6
	4	7	10	9	9	10	7	7	8	9	10
C_i^3	12	10	13	14	14	13	13	10		14	13

В нижней строке таблицы указаны значения оценок компактности. Данные оценки в упорядоченной по возрастанию последовательности образуют инвариант в виде n -мерного вектора $C^3(G)$. В нашем примере $C^3(G)=(10,10,12,13,13,13,13,14,14,14)$. Нетрудно убедиться, что инвариант на основе компактных подграфов G_i^3 для дерева на рис. 2 соответствует вектору $C^2(G)=(4,5,5,6,6,6,6,7,7)$.

Компактные подграфы G_i^g , наряду с тем, что являются основанием для формирования инвариантов $C^g(G)$, обладают примечательным свойством, которое заключается в способности отражать степень обособления вершин в графе. Эффект обособления поясним на примере компактных подграфов G_i^3 , представленных в таблице. Для этого на рис. 3 визуальнo покажем принадлежность вершин графа G компактным подграфам G_i^3 .

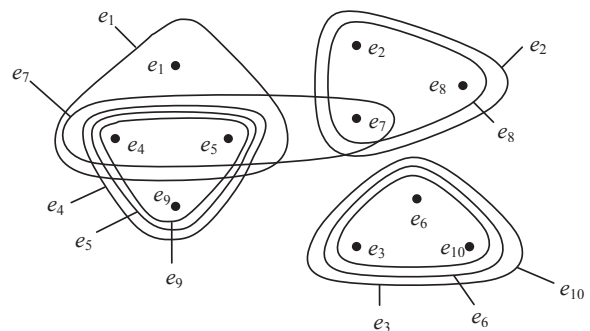


Рис. 3. Иллюстрация эффекта обособления вершин

Множества вершин E_i^3 на рис. 3 обведены линиями, помеченными вершинами e_i , для которых получены соответствующие компактные подграфы G_i^3 .

Эффект обособления вершин легко просматривается визуально. В частности, если исключить подграф G_7^3 , то вершины графа G разобьются на 3 обособленные подмножества с высокой оценкой компактности. Очевидно, что эффект обособления вершин взвешенных графов G на основе компактных подграфов G_i^* может быть использован при решении задач декомпозиции графа на подграфы. Применение компактных подграфов для решения таких задач требует отдельного исследования и в данной статье не рассматривается.

Выводы

1. Введение оценки компактности для подграфов взвешенного обыкновенного графа даёт возможность сформулировать ряд инвариантов, которые оказываются полезными при анализе структур графов с учётом весов рёбер.

СПИСОК ЛИТЕРАТУРЫ

1. Дэмер М., Эммерт-Штрайб Ф., Цой Ю.Р., Вармуза К. Новый функционал информативности для анализа структуры хими-

2. Предложенный алгоритм формирования связанных подграфов позволяет избежать полого перебора подграфов, исключая несвязные и связанные дублирующие подграфы. Алгоритм может быть полезен в любых других исследованиях, где возникает потребность в полном или частичном выделении связанных подграфов.
3. Введение понятия компактного подграфа и определение на этой основе инварианта в виде n -мерного вектора расширило возможность по увязке неравномерности распределения значений весов по рёбрам графа с его структурой. Кроме того, компактные подграфы выступают в роли «усилителя» эффекта обособления подмножеств вершин с высокой оценкой компактности, что может оказаться весьма полезным при решении задач декомпозиции графов.

Работа выполнена в рамках госзадания «Наука».

ческих графов // Известия Томского политехнического университета. – 2010. – Т. 316. – № 5. – С. 5–11.

2. Зыков А.А. Основы теории графов. – М.: Изд-во «КомКнига», 2004. – 644 с.

УДК 681.3.06(076.5)

ПРОГРАММНО-ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ АНАЛИЗА ПАРАМЕТРОВ СВЧ ИМПУЛЬСОВ

В.С. Косицын, В.Г. Гальченко, Т.А. Гладкова

Томский политехнический университет

E-mail: gal@am.tpu.ru; am@am.tpu.ru

Оценка параметров СВЧ импульсов является важной задачей при проведении экспериментальных исследований. Существующие программы построения графиков СВЧ импульсов и оценка их параметров реализованы в среде MathCad и обладают существенными недостатками, так как многие параметры среды, по которой проходят СВЧ импульсы, требуется вводить непосредственно в программу. В работе описано программное обеспечение оценки параметров СВЧ импульсов, реализованное в среде Qt 4.5 C++.

Ключевые слова:

СВЧ импульс, система программирования, параметры импульса, программное обеспечение.

Key words:

Microwave pulse, programming system, pulse parameters, software.

Введение

В работе [1] описана схема измерения параметров СВЧ импульса. Детектированные сигналы с трёх измерительных каналов поступают на цифровой осциллограф и записываются в виде текстовых файлов. Для приёмных детекторов имеются предварительно полученные с помощью генератора импульсов (ГИН) калибровочные кривые. Кривые показывают зависимость выходного напряжения датчиков от СВЧ мощности на входе измерительного канала при различных частотах СВЧ сигнала. Обработка экспериментальных данных, полученных с помощью цифрового осциллографа, проводилась с использованием программного обеспечения, созданного в среде MathCad. Текстовые файлы считывались программой MathCad для получения графиков различных параметров СВЧ импульса. Для экспериментатора программное обеспечение в среде MathCad является достаточно неудобным, так как требует ввода множества исходных данных непосредственно в программу.

В работе для обработки экспериментальных данных, записанных с помощью цифровых осциллографов, представлено программное обеспечение в среде Qt4.5 C++. В отличие от программы в MathCad, представленной в работе [1], все необходимые исходные данные вводятся из файла входных параметров, которые пользователь может изменить в диалоговом окне.

Измерения параметров СВЧ импульса

Схема эксперимента представлена на рис. 1. В соответствии с этой схемой СВЧ импульс поступает на антенну и далее через согласованный тройник распространяется по двум измерительным каналам. В одном из каналов, называемом «линейным», после фиксированного аттенюатора установлена коаксиальная детекторная головка с полупроводниковым СВЧ диодом. СВЧ диод в головке осуществляет детектирование СВЧ импульса. Во втором канале – «нелинейном», между фиксированным аттенюатором и коаксиальной детектор-

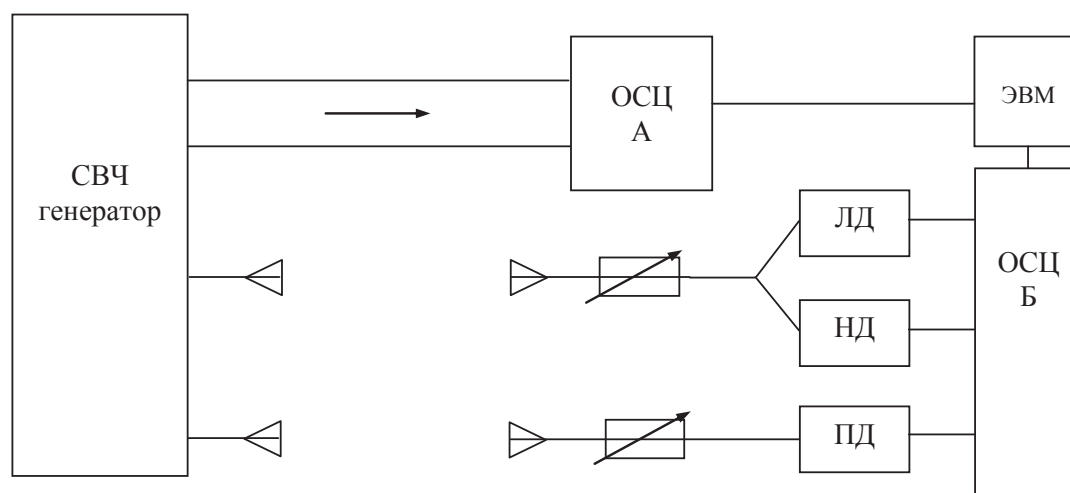


Рис. 1. Схема эксперимента: ОСЦ А, Б – осциллографы А, Б; ЛД – линейный детектор; НД – нелинейный детектор; ПД – подвижный детектор

ной головкой, которая подобна головке в линейном канале, установлен «запредельный» аттенуатор, величина ослабления которого изменяется от частоты по определенному закону. Кроме этого, СВЧ импульс принимается антенной пространственно-подвижного линейного детектора, что позволяет построить диаграмму направленности СВЧ-генератора и определить общую излучаемую мощность.

Регистрация параметров СВЧ импульса осуществляется линейным, нелинейным и подвижным детекторами измерительных каналов, которые подключаются к осциллографу Б. Частотные зависимости измеренной мощности линейного и нелинейного каналов для выходных напряжений датчиков и приведенная импульсная мощность находятся с использованием калибровочных характеристик линейного и нелинейного каналов.

Структура программного обеспечения

В процессе эксперимента данные подаются на два цифровых осциллографа – А и Б, и записываются в виде текстовых файлов. На один осциллограф подаются сигналы с датчиков, регистрирующих параметры самого СВЧ-генератора, что позволяет оценить динамику его работы и осуществить временную привязку СВЧ импульсов с детекторов, которые поступают на вход второго осциллографа. Имя каждого текстового файла с осциллографов является составным и содержит букву с номером канала, далее через точку пять цифр номера эксперимента и через точку расширение txt или dat. Наличие номера эксперимента в имени файла позволяет выбрать файлы, соответствующие нужному эксперименту.

Входные данные, необходимые для обработки экспериментальных данных, находятся в файле входных данных, который считывается при запуске программы. Эти данные размещаются в таблице данных. После считывания экспериментальных данных из текстовых файлов, записанных с помощью двух цифровых осциллографов А и Б, создаются формы изображения малых графиков, на которые добавляются экспериментальные данные сигналов с датчиков.

Для расчета параметров СВЧ импульсов по полученным сигналам с датчиков используются файлы, которые формируются в процессе калибровки датчиков. При этом используется необходимый набор частот и соответствующие уровни мощности СВЧ-генератора импульсов. После преобразования сигналов с датчиков с учетом калибровочных параметров осуществляется расчет временной задержки каждого сигнала с учетом среды прохождения сигнала и длины кабеля, по которому передается сигнал. Данные о задержке сигнала по каждому измерительному каналу представлены в таблице.

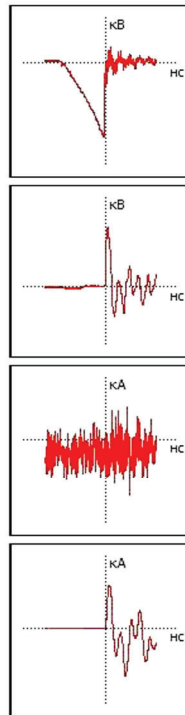
После преобразования сигналов с датчиков осуществляется создание форм графиков и добавление графиков на форму. На каждом графике имеется возможность подробно исследовать его от-

дельные участки. С этой целью используется временное отсечение графика с помощью задания начальной и конечной временной точки отсечения.

Таблица. Входные данные

Осциллограф А			
Канал 1			
Данные		Обозначение	Значение
Напряжение ГИН (кВ)		Um	Файл C1
Делитель 1А		Div_m	8,80E+03
Ослабитель 1А		DB5fix	50
Коэффициент		Kum=DB5fix*Div_m	4,40E+05
Канал 2			
Напряжение на диоде (кВ)		Un	Файл C2
Делитель 2А		Div_n	2,05E+03
Ослабитель 2А		D1fix	205
Коэффициент		Kun=D1fix*Div_n	4,20E+05
Канал 3			
Ток в диоде (кА)		Ish	Файл C3
Шунт		Rsh	3,00E-02
Ослабитель 3А		D3fix	440
Коэффициент		Klsh=D3fix/Rsh	1,47E+04
Канал 4			
Ток в диоде, пояс Роговского (2) (кА)		Irc	Файл C4
Коэффициент		RCoil	1,92E+02
Ослабитель 4А		DA6fix	50
Коэффициент		KIrc=DA6fix*RCoil*2	1,92E+04
Осциллограф Б			
Канал 1			
Данные		Обозначение	Значение
Ток в диоде, пояс Роговского (1) (кА)		Ikp	Файл C4
Коэффициент		RCoil	1,92E+02
Ослабитель 1А		DA6fix	50
Канал 2			
Напряжение линейного детектора (кВ)		U7	Файл C2
Канал 3			
Напряжение нелинейного детектора (кВ)		U8	Файл C3
Канал 4			
Напряжение подвижного детектора (кВ)		U1d	Файл C4
Суммарное запаздывание			
Номер канала ОСЦ	Обозначение	Вычисление	Значение
1 канал А	TSCns	TLCns	122
4 канал А	TSCrc	DLrc+TLCrc	131,373
3 канал А	TSCsh	DLsh+TLCsh	126,352
2 канал А	Hrc	TSCrc-TSCns	9,373
1 канал Б	Hrc_sn	TSCrc-TSCsh	5,021
4 канал Б	TSCdet	DLdet+TLCdet	183,797
2 канал Б	Hdet	TSCdet-TSCns+hAB	60,618
3 канал Б	Hkp	TSCkp-TSCns+hAB	8,194

Осциллограф А



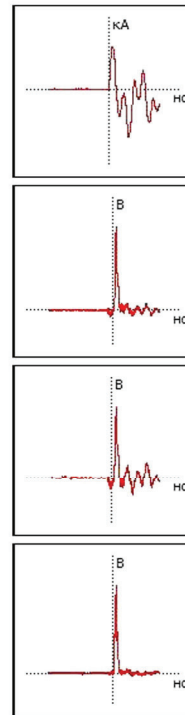
C1

C2

C3

C4

Осциллограф Б



C1

C2

C3

C4

Рис. 2. Графики сигналов с каналов осциллографов А и Б

Для количественной оценки параметров СВЧ импульса на отдельном графике используется красный маркер, перемещая который по графику в соответствующем окне можно получить количественное значение данного параметра. На графике предусмотрена возможность регистрации временной задержки между отдельными элементами. Для этого используется синий маркер, задавая положение которого в соответствующем окне, можно получить временную задержку между положением красного и синего маркера.

Экспериментальные графики

На рис. 2. представлены графики сигналов с датчиков, преобразованных в цифровую форму с помощью цифровых осциллографов А и Б. Экспериментальные данные имеют интервал дискретизации 2 нс.

В программе предусмотрена возможность получить увеличенное изображение графика, если дважды щелкнуть по нему левой кнопкой мыши.

При анализе экспериментальных данных важное значение имеют следующие параметры: форма импульса СВЧ на диод, форма импульса с линейного детектора, форма импульса СВЧ с подвижного детектора. Формы импульса данных параметров импульса СВЧ представлены на рис. 3, где показано расположение красного и синего маркера, с помощью которых оценивается значение параметров в месте нахождения красного курсора в соответствующем окне и запаздывание между максимумами импульсов выбранных параметров.

Сравнение напряжений с линейного, нелинейного и подвижного детектора представлено на рис. 4. Экспериментатор по форме импульса может оценить правильность взаимного расположения датчиков.

Мощность импульса относится к одному из важнейших параметров СВЧ-генератора, от которого зависит в целом качество самой установки, поэтому данный параметр необходимо оценивать при эксплуатации всей системы. На рис. 5 показана мощность импульса, которая измеряется линейным детектором.

Очень важно установить на какой частоте находится максимальная мощность СВЧ импульса. Оценка данного параметра показана на рис. 6.

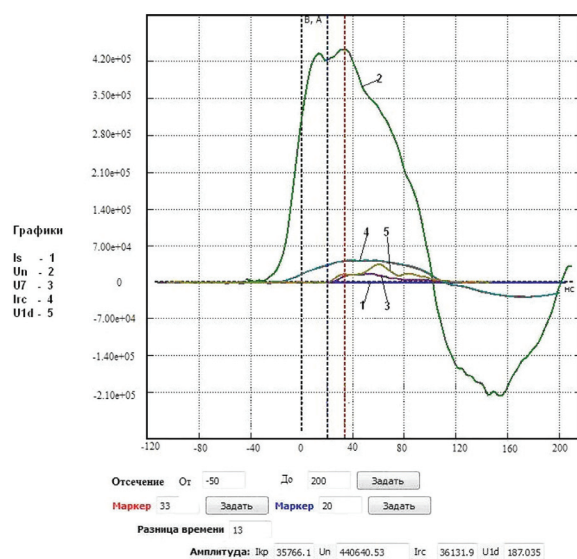


Рис. 3. 1) I_s – ток в диоде; 2) U_n – импульс СВЧ на диоде; 3) U_7 – импульс с линейного детектора; 4) I_{rc} – ток в диоде (пояс Роговского 2); 5) U_{1d} – импульс СВЧ с подвижного детектора

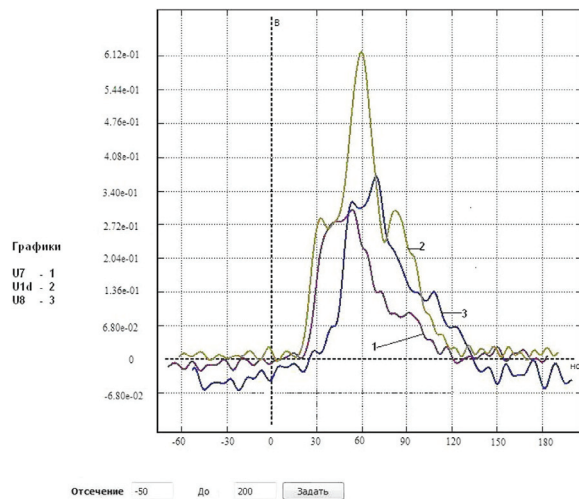


Рис. 4. Напряжения с датчиков: 1) U_7 – линейного; 2) U_{1d} – подвижного; 3) U_8 – нелинейного детекторов

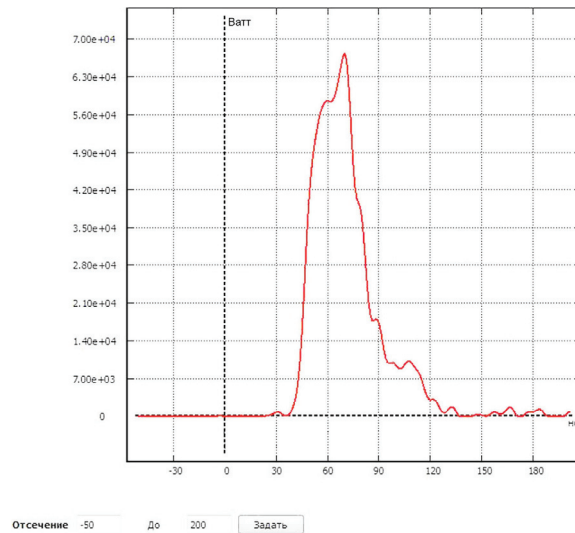


Рис. 5. Мощность импульса с линейного детектора

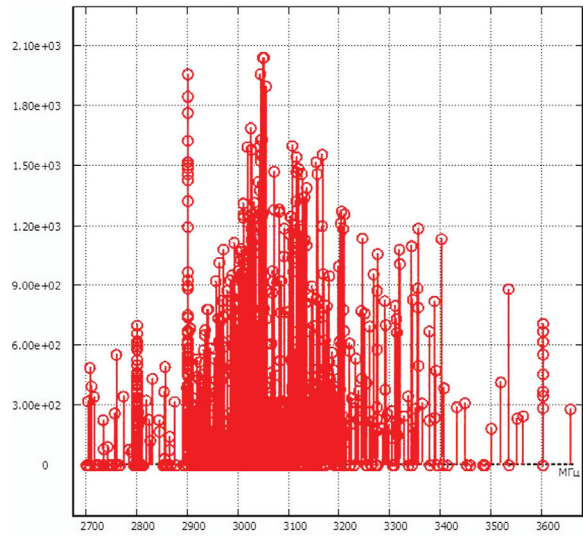


Рис. 6. Мощность импульса (Вт) по частотам с линейного детектора

Выводы

1. Разработанное программное обеспечение оценки параметров СВЧ импульсов в среде Qt4.5 C++ [2] имеет удобный интерфейс, доступный экспериментатору, содержит все необходимые исходные данные, которые легко могут быть исправлены при замене датчиков, а также изменения параметров среды прохождения СВЧ сигнала и применении различных типов генераторов мощных СВЧ-импульсов.
2. Экспериментатору не требуется редактировать программное обеспечение, достаточно только откорректировать параметры в диалоговом окне, имеющем вид таблицы.
3. Все исходные данные, находящиеся в таблицах, имеют пояснительный текст, что существенно упрощает процедуру внесения изменений.
4. Графический интерфейс позволяет в удобной форме представлять параметры СВЧ импульса, задавая необходимые временные отсечения.

СПИСОК ЛИТЕРАТУРЫ

1. Бабичев Д.А., Шиян В.П., Мельников Г.В. Измеритель частотного состава мощных СВЧ импульсов наносекундной длительности // Приборы и техника эксперимента. – 2003. – № 3. – С. 93–96.

2. Шлее М. Qt4.5 C++. Профессиональное программирование на C++. – СПб.: БХВ-Петербург, 2010. – 896 с.

Поступила 15.02.2013 г.

ОНТОЛОГИЧЕСКИЕ СИСТЕМЫ ПЕРЕХОДОВ И ИХ ПРИМЕНЕНИЕ К СЕМАНТИКЕ КОМПЬЮТЕРНЫХ ЯЗЫКОВ

И.С. Ануреев

Институт систем информатики имени А.П. Ершова, г. Новосибирск

E-mail: anureev@iis.nsk.su

Операционно-онтологический подход к формальной спецификации компьютерных языков был разработан автором как методология создания операционной семантики компьютерного языка на базе его онтологии – так называемой операционно-онтологической семантики. В статье предлагается формализм для описания операционно-онтологических семантик компьютерных языков – онтологические системы переходов, и проводится сравнение онтологического описания компьютерных языков с традиционным грамматическим описанием.

Ключевые слова:

Спецификация, операционно-онтологическая семантика, компьютерные языки, онтологии, онтологические системы переходов.

Key words:

Specification, operational ontological semantics, computer languages, ontologies, ontological transition systems.

Введение

Традиционный подход к формальной спецификации компьютерных языков (КЯ) обычно базируется на операционной семантике языка, описываемой в терминах помеченной системы переходов, в которой применимость правил перехода (правил операционной семантики) к некоторой конструкции языка определяется посредством синтаксического сопоставления с образцом на уровне грамматики этого языка.

Ю. Гуревичем был предложен логико-алгебраический подход к решению этой задачи, основанный на машинах абстрактных состояний (abstract state machines) [1]. Машины абстрактных состояний – специальный вид систем переходов, состояниями в которых являются алгебраические системы. Выбор подходящей сигнатуры алгебраической системы позволяет приблизить формальную спецификацию КЯ к его естественной концептуальной структуре. Примеры приложений этого формализма могут быть найдены в [2]. Этот подход реализован в языках ASML [3] и XASM [4].

Операционно-онтологический подход к формальной спецификации КЯ был предложен автором в [5] как методология создания операционной семантики КЯ на базе его онтологии – так называемой операционно-онтологической семантики. В отличие от обычной операционной семантики, которая не накладывает никаких ограничений на состояния, в операционно-онтологической семантике состояния определяются как онтологические модели (множества экземпляров понятий онтологии) КЯ.

В статье для описания операционно-онтологической семантики КЯ предлагается специальный вид помеченных систем переходов – онтологические системы переходов (ОСП), в которых онтологические модели определяются как алгебраические системы специального вида. Использование алгебраических систем в качестве состояний позволяет также рассматривать ОСП как специальный случай машин абстрактных состояний.

Соглашения и обозначения

Будем использовать для представления выражений, обозначающих данные или действия, списковую нотацию $(A_1...A_n)$, в которой подвыражения A_i разделены пробелами. Подвыражение A_i , которое не является списком, будем называть атомом. Например, применение функции f к аргументам $v_1,...,v_n$ запишется в этой нотации как $(f\ v_1...v_n)$, а кортеж из элементов $v_1,...,v_n$ как $(v_1...v_n)$.

Пусть *bool* обозначает множество $\{true, false\}$, а *undef* – отсутствие значения у некоторого выражения.

Атрибутивной структурой *as* называется конечное множество пар $\{(a_1v_1),..., (a_nv_n)\}$ таких, что $(a_i \neq a_j)$ при $(i \neq j)$. Объект a_i называется атрибутом структуры *as*, а объект v_i – значением атрибута a_i структуры *as*. Атрибутивная структура *as* имеет упрощенную запись $(a_1.v_1,..., a_n.v_n)$. На атрибутивных структурах определены операции доступа (.) и модификации (*upd*): $(as.a_i=v_i)$ и $((upd\ as\ a\ v)=aas)$, где атрибутивная структура *aas* может отличаться от *as* только значением атрибута *a* и $(aas.a=v)$. Эти операции расширяются на случаи последовательных и одновременных доступов и модификаций с помощью модификаторов *#s* и *#p*, соответственно:

- $(a.(#s\ b_1...b_n)=a.b_1.(#s\ b_2,...,b_n))$;
- $(a.(#s)=a)$;
- $(a.(#p\ b_1...b_n)=(a.b_1...a.b_n))$;
- $((upd\ a\ (#s\ b_1...b_n)\ e)=(upd\ a.b_1\ (#s\ b_2,...,b_n)))$;
- $((upd\ a\ (#s)\ e)=a)$;
- $((upd\ a\ (#p\ b_1...b_n)\ (#p\ e_1...e_n))=(upd\ (upd\ a\ b_1\ e_1)\ (#p\ b_2...b_n)\ (#p\ e_2...e_n)))$;
- $((upd\ a\ (#p))=(#p)\ a)$.

Операционная семантика компьютерных языков

Операционная семантика КЯ обычно определяется как помеченная система переходов.

Помеченная система переходов *lts* – это тройка $(sts\ labs\ tr)$, где *sts* и *labs* – множества, элементы которых называются состояниями и метками, соответственно, функция $tr \in (sts \times labs \times sts) \rightarrow bool$ назы-

вается функцией перехода. Говорят, что из состояния s можно перейти в состояние ss по метке lab , если $(tr\ s\ lab\ ss)$.

Пусть lan — некоторый КЯ. Конструкции (операторы, выражения, классы, интерфейсы и т. д.) компьютерного языка lan будем называть lan -выражениями. Пусть $(alph\ lan)$ — обозначение для алфавита языка lan . Пусть $exprs$ обозначает множество конечных последовательностей lan -выражений.

При описании операционной семантики lan метку lab можно представить атрибутивной структурой с обязательными атрибутами in и out такими, что $lab.in \in exprs$ и $lab.out \in exprs$. Говорят, что выполнение $lab.in$ в s сводится к выполнению $lab.out$ в ss , если $(tr\ s\ lab\ ss) = true$. Случай, когда tr определяется так, что результирующая последовательность всегда пуста, называется операционной семантикой входа–выхода.

В зависимости от выбранного описания операционной семантики языка lan метка lab может иметь те или иные дополнительные атрибуты. Например, дополнительным атрибутом может быть атрибут, значением которого является значение, возвращаемое при выполнении $lab.in$.

Конфигурацией называется кортеж $(e\ s)$, где $e \in exprs$ и $s \in sts$. Говорят, что lts переходит из конфигурации $(e\ s)$ в конфигурацию $(ee\ ss)$, если $(tr\ s\ lab\ ss) = true$ для некоторой метки lab такой, что $lab.in = e$ и $lab.out = ee$.

Отношение перехода tr определяется как объединение отношений перехода $(tr\ es)$ для отдельных классов es последовательностей выражений $lab.in$. Каждое отношение $(tr\ es)$ определяется множеством правил перехода (правил операционной семантики). Правило перехода r является атрибутивной структурой с атрибутами $cond$ и res , определяющими условие и результат применения правила r , соответственно.

Условие применения r правила r определяет es с помощью логической функции $match$ следующим образом: $e \in es$ в состоянии s тогда и только тогда, когда существует mr такой, что $(match\ e\ r\ cond\ mr\ s) = true$. Объект mr называется результатом сопоставления e с правилом r и описывает информацию, полученную при этом сопоставлении.

Множество результирующих последовательностей выражений ees для исходной последовательности e в состоянии s определяется с помощью логической функции $transform$ следующим образом: $ee \in ees$ в s тогда и только тогда, когда $(transform\ e\ ee\ r\ res\ mr\ s\ ss) = true$ для некоторого состояния $ss \in sts$. Таким образом, $r.res$ содержит информацию о том, что получается из e , а функция $transform$ на основе этой информации выделяет класс ees результирующих выражений, к выполнению которых сводится выполнение e .

Функция $match$ в типичных операционных семантиках для КЯ выполняет сопоставление с образцом либо на уровне строки исходной последовательности выражений, либо на уровне последовательности синтаксических деревьев этих выра-

жений. В дальнейшем будем представлять деревья в виде термов.

В первом случае $r.cond$ определяется как атрибутивная структура с атрибутами $samp$ и var , где $r.cond.samp$ — последовательность строк символов алфавита $(union\ (alph\ lan)\ vs)$, называемая образцом правила r ; $r.cond.var$ — атрибутивная структура, атрибутами которой являются элементы, называемые переменными образца $r.cond.samp$, а значениями этих атрибутов — элементы множества $\{str, seq\}$, называемые типами переменных образца; vs — множество переменных образца. Результат сопоставления mr есть подстановка на множестве переменных $r.cond.var$ (функция, сопоставляющая переменным образца их значения). Значениями переменных образца типа str и seq являются строки и последовательности строк, соответственно. Функция $match$ определяется следующим образом: $(match\ e\ r\ cond\ mr\ s) = true$ тогда и только тогда, когда $(str\ subst\ r.cond.samp\ mr) = e$, где $str\ subst$ — функция, выполняющая подстановку на строках (заменяющая переменные образца в каждой из строк последовательности $r.cond.samp$ на их значения).

Во втором случае $r.cond$ определяется как атрибутивная структура с атрибутами $samp$ и var , где $r.cond.samp$ — последовательность термов, называемая образцом правила r ; $r.cond.var$ — атрибутивная структура, атрибутами которой являются элементы, называемые переменными образца $r.samp$, а значениями этих атрибутов — элементы множества $\{term, seq\}$, называемые типами переменных образца. Результат сопоставления mr есть подстановка на множестве переменных $r.cond.var$. Значениями переменных образца типа $term$ и seq являются термы и последовательности термов, соответственно. Функция $match$ определяется следующим образом: $(match\ e\ r\ cond\ mr\ s) = true$ тогда и только тогда, когда $(subst\ r.cond.samp\ mr) = e$, где $subst$ — функция, выполняющая подстановку на термах (заменяющая переменные образца в каждом из термов последовательности $r.cond.samp$ на значения соответствующих переменных образца).

Рассмотрим в качестве примера правило операционной семантики для оператора присваивания $(x := u)$, где x — целочисленная переменная, u — целое число. Пусть $((x_1\ v_1) \dots (x_n\ v_n))$ обозначает подстановку sub на переменных x_i такую, что $(sub\ x_i) = v_i$. Пусть состояниями являются атрибутивные структуры, где атрибуты — целочисленные переменные, а значения атрибутов — значения этих переменных. Тогда правило перехода r для оператора присваивания имеет вид $(cond: (samp: (a := b)\ c, var: (a: term, b: term, c: seq)), res: (upd\ s\ a\ b))$, а функции $match$ и $transform$ принимают для этого правила следующий вид:

- $(match\ e\ r\ cond\ mr\ s) = true$ тогда и только тогда, когда e имеет вид $(set\ aa\ bb)\ cc$ и $mr = ((a\ aa)\ (b\ bb)\ (c\ cc))$;
- $(transform\ e\ ee\ r\ res\ mr\ s\ ss) = true$ тогда и только тогда, когда $ss = (upd\ s\ aa\ bb)$ и $ee = cc$.

Для современных КЯ помимо выражений, описывающих инструкции КЯ, требуется определить операционную семантику выражений, описывающих концепции КЯ, такие как, например, просачивание исключений, разрешение перегрузки операций, взаимодействие приложения с операционным окружением, нахождение динамического типа объекта в объектно-ориентированных КЯ и т. д.

При типичном грамматическом определении операционной семантики КЯ мы получаем громоздкие и трудные для понимания спецификации для современных концептуально-сложных КЯ. Естественный способ преодолеть эту сложность состоит в переходе от описаний, ориентированных на грамматику КЯ, к концептуальным описаниям, использующим онтологию КЯ. В качестве формализма, обеспечивающего концептуальные описания, предлагается использовать специальный вид помеченных систем переходов — онтологические системы переходов.

Онтологические системы переходов

Состояниями в ОСП являются алгебраические системы специального вида. Пусть $alph$ — некоторое множество символов, не содержащее символа $_$. Состояния над $alph$ определяются как атрибутивные структуры с атрибутами sig , val , int и arg . Пусть A — алгебраическая структура. Множество $A.sig$ называется сигнатурой A , а элементы $A.sig$ — составными символами A . Составной символ определяется как список, элементы которого принадлежат $alph \cup \{_ \}$, а i -е вхождение символа $_$ в составной символ называется i -м аргументом этого составного символа. Число аргументов в составном символе называется его местностью. Множество $A.val$ называется носителем A . Функция $A.int$, определенная на $A.sig$, называется интерпретацией символов A . Для нее выполнено свойство: если $B \in A.sig$ и n — местность B , то $(A.int B) \in A.val^n \rightarrow A.val$. Например, для операции равенства соответствующий составной символ будет иметь вид $(_ = _)$. Функция $A.arg$, определенная на $A.sig$, называется аргументной интерпретацией символов A . Для нее выполнено следующее свойство: если $B \in A.sig$ и n — местность B , то $(A.arg B)$ — список длины n , и любой элемент этого списка принадлежит множеству $\{val, itself\}$.

Говорят, что выражение E является примером составного символа B местности n относительно списка $(v_1 \dots v_n)$, если E получается из B последовательной заменой вхождений символа $_$ слева направо на $v_1, \dots, v_n$, соответственно.

Значение $(val E A)$ выражения в алгебраической системе A определяется следующим образом:

- если E — атом, то $(val E A) = E$;
- если E — список, E — пример $B \in A.sig$ местности n относительно $(v_1 \dots v_n)$, то $(val E A) = A.int(vv_1 \dots vv_n)$;
- если E — список и E — пример более одного символа из $A.sig$, или E не является примером ни одного символа из $A.sig$, то $(val E A) = undef$.

Объекты vv_i в вышеприведенном определении задаются следующим образом. Если val — i -й эле-

мент списка $(A.arg B)$, то $vv_i = (val v_i A)$. Если $itself$ — i -й элемент списка $(A.arg B)$, то $vv_i = v_i$.

Онтологической сигнатурой $o-sig$ будем называть множество составных символов над $alph$, специфицирующее элементы онтологии (понятия, атрибуты и т. п.). Будем называть элементы множества $o-sig$ онтологическими символами.

Рассмотрим пример онтологической сигнатуры, одновременно последовательно помечая аргументы буквами a , b , c и определяя неформальный смысл символов, входящих в нее:

- $(a \text{ is concept})$ означает, что a — понятие;
- $(a \text{ is attribute of } b)$ означает, что a — атрибут понятия b ;
- $(a \text{ is attribute of } b \text{ of type } c)$ означает, что a — атрибут понятия b типа c .

Сигнатурой экземпляризации $i-sig$ будем называть множество составных символов над $alph$, специфицирующее связи элементов онтологии с экземплярами. Будем называть элементы множества $i-sig$ символами экземпляризации.

Рассмотрим пример сигнатуры экземпляризации, одновременно последовательно помечая аргументы буквами a , b , c и определяя неформальный смысл символов, входящих в нее:

- $(a \text{ is } b)$ означает, что a — экземпляр понятия b ;
- $(value \text{ of attribute } a \text{ of } b)$ означает значение атрибута a экземпляра b некоторого понятия. Для этого составного символа также используется упрощенная запись $b.a$;
- $(value \text{ of attribute } a \text{ of instance } b \text{ of } c)$ означает значение атрибута a экземпляра b понятия c . Этот символ используется, чтобы разрешить конфликт в случае, когда b является экземпляром нескольких понятий, имеющих атрибут a .

Пусть val — некоторое множество элементов, называемое носителем. Оно описывает элементы, над которыми определяется онтологическая структура.

Пусть $c-sig$ — сигнатура над $alph$, называемая константной сигнатурой. Она включает символы, семантика (интерпретация) которых не меняется при переходе из состояния в состояние в ОСП. Пусть $c-int$ — (постоянная) интерпретация этих символов.

Пусть $h-sig$ — бесконечная сигнатура составных символов вида (f) , где $f \in alph$, называемая сигнатурой истории перехода. Она включает символы, семантика (интерпретация) которых в конкретном состоянии s содержит информацию о состояниях, предшествующих s относительно tr , т. е. историю перехода в s .

Пусть $sig = (o-sig \ i-sig \ c-sig \ h-sig)$, где сигнатуры $o-sig$, $i-sig$, $c-sig$ и $h-sig$ попарно не пересекаются.

Онтологическая система переходов относительно $(alph \ sig \ val \ c-int)$ — это помеченная система переходов $(sts \ labs \ tr)$, где sts — множество состояний над $alph$ с носителем val , такое, что

- $s.sig \setminus (o-sig \cup i-sig \cup c-sig)$ конечно для любого $s \in sts$;
- $(s.sig \setminus (o-sig \cup i-sig \cup c-sig)) \subseteq h-sig$ для любого $s \in sts$;
- $(s.int \ f) = c-int$ для любых $s \in sts$ и $f \in c-sig$;

- последовательности *lab.in* и *lab.out* состоят из экземпляров понятий и метавыражений. Метавыражения позволяют напрямую оперировать состояниями.
 - если $(tr\ s\ lab\ ss)$, то $(s.sig \cap h.sig) \subseteq (ss.sig \cap h.sig)$. Метавыражения бывают двух видов и имеют следующую семантику:
 - если *lab.in* имеет вид $((a_1...a_n)::=b)\ eee$, то $(tr\ s\ lab\ ss)=true$ тогда и только тогда, когда *lab.out*=*eee*, $ss=(upd\ s\ (\#s\ int\ ((val\ aa_1\ s.int)\dots (val\ aa_n\ s.int)))\ (val\ b\ s.int))$. Метавыражение такого вида называется метаприсваиванием;
 - если *lab.in* имеет вид $(assume\ a)\ eee$, то $(tr\ s\ lab\ ss)=true$ тогда и только тогда, когда *lab.out*=*eee*, $ss=s$ и $(val\ a\ s.int)=true$. Метавыражение такого вида называется метасловием продолжения.
- Часто используемые метаприсваивания $(a::=true)$ и $(a::=undef)$ имеют упрощенный синтаксис $(a::=t)$ и $(a::=u)$, соответственно.
- ОСП сохраняет онтологию, если для любых $s \in sts$, $ss \in sts$ и $lab \in labs$ из $(tr\ s\ lab\ ss)=true$ следует $(s.int\ f)=(ss.int\ f)$ для любого символа $f \in o.sig$.

Операционно-онтологическая семантика компьютерных языков

Операционно-онтологическая семантика (ООС) КЯ является видом операционной семантики, при котором в качестве помеченных систем переходов используются ОСП.

Формально ООС языка *lan* есть кортеж $(o-ots\ e-ots\ alph\ sig\ val\ c-int)$, где *o-ots* – ОСП относительно $(alph\ sig\ val\ c-int)$, определяющая онтологию *lan*, *e-ots* – ОСП относительно $(alph\ sig\ val\ c-int)$, сохраняющая онтологию и определяющая семантику *lan*-выражений.

Исходная последовательность *e* в конфигурации $(e\ s)$ состоит из экземпляров понятий онтологии языка *lan* и метавыражений. Семантика метавыражений определена выше, а семантика экземпляров понятий задается с помощью правил.

Пусть *eee* – последовательность выражений.

Правило ООС *r* для экземпляров понятий имеет вид: $(if\ a\ hvar\ b\ then\ c)$, где *a* – *lan*-выражение, называемое условием применимости правила *r*; *b* – последовательность промежуточных переменных, разделенных запятыми; *c* – последовательность *lan*-выражений и метавыражений.

Пусть $b=x_1...x_n$ и $c=c_1...c_m$. Пусть $e=e_1\ eee$, где e_1 – экземпляр некоторого понятия. Пусть $v_j \in val$, $f_j \in v-sig$ и $f_j \notin s.sig$ для всех $j \in \{1,...,n\}$.

Пусть $sss=(upd\ s\ (\#p\ sig\ int)\ (\#p\ (s.sig \cup \{f_1,...,f_n\})\ (upd\ s.int\ (\#p\ (f_1)...(f_n))\ (\#p\ v_1...v_n))))$.

Пусть $sub=((\#e_1)\ (x_1\ f_1())\dots (x_n\ f_n\ ()))$. Символ $\#$ в условии применимости *a* обозначает первый элемент исходной выполняемой последовательности *e* в случае, если он является экземпляром некоторого понятия языка *lan*.

Пусть $cc=cc_1...cc_m$, где cc_i определяются следующим образом:

- если c_i – *lan*-выражение, то $cc_i=(val\ (subst\ c_i\ sub)\ sss.int)$;
- если c_i – метавыражение, то $cc_i=(subst\ c_i\ sub)$.

Атрибуты *cond* и *res* и функции *match* и *transform* определяются для правила *r* следующим образом:

- $r.cond=a$ и $r.res=(b\ c)$;
- $(match\ e\ r.cond\ mr\ s)=true$ тогда и только тогда, когда $(s.int\ (subst\ a\ ((\#e_1))))=true$ и $mr=e_1$;
- $(transform\ e\ ee\ r.res\ mr\ s\ ss)=true$ тогда и только тогда, когда $ss=sss$ и $ee=(del-stop\ c\ cc\ e)$, где
 - если $c_m=stop$, то $(del-stop\ c\ cc\ e)=cc_1...cc_{m-1}$;
 - в противном случае $(del-stop\ c\ cc\ e)=cc\ eee$.

Объект *stop* служит признаком исключения хвоста *eee* исходной последовательности *e* из результирующей последовательности *lab.out*.

В качестве примера рассмотрим ООС некоторых простых операторов.

Онтология оператора присваивания $(x:=u)$, где *x* – имя переменной, *u* – либо переменная, либо целое число, а значения переменных ограничены множеством целых чисел, задается метаприсваиваниями $((assignment-statement\ is\ concept)::=t)$, $((left-side\ is\ attribute\ of\ assignment-statement)::=t)$ и $((right-side\ is\ attribute\ of\ assignment-statement)::=t)$.

Первое метаприсваивание определяет новое понятие *assignment-statement*, экземплярами которого являются операторы присваивания. Два последних метаприсваивания определяют атрибуты *left-side* и *right-side* для операторов присваивания, значением которых являются левая и правая части оператора присваивания, соответственно.

Семантика оператора присваивания задается правилом $(if\ (\#is\ assignment-statement)\ hvar\ w\ then\ (assume\ (w=\#.left-side))\ ((w\ is\ variable)::=t)\ \#.right-side\ (w.value::=(val)))$.

Правило для оператора присваивания использует понятие *variable*, добавляемое в онтологию *o-ots* посредством метаприсваиваний $((variable\ is\ concept)::=t)$ и $((value\ is\ attribute\ of\ variable)::=t)$. В случае, если бы КЯ помимо целочисленного типа включал переменные других типов, к понятию *variable* был бы добавлен атрибут *type*, определяемый метаприсваиванием $((type\ is\ attribute\ of\ variable)::=t)$.

Кроме того, правило для оператора присваивания вычисляет правую часть оператора присваивания, выполняя выражение $\#.right-side$. Поскольку правая часть оператора присваивания является либо переменной, либо целым числом, вычисление правой части задается двумя правилами $(if\ (\#is\ variable)\ then\ ((val)::=\#.value))$ и $(if\ (\#is\ integer)\ then\ ((val)::=\#))$. Во втором правиле используется предопределенное понятие *integer*, экземплярами которого являются целые числа. Множество экземпляров предопределенного понятия нельзя изменять с помощью метаприсваивания.

Онтология оператора удаления переменной $(del\ x)$ задается метаприсваиваниями $((delvar-statement\ is\ concept)::=t)$ и $((variable\ is\ attribute\ of\ delvar-statement)::=t)$. Семантика этого оператора задается правилом $(if\ (\#is\ del-statement)\ hvar\ w\ then\ (assume\ (w=\#.variable))\ (w.value::=u)\ ((w\ is\ variable)::=u))$.

Онтология условного оператора $(if\ then\ b\ else\ c)$ задается метаприсваиваниями $((if-statement\ is\ concept)::=t)$, $((condition\ is\ attribute\ of\ if-statement)::=t)$,

$((then\ is\ attribute\ of\ if\ statement)::=t)$ и $((else\ is\ attribute\ of\ if\ statement)::=t)$.

Заметим, что онтологические символы позволяют типизировать атрибуты. Например, если необходимо учитывать тип атрибута *then*, то мета-присваивание для этого атрибута следует переписать к виду $((then\ is\ attribute\ of\ if\ statement\ of\ type\ statement)::=t)$.

Семантика условного оператора задается правилами $(if\ (\# \ is\ if\ statement)\ then\ \#condition\ (assume\ ((val)=true))\ \#.then)$ и $(if\ (\# \ is\ if\ statement)\ then\ \#condition\ (assume\ ((val)=false))\ \#.else)$.

В качестве иллюстрации применения правил рассмотрим выполнение следующей исходной последовательности операторов: $(x:=3)(y:=x)(del\ x)(x)$.

При выполнении этой последовательности используются правила для оператора присваивания, оператора удаления переменной, вычисления переменной и целого числа. Опишем результат применения этих правил в виде последовательностей метавыражений, соответствующих этим операторам:

$(assume\ ((f_1)=x))\ ((f_1)\ is\ variable)::=t)\ ((val)::=3)$
 $((f_1).value)::=3)$
 $(assume\ ((f_2)=y))\ (((f_2)\ is\ variable)::=t)$
 $((val)::=x.value)\ ((f_2).value)::=3)$
 $(assume\ ((f_3)=x))\ ((f_3).value)::=u)\ (((f_3)\ is\ variable)::=u)$
 $((val)::=u)$

Выводы

Предложен новый подход к формальной спецификации КЯ, основанный на специальном виде помеченных систем переходов — онтологических системах перехода и модифицированном понятии операционно-онтологической семантики. Проведено его сравнение с традиционным подходом, основанным на помеченных системах переходов.

Работа выполнена при финансовой поддержке гранта РФФИ № 11-01-00028-а и междисциплинарного интеграционного проекта СО РАН № 3 «Принципы построения онтологии на основе концептуализаций средствами логических дескриптивных языков».

СПИСОК ЛИТЕРАТУРЫ

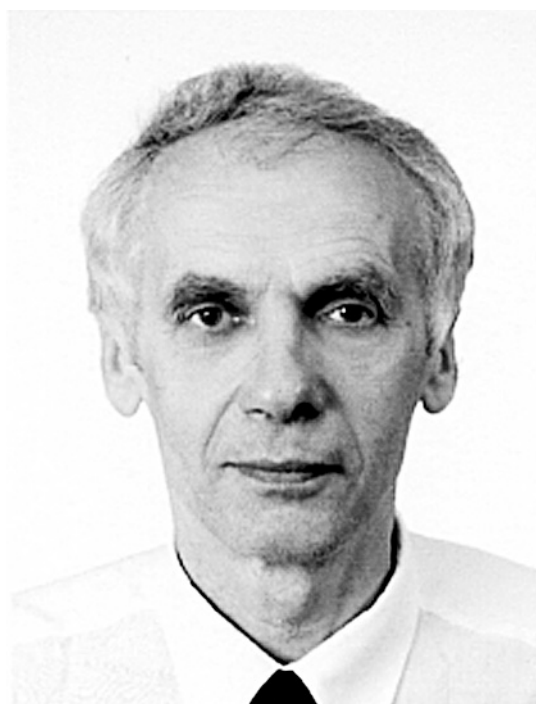
1. Гуревич Ю. Последовательные машины абстрактных состояний // Формальные методы и модели информатики: Сб. науч. тр. Серия «Системная информатика». — Новосибирск: Изд-во СО РАН, 2004. — Вып. 9. — С. 7–50.
2. Huggins J. Abstract State Machines Web Page. URL: <http://www.eecs.umich.edu/gasm> (дата обращения: 20.04.2012).
3. AsmL: The Abstract State Machine Language. URL: <http://research.microsoft.com/en-us/projects/asml/> (дата обращения: 20.04.2012).

4. XasM: An Extensible, Component-Based Abstract State Machines Language. URL: <http://xasm.sourceforge.net/XasmAnl00/XasmAnl00.html> (дата обращения: 20.04.2012).
5. Ануреев И.С. Операционно-онтологический подход к формальной спецификации языков программирования // Программирование. — 2009. — № 1. — С. 1–11.

Поступила 30.09.2012 г.

Наши юбиляры

ПРОФЕССОРУ В.Г. СПИЦЫНУ – 65 ЛЕТ



6 января 2013 г. исполнилось 65 лет со дня рождения профессора ТПУ, доктора технических наук, профессора Владимира Григорьевича Спицына. После окончания с золотой медалью средней школы в Алтайском крае (рабочий пос. Шипуново) в 1965 г. он приехал в г. Томск и поступил на радиофизический факультет Томского государственного университета (ТГУ). Завершив обучение в ТГУ в 1970 г., В.Г. Спицын по распределению поступает на работу в должности младшего научного сотрудника в Сибирский физико-технический институт (СФТИ) при ТГУ. В последующие годы Владимир Григорьевич в качестве ответственного исполнителя и научного руководителя участвует в исследованиях, проводившихся в отделе радиофизики Сибирского физико-технического института при Томском государственном университете в рамках ряда хоздоговорных и госбюджетных НИР, вы-

полнявшихся по постановлениям Правительства. В период с 1981 по 1995 гг. В.Г. Спицын исполняет обязанности заместителя заведующего ионосферной лабораторией СФТИ. В 1981 г. он защищает диссертацию на соискание ученой степени кандидата физико-математических наук. В 1989 г. решением ВАК ему присваивается ученое звание «старший научный сотрудник».

В 1996 г. Владимир Григорьевич поступает в докторантуру Томского государственного университета и приступает к работе над докторской диссертацией. В 1996–1998 гг. он выигрывает тревел-гранты РФФИ № 96-02-26649, 97-02-27075 и 98-02-26996 и выступает с докладами на научных симпозиумах в Греции, Канаде и США: Trans Black Sea Region Symposium on Applied Electromagnetism (Мецово, Греция, 1996 г.); IEEE Antennas and Propagation Society International Symposium and North

American Radio Science Meeting (Монреаль, Канада, 1997 г.) и IEEE Antennas and Propagation Society International Symposium and USNC/URSI Radio Science Meeting (Атланта, США, 1998 г.).

В 1998–2000 гг. В.Г. Спицын руководит инициативным проектом «Моделирование и диагностика воздействия запусков космических аппаратов на состояние электромагнитного поля окружающей среды», поддержанным грантом РФФИ № 98805503182. После окончания докторантуры в январе 2000 г. он поступает на работу в должности доцента на кафедру вычислительной техники Томского политехнического университета. 28 июня 2000 г. на заседании диссертационного совета Д 063.53.03 при ТГУ Владимир Григорьевич защищает диссертационную работу «Моделирование рассеяния радиоволн на возмущениях ионосферной плазмы, создаваемых космическим аппаратом» на соискание ученой степени доктора технических наук по специальностям: 05.13.16 – применение вычислительной техники, математического моделирования и математических методов в научных исследованиях; 01.04.03 – радиоп физика.

Автором исследован механизм образования области высокой концентрации заряженных частиц ионосферной плазмы в окрестности границы факела реактивного двигателя космического аппарата. Результаты проведенного численного моделирования процесса диффузии ионов ионосферной плазмы внутрь сверхзвукового потока слабоионизованного газа подтвердили существование области высокой концентрации вблизи границы газовой струи. На основе проведенных расчетов построены модели возмущений ионосферной плазмы, создаваемых сверхзвуковыми потоками выхлопных газов реактивных двигателей космических аппаратов. Реализована статистическая модель многократного рассеяния радиоволн на образующемся турбулентном плазменном теле вращения и проведено сопоставление результатов расчетов с известными американскими экспериментальными данными по радиозондированию вслед запускаемому космическому аппарату. Предложена, реализована и апробирована трехмерная модель ракурсного рассеяния коротких радиоволн на возмущении ионосферной плазмы, создаваемом ударной волной, генерируемой космическим аппаратом при движении с работающим двигателем.

22 октября 2001 г. на заседании Ученого Совета Томского политехнического университета В.Г. Спицын избирается на должность профессора кафедры вычислительной техники ТПУ. С 14 ноября по 14 декабря 2001 г. он находился на стажировке в г. Страсбурге (Франция) по приглашению администрации Института ядерных исследований университета Луи Пастера.

В 2002 г. В.Г. Спицыным подготовлен проект по изданию книги «Моделирование рассеяния радиоволн на возмущениях ионосферной плазмы, создаваемых космическим аппаратом», который

был поддержан грантом РФФИ № 03-02-30012. В 2003 г. монография вышла из печати в издательстве «Физматлит», г. Москва.

В 2004 г. Владимир Григорьевич выступает с тремя устными докладами на симпозиуме EuroElectromagnetics-2004 в г. Магдебург (ФРГ). По предложению редколлегии журнала «IEEE Transactions on Antennas and Propagation» он становится рецензентом статей, поступающих в журнал. В этом же году он избран членом-корреспондентом Международной академии информатизации.

В 2006 г. В.Г. Спицыным подготовлен проект «Разработка технологии автоматизированного улучшения качества цифровых изображений на основе применения эволюционирующей нейронной сети», поддержанный грантом РФФИ № 06-08-00840. В этом же году он выигрывает тревел-грант РФФИ № 06-02-26939 и выступает с тремя устными докладами на объединенном научном симпозиуме в США (Joint IEEE APPS/URSI/AMEREM 2006 Symposium, Albuquerque, USA).

В 2007 г. в диссертационном совете при ТПУ Ю.Р. Цой успешно защитил диссертацию на соискание ученой степени кандидата технических наук «Нейроэволюционный алгоритм и программные средства для обработки изображений» по специальности 05.13.01, выполненную под руководством В.Г. Спицына. В этом же году Владимир Григорьевич обучался по программе «Engineering Education: Curriculum Development, Teaching and Quality Assurance» aimed at providing competencies for improvement and evaluation of engineering programmes against RAEE criteria and procedures consistent with «EUR-ACE Framework Standards for the Accreditation of Engineering Programmes», Barcelona, Spain, 22 September – 04 October, 2007. В декабре 2007 г. В.Г. Спицын принимал участие в «European Projects» Russia–Austrian Seminar «Development of Integrated Research-Educational-Information Media of the World Level at the University», Vienna, Austria, December 2–8, 2007.

В 2008 г. В.Г. Спицыным подготовлен проект «Создание программного комплекса автоматизированной обработки изображений и распознавания образов на основе применения искусственных нейронных сетей, регуляторных сетей и эволюционных алгоритмов», поддержанный грантом РФФИ № 09-08-00309. В этом же году он выигрывает тревел-грант РФФИ № 08-02-08541 и выступает с тремя устными докладами на научном симпозиуме EuroElectromagnetics-2008 в г. Лозанна (Швейцария).

В 2010 г. Владимир Григорьевич принимал участие в Scientific Workshop «Modern Technologies of Education and Research in HEI of France and Russia» held under the patronage of Paris Institute of Technology Paris Tech in France, Paris, October 10–17, 2010. В декабре 2010 г. А.А. Белоусовым в диссертационном совете при ТПУ успешно защищена диссертация на соискание ученой степени кандидата технических наук «Алгоритмы и программные средства

эволюционной обработки изображений» по специальности 05.13.01, выполненная под руководством В.Г. Спицына.

С 2009 В.Г. Спицын является членом программных комитетов ряда международных конференций:

- Workshop on Soft Computing in Image Processing and Computer Vision (SCIPCV) in conjunction with International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV 2009), Las Vegas, July 13–16, 2009 (<http://vision.cs.aston.ac.uk/CfP/SCIPCV2009/scipcv-committee.html>).
- 9th International Conference on Information Technology and Applications in Biomedicine (ITAB), Larnaca, Cyprus, November 5–7, 2009 (<http://www.cs.ucy.ac.cy/itab2009/committees.html>).
- Workshop on Nature Inspired Biometrics (WNIB) in conjunction with World Congress on Nature and Biologically Inspired Computing (NABIC 2009), Coimbatore, India, December 9–11, 2009 (<http://www-staff.lboro.ac.uk/~cogs/CfP/WNIB2009/>).
- 2nd Workshop on Soft Computing in Image Processing and Computer Vision (SCIPCV) in conjunction with International Conference on Image Processing, Computer Vision, and Pattern Recognition, Las Vegas, Nevada, USA, July 12–15, 2010 (<http://www-staff.lboro.ac.uk/~cogs/CfP/SCIPCV2010/scipcv-committee.html>).
- 3^d Workshop on Soft Computing in Image Processing and Computer Vision (SCIPCV) in conjunction with International Conference on Image Processing, Computer Vision, and Pattern Recognition, Las Vegas, Nevada, USA, July 18–21, 2011 (<http://www-staff.lboro.ac.uk/~cogs/CfP/SCIPCV2011/scipcv-committee.html>).
- Online Conference on Soft Computing in Industrial Applications WWW, December 5–16, 2011 (<http://wsc16.cs.lboro.ac.uk/organization.html>).

В 2011 г. Владимиром Григорьевичем подготовлен проект «Создание комплексных технологий распознавания объектов на изображениях на основе применения моделей зрительного восприятия и методов вычислительного интеллекта», поддержанный грантом РФФИ № 12-08-00296. В этом же году В.Г. Спицын назначен руководителем магистерской программы «Компьютерный анализ и интерпретация данных». С 2011 г. по его инициативе в указанную магистерскую программу введено

5 новых дисциплин, посвященных применению методов искусственного интеллекта в распознавании образов на изображениях и видеопоследовательностях.

В 2012 г. приказом Министерства образования и науки Российской Федерации В.Г. Спицыну присвоено ученое звание профессора. В 2013 г. в диссертационном совете при ТГУ Ю.А. Болотовой успешно защищена диссертация на соискание ученой степени кандидата технических наук «Алгоритмы обработки и анализа изображений иерархической временной сетью» по специальности 05.13.01, выполненная под руководством В.Г. Спицына.

Владимир Григорьевич опубликовал 240 научных работ, в том числе 2 монографии.

Параллельно с научной работой В.Г. Спицын читает лекции по дисциплинам: «Представление знаний в информационных системах», «Интеллектуальные информационные системы», «Информационная безопасность и защита информации», «Интеллектуальные системы», «Нейронные сети», «Методы распознавания образов» и «Методы вычислительного интеллекта». По тематике указанных дисциплин опубликовано 10 учебных пособий. Под руководством В.Г. Спицына студентами ТПУ, ТГУ и ТУСУР выполнены и защищены 30 дипломных проектов и 4 магистерские диссертации. Владимир Григорьевич является членом двух диссертационных советов по защите докторских диссертаций при Томском политехническом университете, руководит подготовкой семи аспирантов.

В настоящее время совместно с аспирантами и студентами он продолжает исследования по следующим научным направлениям:

- создание комплексных технологий распознавания объектов на изображениях на основе применения моделей зрительного восприятия и методов вычислительного интеллекта;
- разработка способов распознавания искаженных текстов на основе применения вейвлет-преобразований, метода главных компонент и нейронных сетей;
- создание методов моделирования нелинейного распространения волн в трехмерно-неоднородных дискретных случайных средах.

Коллектив Института кибернетики ТПУ желает юбиляру активного творческого долголетия, дальнейших успехов в научной и преподавательской деятельности.

ПРОФЕССОРУ Б.Л. АГРАНОВИЧУ – 75 ЛЕТ



Агранович Борис Львович родился 5 марта 1938 г. в городе Константиновка, Донецкой области (Украина). После окончания школы в 1955 г. работал в учреждениях связи и на радиотехническом заводе, увлекался разработкой радиолюбительских схем.

По совету друзей, которые уже учились в Томском политехническом институте (ТПИ) на разных факультетах и восторженно отзывались о радиотехническом факультете, в 1957 г. приехал в Томск и поступил на радиотехнический факультет ТПУ. До настоящего времени Борис Львович с благодарностью вспоминает таких самоотверженных, высокопрофессиональных учёных и педагогов, как К.М. Шульженко, С.П. Кузнецова, В.Н. Белоусова, В.И. Перегудова, В.С. Мелихова и многих других. Они всегда видели в студентах соратников, а студентам доставляло огромное удовольствие общение с ними и мотивировало успешно учиться и заниматься научно-исследовательской работой. Кроме того, это были наставники, они учили жить по-настоящему, отдаваться любимому делу полностью.

Во время учёбы в институте Борис Львович активно участвовал в научно-исследовательской работе на кафедре «Теоретические основы радиотехники» под руководством К.М. Шульженко. В 1962 г. закончил радиотехнический факультет ТПИ по специальности «Радиотехника». После окончания ТПИ занимал там должность ассистента, в сентябре 1962 г. был переведен во вновь созданный Томский институт радиоэлектроники и электронной техники (ТИРиЭТ), где занимал должности ассистента, старшего преподавателя (1968), доцента (1968), а с сентября 1970 по 1974 гг. — декана радиотехнического факультета.

Б.Л. Агранович в 1967 г. защитил кандидатскую диссертацию по решению проблем проектирования прецизионных радиоимпульсных фазометрических систем. Результаты работы были внедрены на ряде предприятий военно-промышленного комплекса и защищены тремя авторскими свидетельствами. В 70-е гг. Борис Львович был руководителем и участником ряда проектов отраслевого значения по разработке фазометрических систем и устройств. Результаты работы отмечены двумя серебряными медалями ВДНХ. С 1981 г. переведен в ТПИ на кафедру «Оптимизация систем управления», где работал доцентом, профессором кафедры, руководил Западно-Сибирским региональным центром информационных и социальных технологий (с 1991 по 2008 гг.).

С 2008 г. назначен на должность начальника вновь созданного отдела информационных технологий высшей школы Института кибернетики Томского политехнического университета (ТПУ).

Б.Л. Агранович является вице-президентом Ассоциации инженерного образования России, ответственным секретарем журнала «Инженерное образование». С 1981 по 2003 гг. был зам. главного редактора межвузовского сборника «Кибернетика и вуз».

С начала 80-х гг. Борис Львович активно занимается исследованием проблем высшего профессионального образования, является руководителем крупных проектов по созданию автоматизированных систем управления в технике, образовании, обучении и экономике. Признанием научных результатов этих работ явилось назначение Б.Л. Аграновича в 1978 г. главным конструктором типовых автоматизированных систем управления вузом Минвуза РСФСР, а в 1994 г. — главным конструктором Интегрированной отраслевой автоматизированной

системы управления Минвуза РСФСР. Под его руководством была спроектирована и введена в действие Интегрированная Отраслевая автоматизированная система управления Минвуза РФ, системно и технологически сопряженная с типовыми автоматизированными комплексами управления вузом. Это положило начало новой культуре управления в отрасли, позволило значительно повысить оперативность и качество принимаемых решений.

Значительные результаты получены Б.Л. Аграновичем в области решения проблем университетского инженерного образования. Им совместно с учёными ведущих вузов страны сформирована концепция технического университета, разработаны методические материалы по формированию комплексной программы перестройки деятельности вуза как технического университета, созданы конструктивные методы системного проектирования содержания университетского образования. Эти работы послужили базой для формирования в России новой образовательной системы — технического университета. Борис Львович был одним из организаторов формирования системы университетского технического образования в России. В 2003–2004 гг. он был ответственным исполнителем межвузовских научно-исследовательских работ «Исследование проблем и разработка путей становления и развития инновационных университетов России» и «Исследование и разработка системы управления инновационного университета», выполняемых по инициативе Томского политехнического университета, Ассоциации инженерного образования и финансируемых Министерством образования РФ. В результате выполнения этих работ на основе системного анализа инновационной деятельности в зарубежных и отечественных университетах были разработаны концептуальные основы формирования инновационных университетов России, критерии инновационного университета, а также основные принципы трансформации традиционного университета в инновационный, миссия инновационного университета и его организационный дизайн. Важным результатом работы было исследование взаимодействия университета — властных структур и научно-образовательного сообщества (тройная спираль), на основе чего подготовлены проекты законодательных предложений по совершенствованию этого взаимодействия.

В 2005–2006 гг. Б.Л. Аграновичем, как заместителем научного руководителя программы «Томск — центр образования, науки и инноваций мирового уровня», проведена большая и плодотворная работа по диагностическому анализу возможностей и концепции развития города как центра образования, науки и инноваций мирового уровня, сформированы совместно с администрацией и учёными города цели, способы их достижения и планы мероприятий по реализации поставленных задач по 15 направлениям: образование, наука, здоровье населения и развитие системы здравоохранения, промышленное производство, строительство и архитектура и др., а также экономическое и финан-

совое обеспечение программы, совершенствование управления городом.

В 2007–2008 гг. Борис Львович был научным руководителем инновационной образовательной программы Томского политехнического университета «Опережающая подготовка элитных специалистов и команд профессионалов мирового уровня по приоритетным направлениям науки, технологий и техники». Реализация программы во многом способствовала становлению и развитию университета как вуза инновационного типа.

В преподавательской деятельности Б.Л. Агранович читал циклы лекций и подготовил учебно-методические пособия по курсам «Радиотехнические цепи и сигналы», «Прикладные методы системного анализа», «Компьютерные технологии в науке и образовании», «Системный анализ в экономике», «Мировые тенденции в высшем профессиональном образовании», «Инновационные образовательные технологии».

В 2002 г. Международная организация IGIP присвоила Б.Л. Аграновичу титул «Европейский преподаватель инженерного вуза».

Борис Львович с 2005 г. занимает должность профессора кафедры оптимизации систем управления, является действительным членом Академии информатизации образования (1996), член-корреспондентом Международной академии энерго-информационных наук (1995), автором и соавтором более 200 научных публикаций в отечественных и зарубежных источниках, в том числе 3 изобретений, 15 коллективных монографий.

Награжден медалью «За доблестный труд» (1970), медалью «За заслуги перед отечеством II ст.» (1996), удостоен звания «Заслуженный работник высшей школы» (2000), «Почетный работник высшего профессионального образования РФ» (2005), награжден серебряной медалью за заслуги перед Томским политехническим университетом (1998), юбилейной медалью «400 лет городу Томску» (2004), золотой медалью Петра I «За развитие инженерного дела и образования» (2011), почетными грамотами Минобрнауки РФ (2006), Государственной думы Томской области (2008), Администрации г. Томска (2008).

Борис Львович является лауреатом премий: Президента РФ в области образования (1998) «За разработку научных основ университетского инженерного образования и их реализацию в вузах России», Томской области в сфере науки и образования (1995), Томского обкома ВЛКСМ (1970) «За разработку и внедрение прецизионных приборов для фазовых измерений».

В настоящее время проф. Б.Л. Агранович проводит исследования по проблемам становления и развития профессиональной образовательной системы постиндустриального общества на основе smart-технологий, социального конструктивизма, использования сетевого интеллекта и облачных вычислений. По его мнению, каждая эпоха должна создать свой вид образования.

Пожелаем Борису Львовичу успехов в его новых начинаниях и долгих плодотворных лет жизни.

Summaries

UDC 519.2

Rozhkova S.V.
**THE RESEARCH OF THE EFFICIENCY
OF SIGNAL OPTIMAL TRANSMISSION**

The article considers the task of studying the efficiency of stochastic process optimal transmission through the continuous-discrete channels with memory and delay.

UDC 519.2

Rozhkova S.V.
**RECOGNITION OF STATES OF STOCHASTIC
SYSTEMS WITH FIXED MEMORY**

The article considers the task of recognizing the arbitrary number of hypotheses when the process observed is the process with continuous time; it represents the totality of processes with continuous and discrete time with fixed memory of random multiplicity.

UDC 517

Churikov V.A.
**THE DERIVATIVES AND INTEGRALS OF FRACTIONAL
COMPLEX ORDERS OF DISCRETE VARIABLE FUNCTION**

The article introduces the discrete d -operator of fractional integro-differentiation of complex orders. The algorithm of discrete differentiation and digital integration of discrete variable function is considered.

UDC 519.2

Ustinova I.G.
**SELECTION OF DISPERSION TREND OF RANDOM PROCESS
FOR POISSON CURRENT OF MEASURING MOMENTS**

The authors have obtained the estimations of explicit dispersion trend at Poisson current of measurement moments. Special cases of the trend: linear, square and in the form of first-order spline were considered. The statistical characteristics of the estimations obtained were studied. Their nonbias was shown; the covariance matrix of trend parameters estimations was determined.

UDC 519.25 (550.831.05)

Ustinova V.N., Ustinova I.G.
**ESTIMATION OF AUTOCORRELATION FUNCTION
IN THE FORM OF FIRST-ORDER SPLINE**

The authors have proposed the technique for selecting autocorrelation function trend in the form of first-order spline. By the example of model seismic trace and Bessel function it was shown that the technique proposed approximates better the autocorrelation function than the approximation obtained when using the standard formula for estimating the autocorrelation function.

UDC 519.874

Kitaeva A.V., Stepanova N.V.
OPTION RESOURCE CONTROL

The article considers the model with one resource type with limited life time supplied at the beginning of production cycle. The demand for the resource during production is of random nature: the

request stream forms steady process, the request volumes are independent equally distributed random variables with specified mean and dispersion. The asymptotic distributions of the total volume of demand for raw materials during the production cycle and time of resource use were determined. The authors have obtained the equations for statistical estimation of the required parameters and approximate value of optimal volume of option resource in the matter of maximum of the enterprise average profit.

UDC 519.6

Reyzlin V.I., Orlov V.A.
**INTERACTIVE PROCEDURE OF SOLVING VECTOR
OPTIMIZATION PROBLEMS WITH UNCERTAIN PARAMETERS**

The article considers the technique of solving multicriteria optimization problems with mathematical models containing variables; their values are not controlled by a decision-maker. The notion of alternate solution tolerance is introduced. The latter is related to stability, vitality.

UDC 519.6:004.93

Luneva E.E., Avramchuk V.S.
**THE ANALYSIS OF WAYS FOR INCREASING
THE EFFICIENCY WHEN CALCULATING
TIME-AND-FREQUENCY CORRELATION FUNCTION**

The article considers the techniques for increasing the efficiency of mathematical computing on multiprocessing systems using graphic processors NVIDIA CUDA. It was shown that the graphic processors exceed general-purpose ones in speed when solving the problems connected with computation. The efficiency of using the Microsoft.NET Framework technique depends totally on a number of computing kernels in the processor. CUDA technique application allows invoking the unique computational architecture of graphic processors and decreasing considerably the total calculation time.

UDC 004.492.2

Kukalo I.A., Kshnyankin A.P., Grivtsov S.N.
**THE MODEL OF PHYSICAL SECURITY THREATS TO LINEAR
PART OF OIL PIPELINE**

The paper indicates the feature of controlling security of such extended object of federal importance as linear part of oil pipeline. The hierarchical model of threats to typical linear part of oil pipeline has been proposed. The article introduces the daily estimation of probability of occurrence of threats to its physical security applying the technique based on statistical data for linear part of oil pipeline «TsentrSibnefteprovod». The groups of factors effecting the model as well as weight coefficients considering each factor contribution were determined. The authors made illustrative estimation of line distribution of probability physical threat occurrence for linear part of oil pipeline «Aleksandrovscoe-Anzhero-Sudzhensk».

UDC 519.688:622.279.23

Naymushin A.G., Sergeev V.L.
**IDENTIFICATION OF EVOLUTION PROCESSES OF A SYSTEM
LIFE CYCLE CONSIDERING A-PRIORI INFORMATION**

The article considers the problems of parametric identification of evolution processes. The integrated systems of phenomenological models are proposed to be used for solving the problems considering

a-priori information. The paper introduces the examples of solving the tasks of identification and prediction of accumulated production life cycle while oil deposit developing.

UDC 681.5

**Kurgankin V.V., Zamyatin S.V.,
Zamyatin V.M., Pushkarev M.I.
SYNTHESIS OF EMBEDDED SINGLE-CIRCUIT AUTOMATED
CONTROL SYSTEM WITH SELF-TUNING CONTROLLER
AND ITS ROBUSTNESS ESTIMATION**

The authors consider the possibility of applying the dynamic compensation principle for synthesis of embedded single-circuit automated control system with self-tuning controller. The paper introduces the results of full-scale experiments. The robustness properties of the synthesized system were analyzed.

UDC 622.333.03:556.3

**Odnokopylov I.G., Gneushev V.V.,
Sizikov D.A., Shishlyayev V.V.
INJECTION AUTOMATION AT HYDRODYNAMIC
RESEARCHES OF COAL SEAMS FILTRATION
CHARACTERISTICS**

The improvement of methods for researching the formations of porous fractured reservoir rocks with average and low permeability is the urgent task when selecting the techniques of development of coalbed methane reservoirs. The paper considers the engineering solutions of automation of fluid injection control at coal seams overbalance hydrodynamic researches (injection-falloff test).

UDC 004.352.242

**Marchenko V.V., Berestneva O.G.,
Devatykh D.V., Sukhanova E.F.
THE RESEARCH OF NONPARAMETRIC ESTIMATE
METHOD FOR THE PERIOD LENGTH
AND SIGNAL PERIODIC COMPONENT**

The paper considers the techniques of signal hidden periodic component. The nonparametric estimate method for the period length and periodic components has been selected as the principle and the most suitable one for being used in this work. This is the very technique which allows separating hidden components from any periodic signals but not only from harmonic ones. The technique was studied at model signals. The dependence of the technique relative inaccuracy on the number of hidden periods and sample interval was shown.

UDC 528.854.2

**Zamyatin A.V., Afanasyev A.A.
PARALLEL COMPUTING IN THE PROBLEM
OF LAND COVER CHANGES 3D MODELING**

The task of land cover changes modeling based on application of Markov chains and cellular automata apparatus is discussed. The increased performance of the task is achieved by application of different variants of parallel processing when modeling. The article introduces the results of numerical experiments on estimation of various types of parallel processing, obtained when using the expensive supercomputing cluster and cluster from low-cost PC in local network including the estimation of parallel speed up, parallel efficiency and cluster overall performance.

UDC 519.673

**Khamukhin A.A.
APPLICATION OF HOMOGENEOUS STRUCTURE CELL WHEN
DETECTING NOISE HYDROACOUSTIC SIGNALS BASED ON
INTEGRATED WAVELET SPECTRUM**

When detecting noise hydroacoustic signals based on integrated wavelet spectrum the computation of continuous wavelet transform takes the significant time that results in detection relay. The author has proposed the modified scheme of homogeneous computing structure

for computation paralleling and speed up. The article introduces the results of computer modeling of test task of detecting noise hydroacoustic signals based on integrated wavelet spectrum by means of homogeneous structure cells.

UDC 621.312/313: 621.317.31

**Bedareva E.V., Kostin E.V.
THE RESEARCH OF COAXIAL SHUNT
TEMPERATURE FIELD IN ANSYS**

The distribution of coaxial shunt temperature field has been analyzed in ANSYS. It has been ascertained that temperature field of coaxial shunt is of a nonuniform character owing to material heat capacity and natural convection. The impingement cooling system of the shunt is proposed to be used. It allows decreasing the internal mechanical stress of the resistance element and improving shunt metrological characteristics as the whole.

UDC 533.9.08

**Merkulov S.V., Golobokov Yu.N.,
Baystrukov K.I., Pavlov V.M., Kachkin A.G.,
Mezentsev A.A., Li A.M., Khokhryakov V.S.
THE SYSTEM FOR ACQUISITION AND LOGGING
THE RESULTS OF MEASURING MAGNETIC PARAMETERS
OF ELECTROPHYSICAL UNIT**

The authors have proposed the structure of the system for acquisition and logging the results of measuring magnetic parameters of KTM unit. The algorithms and protocols of measurement results transfer from measuring transducers into data logging units were developed. The units for data recording from measuring transducers were implemented on the basis of modules with programmable logic. The algorithms of acquisition, logging and primary processing of the results of electromagnetic parameters measurements were developed.

UDC 681.5

**Krivtsov P.Yu., Pavlov V.M.
THE DEVELOPMENT OF CONTROL SYSTEM
FOR SODIUM COOLANT PROCESSING**

The paper introduces the general algorithm of controlling sodium processing unit; the principle functions of the system and circuits for controlling process conditions depending on the unit current operation mode have been determined. The control system development is shown by the example of sodium supply system. The authors describe the software configuration and introduce the structure of processing displays of operator interface.

UDC 62-533.65

**Korovikov A.G., Pavlov V.M., Olkhovik D.A.
SOFTWARE MODULE OF SAFETY ALARM, EMERGENCY
PROTECTION AND AID TO THE OPERATOR
OF VACUUM SYSTEM OF TOKAMAK TKM**

The authors have studied the pre-vacuum pumping system of tokamak KTM vacuum unit on using the model implemented by Petri nets. The model qualitative characteristics have been determined. Based on the model data the module of safety alarm, emergency protection and aid to the operator was programmed.

UDC 681.5.015+004.428

**Filipas A.A., Mikolaenko S.N., Kladiev S.N.
SOFTWARE SYSTEM FOR SIMULATION OF INDUSTRIAL
MECHANISM LOAD TORQUE**

A software system for referencing the load torque of industrial mechanisms is developed under MATLAB Simulink environment; it has all the peculiarities of the base software package. Simulation models of the typical industrial mechanisms are grouped in a library and made of the standard program blocks. The models interfaces allow adjusting parameters of the industrial mechanism.

UDC 004.822:004.89

Zagorulko Yu.A.
THE CONCEPT OF KNOWLEDGE REPRESENTATION
INTEGRATED MODEL

The article considers the concept of integrated model which combine mutually complementary methods and knowledge presentation tools. On the basis of the model the language of knowledge representation and processing as well as the tools providing to the full extent the known needs for developing knowledge data base of the applied intelligence systems are set up.

UDC 004.942

Beshta A.A., Kirpo M.A.
CONSTRUCTION OF OBJECT TRUST MODEL
IN THE AUTOMATED INFORMATION SYSTEM
FOR PREVENTING DESTRUCTIVE INFLUENCE
ON THE SYSTEM

A formal model of destructive influence on the system has been described; the main possible destructive effects are singled out. The authors have proposed the object trust model of the automated information system on the basis of object operation monitoring. The paper introduces the service oriented concept of delivering information safety service and multiagent distributed architecture implementing the concept.

UDC 550.8.053

Ivanchenkov V.P., Kochegurov A.I., Cherkasova M.A.
THE ANALYSIS OF INFORMATION FEATURES OF MUTUAL
PHASE SPECTRA OF REFLECTED SEISMIC WAVES

Based on the accepted model of layered absorbing media the authors have considered the features of mutual phase spectra of seismic waves reflected from the top and the bottom of the depth; the main prerequisites of their application for forecasting geologic profile were determined.

UDC 004.056.5, 004.89

Atkina V.S.
THE ANALYSIS SYSTEM OF DISASTER TOLERANCE

The article describes the problem of developing disaster tolerant information systems and their importance for information security. The author has assumed that the current rates of disaster tolerant system should be analyzed and the efficiency of disaster tolerant solutions should be estimated for development and management of disaster tolerant information systems. The analysis system of disaster tolerance formed on the basis of artificial immune system was proposed and described formally. The possibility of applying the solution proposed as a tool for decision support was shown by the example of the results of carried out experiments.

UDC 004.931

Druki A.A.
RECOGNITION OF STRUCTURED SYMBOLS
ON IMAGES USING HISTOGRAM OF AVERAGE INTENSITY
AND CONVOLUTION NEURAL NETWORK

The algorithm of selecting a region of character pitch on a complex background and the algorithm of symbol selection on the basis of average intensity histogram have been developed and introduced. The convolution neural network was developed and introduced to solve the problem of symbol recognition.

UDC 004.021

Belousov A.A., Spitsyn V.G.
APPLICATION OF 3D METHODS
OF VIDEO IMAGE ENHANCEMENT

The authors have studied the applicability of image processing methods on the basis of Illuminance-Reflectance model and histogram equalization for video data processing. The indicated methods

were modified for being applied in real time mode. Their application allows improving considerably such frame criteria in video sequences as quantity and intensity of edge pixel and the level of brightness adaptation to human vision.

UDC 004.89:004.4

Rudometkina M.N., Spitsyn V.G., Ettel V.A.
DEVELOPMENT OF RELATIONAL MEANS
OF PREDICATE DECOMPOSITION

The article is devoted to the development of new means for predicate decomposition on the basis of statements on dependences from relational normalization. The precise definition to decomposition in respect to relationals and predicates has been given. The paper considers the statements on dependences which allow decomposing the relationals; introduces their generalizations. Based on the connection between the relation algebra and predicate algebra the authors have stated the statements on dependences converted into the new means of predicate decomposition.

UDC 004.415

Hoang Van Kuet, Tuzovsky A.F.
CONTROL METHOD OF DIRECT ACCESS
TO SEMANTIC DATA BASE

The paper introduces the main notions, methods for determining and developing security labels of semantic data triplets. The authors have developed the algorithms for forming the security cover of semantic data and safety control of user direct access to data bases.

UDC 004.912 519.689.3

Sidorova E.A.
THE DEVELOPMENT OF LINGUISTIC SUPPORT
OF INFORMATION SYSTEMS ON THE BASIS
OF KNOWLEDGE ONTOLOGICAL MODELS

The article describes knowledge ontological model focused on automatic analysis of texts in concrete subject area. The author considers the linguistic resources and tools required for developing the linguistic support of information systems.

UDC 004.415

Hoang Van Kuet, Tuzovsky A.F.
METHODS FOR DETERMINING SECURITY
LEVELS OF ONTOLOGY ELEMENTS

The paper considers the principle problems of multilevel security in semantic data. The principles of assigning the security levels of notions, properties and individuals of the ontology have been described. The authors have proposed the algorithms which allow defining the security levels for the main elements of ontology and inferences obtained in logical rules.

UDC 004.8; 004.4'2

Zaikin I.A.
THE METHOD OF OWL 2 ONTOLOGY THREE-WAY MERGE

The authors have proposed the algorithm of ontology three-way merge at multiuser editing by pair-wise comparison of ontologies based on detection of coinciding and conflicting changes. The algorithm software implementation was developed.

UDC 004.02:004.82

Le Hoai, Tuzovsky A.F.
DOCUMENT SEMANTIC ANNOTATION
IN DIGITAL LIBRARIES

The paper considers the task of document annotation in digital libraries applying semantic technologies. The reasons and advantages of such annotation application are described. The authors have proposed a new technique for semi-automatic annotation; the results of testing its software implementation are shown.

UDC 004.942, 004.652.5

Vichugova A.A., Vichugov V.N., Tsapko G.P.
FORMAL MODEL OF A STRUCTURE OF DIFFERENT DESIGN
OBJECTS RELATIONSHIPS

Within the development of the existing theoretical and practical states of information support techniques of product life cycle the authors have set the problem of developing the method which allows structuring the entities formed at high-technology product design by the example of radio electronics. The set-theoretic models are formed. The latter describe the composition and relationship of different design objects: product, its information models, electron structure and design documents. Using the object-oriented approach the authors have carried out the conceptual design of data base: a set of attributes describing life cycle of the design objects was determined and the formal information model of their relationship structure was proposed in the form of UML-class diagram.

UDC 681.3.06

Mytsko E.A., Malchukov A.N.
THE RESEARCH OF SOFTWARE IMPLEMENTATION
OF CRC COMPUTING ALGORITHMS COMPATIBLE
WITH PKZIP, WINRAR, ETHERNET

The paper introduces the numerical results of computer experiment for determining speed of software implementation of CRC32 computing algorithms. It was shown that the fast four byte matrix-driven algorithm should be applied in embedded systems and data transfer industrial systems.

UDC 004.62

Sharabayko M.P., Markov N.G.
THE EFFICIENCY OF BLOCK INTRA PREDICTION MODES
IN MODERN STANDARDS OF VIDEO COMPRESSION

In 2013 the final version of H.265/HEVC video compression standard is going to appear. Its implementation is supposed to start the mass shift to super resolution systems. The paper introduces the results of analysis of block intra prediction methods efficiency of H.265/HEVC standard from the point of view of their influence on key frame compression degree in digital video. Besides, the results of comparison of key frame compression efficiency in H.265/HEVC and H.264/AVC standards are given.

UDC 681.3.06

Mytsko E.A., Malchukov A.N.
THE RESEARCH OF HARDWARE IMPLEMENTATION
OF TABLE- AND MATRIX-DRIVEN ALGORITHMS
OF COMPUTING CRC32

The paper introduces the description of hardware implementation of matrix- and table-driven algorithms for computing CRC32 check sum on Altera programmed integrated logic circuits Cyclone of SDK-6.1 layout. The hardware implementation features are shown by the example of CRC32 valuator description; the designed units availability is demonstrated in a specific context. The algorithms have been studied by CRC32 valuator comparison in occupied logic cells and time delays.

UDC 004.9

Tarkov M.S., Osipov M.I.
ESTIMATION OF CELL NUMBER ON IMAGES
OF CYTOLOGICAL HERBAL PREPARATIONS

The authors have proposed the algorithm for calculating vacuoles (plant cells) on the image. The algorithm allows selecting cell outlines

in arrays of cells binding together by image brightness range partition into layers with further application half-tone erosion and dilatation, «hat top» transformation and liminal segmentation in each layer. When calculating cells the approximation of their outlines by ellipses is used. The algorithm may be used in various applications, for example when determining cell structure disintegration rate over a set of patterns.

UDC 519.175.1

Pogrebnoy V.K.
THE TASK OF ESTIMATING THE SIMILARITY OF TWO
GRAPHS STRUCTURES ON THE BASIS OF GENERAL PARTS
DEFINITION

The author has proposed the algorithm for solving the problem of estimating the similarity structures represented by ordinary graphs on the basis of their general parts definition in the form of isomorphic and partial graphs. The algorithm functions in the class of graphs with similar vectors of initial descriptors.

UDC 519.175.1

Pogrebnoy An.V., Pogrebnoy V.K.
INVARIANT OF A GRAPH BASED
ON DENSE SUBGRAPHS AND THE ALGORITHM
OF ITS COMPUTATION

The paper introduces the estimation of subgraph density for weighted ordinary graphs. On the basis of this estimation the authors have determined a number of invariants characterizing graph structure considering edge nonuniformity of weight values. Special attention is given to the notion of dense subgraph. The authors have proposed the algorithm of dense subgraph selection and computation of invariants on their base. The significant property of the dense subgraphs is the ability to reflect the isolation effect of vertex subset with high density estimation.

UDC 681.3.06(076.5)

Kositsyn V.S., Galchenko V.G., Gladkova T.A.
SOFTWARE AND DATAWARE FOR ANALYZING
MICROWAVE PULSE PARAMETERS

The estimation of microwave pulse parameters is the significant task in the experimental researches. The existing programs for plotting microwave pulse graphs and estimation of their parameters have been implemented in MathCad. The programs possess significant disadvantages as the majority of the parameters of the environment where microwave pulses travel should be entered immediately in the program. The paper describes the software of estimating microwave pulse parameters implemented in Qt 4.5 C++.

UDC 519.681.3

Anureev I.S.
ONTOLOGICAL TRANSITION SYSTEMS
AND THEIR APPLICATION TO COMPUTER
LANGUAGE SEMANTICS

Operational ontological approach to a formal specification of computer languages has been developed by the author as a methodology of forming operational semantics of computer language on the basis of its ontology, known as operational ontological semantics. The author proposes formalism for describing operational ontological semantics of computer languages, ontological transition systems; ontological description of computer languages is compared with the traditional grammatical one.

Сведения об авторах

Аврамчук Валерий Степанович, 1978 г.р., канд. техн. наук, доцент кафедры автоматики и компьютерных систем Института кибернетики ТПУ. Р.т. 41-89-07. E-mail: avs@tpu.ru. Область научных интересов: цифровая обработка сигналов, спектральный анализ, обнаружение утечек.

Ануреев Игорь Сергеевич, 1971 г.р., канд. физ.-мат. наук, ст. науч. сотр. лаб. теоретического программирования Института систем информатики имени А.П. Ершова СО РАН, г. Новосибирск. Р.т. 8-(383)-330-63-60. E-mail: anureev@iis.nsk.su. Область научных интересов: спецификация, верификация, семантика, онтологии, компьютерные языки, интеллектуальные системы, информационные системы, предметно-ориентированные языки.

Аткина Владлена Сергеевна, аспирант, старший преподаватель кафедры информационной безопасности Волгоградского государственного университета. Р.т. 8-(442)-46-03-68. E-mail: atkina.vladlena@yandex.ru. Область научных интересов: надежность информационных систем, системы искусственного интеллекта, системы поддержки принятия решений, информационная безопасность.

Афанасьев Антон Аркадьевич, 1988 г.р., аспирант кафедры оптимизации систем управления Института кибернетики ТПУ. Р.т. 70-57-02. E-mail: afanasyeva@tpu.ru. Область научных интересов: моделирование, геоинформационные системы, веб-технологии, параллельное программирование.

Байструков Константин Иванович, 1966 г.р., ст. преподаватель кафедры электроники и автоматики физических установок Физико-технического института ТПУ. Р.т. 42-64-63. E-mail: Baystrukov@tpu.ru. Область научных интересов: системы автоматизации физических экспериментов, методы и средства синхронизации процессов управления и измерения.

Бедарева Елена Вячеславовна, 1985 г.р., инженер, аспирант кафедры компьютерных измерительных систем и метрологии Института кибернетики ТПУ. Р.т. 41-75-27. E-mail: blv85@sibmail.com. Область научных интересов: программное обеспечение измерительных систем, математическое моделирование измерительных систем.

Белосов Артем Анатольевич, 1984 г.р., канд. техн. наук, доцент кафедры вычислительной техники Института кибернетики ТПУ. Р.т. 70-16-09. E-mail: artem.a.belosov@gmail.com. Область научных интересов: анализ изображений и видеопоследовательностей, системы искусственного интеллекта.

Бешта Александр Александрович, 1987 г.р., аспирант кафедры информационной безопасности Физико-технического института Волгоградского государственного университета. Р.т. 8-(844-2)-46-03-68. E-mail: abewta@rambler.ru. Область научных интересов: многоагентные системы, безопасность автоматизированных информационных систем.

Вичугов Владимир Николаевич, 1982 г.р., канд. техн. наук, доцент кафедры автоматики и компьютерных систем Института кибернетики ТПУ. Р.т. 41-89-07. E-mail: vlad@aics.ru. Область научных интересов: программирование, базы данных, автоматизированные системы управления.

Вичугова Анна Александровна, 1986 г.р., ассистент кафедры автоматики и компьютерных систем Института кибернетики ТПУ. Р.т. 41-89-07. E-mail: anya@aics.ru. Область научных интересов: бизнес-моделирование, структурный анализ, базы данных, информационные системы электронного документооборота, информационно-управляющие системы.

Гальченко Валерий Григорьевич, 1944 г.р., канд. техн. наук, доцент кафедры прикладной математики Института кибернетики ТПУ. Р.т. 42-61-00. E-mail: gal@am.tpu.ru. Область научных интересов: применение математических методов в системах управления.

Гладкова Татьяна Александровна, 1957 г.р., программист кафедры прикладной математики Института кибернетики ТПУ. Р.т. 42-61-00. E-mail: am@am.tpu.ru. Область научных интересов: программирование на языках высокого уровня.

Гнеушев Виталий Викторович, 1975 г.р., ведущий инженер представительства ОАО «Газпром промгаз», г. Новокузнецк. Р.т. 8-(384-3)-35-31-10. E-mail: dede1975@mail.ru. Область научных интересов: автоматизация технологических процессов.

Голобоков Юрий Николаевич, 1982 г.р., инженер информационного отдела Физико-технического института ТПУ. Р.т. 42-12-95. E-mail: golobokov@tpu.ru. Область научных интересов: системы автоматизации физических экспериментов, разработка программного обеспечения для систем сбора данных и управления.

Гривцов Сергей Николаевич, 1961 г.р., ведущий специалист отдела охраны магистральных нефтепроводов Службы безопасности ОАО «Центрсибнефтепровод», г. Томск. Р.т. 27-53-82. E-mail: GrivtsovSN@tom.transneft.ru. Область научных интересов: физическая безопасность магистральных нефтепроводов.

Друки Алексей Алексеевич, 1985 г.р., аспирант кафедры вычислительной техники Института кибернетики ТПУ. Р.т. 22-45-08. E-mail: druki2008@yandex.ru. Область научных интересов: информационные технологии, обработка изображений, нейронные сети.

Загоруйко Юрий Алексеевич, 1957 г.р., канд. техн. наук, зав. лабораторией Института систем информатики им. А.П. Ершова СО РАН, г. Новосибирск. Р.т. 8-(383)-332-83-59. E-mail: zagor@iis.nsk.su. Область научных интересов: интеллектуальные системы, представление знаний, разработка онтологий, компьютерная лингвистика, логическое программирование.

Заикин Иван Анатольевич, 1986 г.р., ассистент кафедры оптимизации систем управления, программист лаборатории когнитивных систем и управления знаниями Института кибернетики ТПУ. Р.т. 42-14-85. E-mail: zaikin@tpu.ru. Область научных интересов: семантические технологии, онтологическое моделирование, объектно-ориентированный анализ и проектирование.

Замятин Александр Владимирович, 1978 г.р., канд. техн. наук, проректор-директор Института кибернетики ТПУ. Р.т. 70-57-02. E-mail: zamyatin@tpu.ru. Область научных интересов: геоинформационные системы, анализ и об-

- работка данных, нейросетевые технологии, распознавание образов.
- Замятин Владимир Маркович**, 1951 г.р., канд. техн. наук, доцент кафедры теоретической и прикладной механики Института физики высоких технологий ТПУ. Р.т. 56-45-00. E-mail: zvm@tpu.ru. Область научных интересов: динамика, прочность машин, приборов и аппаратуры.
- Замятин Сергей Владимирович**, 1982 г.р., канд. техн. наук, доцент кафедры автоматизации и компьютерных систем Института кибернетики ТПУ. Р.т. 70-56-60. E-mail: zamsv@tpu.ru. Область научных интересов: робастные и самонастраивающиеся системы управления.
- Иванченко Виктор Павлович**, 1939 г.р., канд. техн. наук, доцент кафедры прикладной математики Института кибернетики ТПУ. Р.т. 42-61-00. E-mail: am@am.tpu.edu.ru. Область научных интересов: цифровая обработка сигналов и изображений волновых полей.
- Качкин Александр Георгиевич**, 1984 г.р., инженер кафедры электроники и автоматизации физических установок Физико-технического института ТПУ. Р.т. 42-12-95. E-mail: rumble@tomics.tomsk.ru. Область научных интересов: системы автоматизации физических экспериментов, системы цифрового управления силовыми полупроводниковыми преобразователями.
- Кирпо Максим Андреевич**, 1986 г.р. аспирант кафедры информационной безопасности Физико-технического института Волгоградского государственного университета. Р.т. 8-(844-2)-46-03-68. E-mail: infsec@volsu.ru. Область научных интересов: обеспечение безопасности автоматизированных систем управления технологическими процессами.
- Китаева Анна Владимировна**, 1957 г.р., д-р физ.-мат. наук, доцент кафедры инженерного предпринимательства Института социально-гуманитарных технологий ТПУ. Р.т. 56-35-29. E-mail: kit1157@yandex.ru. Область научных интересов: непараметрическое и робастное оценивание характеристик случайных процессов, принятие решений в условиях неопределенности, математическое моделирование социальных и экономических систем.
- Кладиев Сергей Николаевич**, 1960 г.р., канд. техн. наук, доцент кафедры электропривода и электрооборудования Энергетического института ТПУ. Р.т. 56-37-59. E-mail: kladiev@tpu.ru. Область научных интересов: автоматизация технологических процессов и производств, автоматизированный электропривод переменного тока общепромышленных механизмов.
- Коровиков Александр Генадьевич**, 1979 г.р., начальник отдела информационных технологий Республиканского государственного предприятия «Национальный ядерный центр», г. Курчатов, Республика Казахстан; аспирант кафедры электроники и автоматизации физических установок Физико-технического института ТПУ. Р.т. 8-(722-51)-2-35-49. E-mail: Korovikov@nnc.kz. Область научных интересов: системы автоматического управления технологическими процессами.
- Косицын Виктор Сергеевич**, 1945 г.р., науч. сотр. лаборатории № 42 Физико-технического института ТПУ. Р.т. 42-38-28. E-mail: coss@tpu.ru. Область научных интересов: диагностика параметров ускорителей.
- Костин Евгений Владимирович**, 1981 г.р., программист лаборатории № 5 отдела информатизации образования ТПУ. Р.т. 42-13-70. E-mail: kostin@tpu.ru. Область научных интересов: силовая электроника, физика полупроводниковых приборов, информатика.
- Кочегуров Александр Иванович**, 1954 г.р., канд. техн. наук, доцент кафедры прикладной математики Института кибернетики ТПУ. Р.т. 42-04-63. E-mail: kai@cc.tpu.edu.ru. Область научных интересов: цифровая обработка сигналов и изображений волновых полей.
- Кривцов Павел Юрьевич**, 1978 г.р., зам. начальника отдела информационных технологий РГП «Национальный ядерный центр Республики Казахстан», г. Курчатов; аспирант кафедры электроники и автоматизации физических установок Физико-технического института ТПУ. Р.т. 8-(722-51)-2-24-21. E-mail: Krivtsov@nnc.kz. Область научных интересов: разработка и исследование систем управления сложными инженерными объектами.
- Кукало Иван Анатольевич**, 1988 г.р., аспирант кафедры радиоэлектроники и защиты информации Томского государственного университета систем управления и радиоэлектроники. Р.т. 50-88-27. E-mail: i@kukalo.ru. Область научных интересов: угрозы физической безопасности, системы управления рисками.
- Курганкин Виктор Витальевич**, 1986 г.р., аспирант кафедры теоретической и прикладной механики Института физики высоких технологий ТПУ. Р.т. 56-45-00. E-mail: kurgankin@tpu.ru. Область научных интересов: системы управления с переменной динамикой.
- Кшнянкин Александр Петрович**, 1950 г.р., доцент кафедры радиоэлектроники и защиты информации Томского государственного университета систем управления и радиоэлектроники. Р.т. 41-33-65. E-mail: ksh@post.tomica.ru. Область научных интересов: теоретические основы информационной безопасности, управление рисками информационной безопасности.
- Ле Хоай**, 1984 г.р., аспирант кафедры оптимизации систем управления института кибернетики ТПУ. Р.т. 42-14-85. E-mail: lehotomsk@yahoo.com. Область научных интересов: искусственный интеллект, информационный поиск, семантические технологии.
- Ли Алексей Михайлович**, 1986 г.р., аспирант кафедры электроники и автоматизации физических установок Физико-технического института ТПУ. Р.т. 42-12-95. E-mail: alee@tpu.ru. Область научных интересов: системы автоматизации физических экспериментов, разработка программного обеспечения систем управления быстропротекающими процессами.
- Лунова Елена Евгеньевна**, 1981 г.р., канд. техн. наук, ассистент кафедры автоматизации и компьютерных систем Института кибернетики ТПУ. Р.т. 41-89-07. E-mail: lee@tpu.ru. Область научных интересов: разработка программного обеспечения, цифровая обработка сигналов.
- Мальчуков Андрей Николаевич**, 1982 г.р., канд. техн. наук, доцент кафедры вычислительной техники Института кибернетики ТПУ. Р.т. 70-16-09. E-mail: jgs@tpu.ru. Область научных интересов: помехоустойчивое кодирование, полиномиальные коды, системы проектирования помехоустойчивых полиномиальных кодов, алгоритмы поиска образующих полиномов, быстродействующие алгоритмы кодирования и декодирования данных полиномиальными кодами.
- Марков Николай Григорьевич**, д-р техн. наук, профессор, зав. кафедрой вычислительной техники института кибернетики ТПУ. Р.т. 61-20-71. E-mail: markovng@vostokgazprom.ru. Область научных интересов: геоинформационные системы и технологии, автоматизация производства в нефтегазовой отрасли.

- Мезенцев Антон Алексеевич**, 1984 г.р., инженер кафедры электроники и автоматики физических установок Физико-технического института ТПУ. Р.т. 42-12-95. E-mail: mezentsev@tpu.ru. Область научных интересов: системы автоматизации физических экспериментов, разработка программного обеспечения для пультов управления экспериментальными физическими установками.
- Меркулов Степан Вадимович**, 1977 г.р., инженер-проектировщик кафедры электроники и автоматики физических установок Физико-технического института ТПУ. Р.т. 42-12-95. E-mail: stepan@tpu.ru. Область научных интересов: системы автоматизации установок управляемого термоядерного синтеза проектированием электронно-измерительной аппаратуры.
- Николаенко Сергей Николаевич**, 1988 г.р., студент кафедры «Электроника и автоматика физических установок» факультета технологий и автоматизации атомной промышленности Северского технологического института Национального исследовательского ядерного университета «МИФИ». Р.т. 8-(382-3)-78-02-97. E-mail: aelker@mail.ru. Область научных интересов: автоматизированный электропривод переменного тока.
- Мышко Евгений Алексеевич**, 1991 г.р., студент кафедры вычислительной техники Института кибернетики ТПУ. Р.т. 64-68-84. E-mail: Evgenrus70@mail.ru. Область научных интересов: программная и аппаратная реализация алгоритмов вычисления контрольной суммы, тестирование и сравнение по быстродействию алгоритмов вычисления CRC.
- Наймушин Антон Георгиевич**, 1985 г.р., аспирант кафедры геологии и разработки нефтяных месторождений Института природных ресурсов ТПУ. Р.т. 42-13-53. E-mail: nai-a@mail.ru. Область научных интересов: мониторинг разработки месторождений углеводородов.
- Однокопылов Иван Георгиевич**, 1983 г.р., канд. техн. наук, доцент кафедры электропривода и электрооборудования Энергетического института ТПУ. Р.т. 56-32-55. E-mail: odivan@yandex.ru. Область научных интересов: автоматизация технологических процессов на основе частотного регулирования асинхронным электроприводом.
- Ольховик Дмитрий Анатольевич**, 1983 г.р., начальник группы информационно-измерительных систем отдела информационных технологий Республиканского государственного предприятия «Национальный ядерный центр», г. Курчатов, Республика Казахстан. Р.т. 8-(722-51)-2-35-49. E-mail: Olkhovik@nnc.kz. Область научных интересов: системы автоматического управления технологическими процессами.
- Орлов Владимир Анатольевич**, 1953–2008 гг., директор ООО «Оптимальное решение», ст. преподаватель кафедры информатики и проектирования систем Института кибернетики ТПУ. Область научных интересов: моделирование, численные методы и абстрактная алгебра.
- Осипов Михаил Игоревич**, 1991 г.р., магистрант Новосибирского государственного университета. E-mail: inq1201@mail.ru. Область научных интересов: обработка изображений.
- Павлов Вадим Михайлович**, 1965 г.р., канд. техн. наук, доцент кафедры электроники и автоматики физических установок Физико-технического института ТПУ. Р.т. 42-12-95. E-mail: pavlov@tpu.ru. Область научных интересов: разработка методов и средств повышения информационной эффективности систем автоматизации научных исследований для экспериментальных физических установок.
- Погребной Владимир Кириллович**, 1942 г.р., д-р техн. наук, профессор кафедры информатики и проектирования систем Института кибернетики ТПУ. Р.т. 42-05-09. E-mail: vkr@tpu.ru. Область научных интересов: дискретная математика, математическое программирование, моделирование, автоматизированное проектирование распределенных систем реального времени.
- Пушкарёв Максим Иванович**, 1987 г.р., ассистент кафедры автоматики и компьютерных систем Института кибернетики ТПУ. Р.т. 41-89-07. E-mail: pushkarev@tpu.ru. Область научных интересов: робастные и самонастраивающиеся системы управления.
- Рейзлин Валерий Израилевич**, 1948 г.р., канд. физ.-мат. наук, доцент кафедры информатики и проектирования систем Института кибернетики ТПУ. Р.т. 42-63-34. E-mail: vir@tpu.ru. Область научных интересов: моделирование, численные методы и методы оптимизации, теория гравитации, ядерная физика.
- Рожкова Светлана Владимировна**, д-р физ.-мат. наук, профессор кафедры высшей математики Физико-технического института Томского политехнического университета. Р.т. 56-37-29. E-mail: rozhkova@tpu.ru. Область научных интересов: фильтрация, интерполяция, экстраполяция, распознавание стохастических процессов, передача стохастических сигналов по непрерывно-дискретным каналам с памятью; обнаружение аномальных помех; математическая теория финансового анализа.
- Рудометкина Моника Николаевна**, аспирант кафедры вычислительной техники Института кибернетики ТПУ. E-mail: monika_7272@mail.ru. Р.т. 8-(721-2)-21-00-37. Область научных интересов: моделирование логических сетей в системах искусственного интеллекта.
- Сергеев Виктор Леонидович**, 1947 г.р., д-р техн. наук, профессор кафедры геологии и разработки нефтяных месторождений Института геологии и нефтегазового дела ТПУ. Р.т. 42-13-53. E-mail: SergeevVL@ignd.tpu.ru. Область научных интересов: мониторинг разработки месторождений углеводородов, системный анализ, идентификация и управление.
- Сидорова Елена Анатольевна**, 1977 г.р., канд. физ.-мат. наук, науч. сотр. Института систем информатики им. А.П. Ершова СО РАН, г. Новосибирск. Р.т. 8-(383)-330-79-91. E-mail: lena@iis.nsk.su. Область научных интересов: компьютерная лингвистика, автоматическая обработка текста, построение словарей, интеллектуальные системы, представление знаний, разработка онтологий.
- Сизиков Дмитрий Александрович**, 1968 г.р., заведующий лабораторией ОАО «Газпром промгаз», г. Москва. Р.т. 8-(495)-817-00-58. E-mail: d.sizikov@yandex.ru. Область научных интересов: гидрогеологические параметры и геолого-промысловые характеристики метанугольных месторождений.
- Спицын Владимир Григорьевич**, 1948 г.р., д-р техн. наук, профессор кафедры вычислительной техники Института кибернетики ТПУ. Р.т. 70-17-77 (вн. 2231). E-mail: spvg@tpu.ru. Область научных интересов: обработка изображений в технических системах, системы искусственного интеллекта, распознавание образов, искусственные нейронные сети, генетические алгоритмы, рассеяние электромагнитных волн.
- Степанова Наталья Викторовна**, 1988 г.р., аспирант кафедры инженерного предпринимательства Института социаль-

- но-гуманитарных технологий ТПУ. Р.т. 56-35-29. E-mail: natalia0410@rambler.ru. Область научных интересов: системный анализ, теория управления, экономико-математическое моделирование.
- Тарков Михаил Сергеевич**, 1955 г.р., канд. техн. наук, доцент, ст. науч. сотр. лаборатории физических основ интегральной микроэлектроники Института физики полупроводников СО РАН, г. Новосибирск. Р.т. 8-(383)-330-84-94. E-mail: tarkov@isp.nsc.ru. Область научных интересов: параллельные вычислительные системы, нейронные сети, обработка изображений.
- Тузовский Анатолий Федорович**, 1952 г.р., д-р техн. наук, профессор кафедры оптимизации систем управления Института кибернетики ТПУ. Р.т. 42-14-85. E-mail: tuzovskyaf@tpu.ru. Область научных интересов: семантические технологии, онтологии, интернет технологии, технологии разработки программ.
- Устинова Вера Николаевна**, д-р геол.-минерал. наук, профессор кафедры геофизики Института природных ресурсов ТПУ. Р.т. 42-61-72. E-mail: ustinovavn@tpu.ru. Область научных интересов: обработка и интерпретация материалов сейсморазведки, математические методы моделирования геофизических полей и геолого-геофизических объектов, фильтрация полей.
- Устинова Ирина Георгиевна**, канд. техн. наук, доцент кафедры высшей математики Физико-технического института ТПУ. Р.т. 56-35-93. E-mail: igu@sibmail.com. Область научных интересов: математические методы моделирования процессов и явлений, фильтрация геофизических полей.
- Филипас Александр Александрович**, 1964 г.р., канд. техн. наук, доцент кафедры «Электроника и автоматика физических установок» факультета технологий и автоматизации атомной промышленности Северского технологического института Национального исследовательского ядерного университета «МИФИ». Р.т. 8-(382-3)-78-02-97. E-mail: alf@ssti.ru. Область научных интересов: автоматизация технологических процессов и производств, автоматизированный электропривод промышленных механизмов.
- Хамухин Александр Анатольевич**, 1954 г.р., канд. техн. наук, доцент кафедры информатики и проектирования систем Института кибернетики ТПУ. Р.т. 42-04-05. E-mail: aaxtru@tpu.ru. Область научных интересов: исследование и разработка компьютерных моделей и методов моделирования.
- Хоанг Ван Куэт**, 1985 г.р., аспирант кафедры оптимизации систем управления Института кибернетики ТПУ. Р.т. 42-14-85. E-mail: student8050@sibmail.com. Область научных интересов: исследование методов поддержки работы с семантическими базами знаний и их эффективности.
- Хохряков Василий Сергеевич**, 1989 г.р., студент кафедры электроники и автоматики физических установок Физико-технического института ТПУ. Р.т. 42-12-95. E-mail: vasilmt@sibmail.com. Область научных интересов: системы автоматизации физических экспериментов, разработка программного обеспечения обработки данных электромагнитных измерений.
- Цапко Геннадий Павлович**, 1949 г.р., д-р техн. наук, профессор, заведующий кафедрой автоматики и компьютерных систем Института кибернетики ТПУ. Р.т. 41-89-07. E-mail: tsapko@aics.ru. Область научных интересов: информационные системы и технологии в космической отрасли промышленности, жизненный цикл изделий.
- Черкасова Маргарита Андреевна**, 1989 г.р., специалист ЗАО «Гринатом», г. Северск. Р.т. 8-(382-3)-52-85-10. E-mail: sunpingvin@mail.ru. Область научных интересов: цифровая обработка и интерпретация геофизической информации.
- Чуриков Виктор Анатольевич**, 1960 г.р., канд. физ.-мат. наук, доцент кафедры высшей математики Физико-технического института ТПУ. Р.т. 56-35-93. E-mail: vachurikov@list.ru. Область научных интересов: дробный анализ, математические модели в психологии, рентгеновская и нейтронная оптика, гамма-лазеры, квантовая теория поля.
- Шарабайко Максим Павлович**, 1988 г.р., аспирант кафедры вычислительной техники Института кибернетики ТПУ. Р.т. 41-91-49. E-mail: sme_box@tpu.ru. Область научных интересов: сжатие данных, видеокompрессия.
- Шишляев Виктор Владимирович**, 1986 г.р., заместитель заведующего лабораторией ОАО «Газпром промгаз», г. Москва. Р.т. 8-(495)-817-00-58. E-mail: shishlyaev@yandex.ru. Область научных интересов: гидродинамические методы исследования скважин, технологии моделирования разработки метаноугольных месторождений.
- Эттель Владимир Абрамович**, 1962 г.р., канд. техн. наук, доцент, профессор кафедры систем автоматизированного проектирования Карагандинского государственного технического университета. Р.т. 8-(721-2)-56-75-98 (вн. 1028). E-mail: ettel62@mail.ru. Область научных интересов: исследование и разработка информационно-образовательной среды учреждений образования.

К СВЕДЕНИЮ АВТОРОВ

Принимаются статьи, подготовленные в MS Word-2003 (файл и распечатка). Статья должна быть подписана авторами и иметь сопроводительное письмо на бланке организации.

Объем статьи до 8 стр., включая рисунки и таблицы, размещенные в тексте по упоминанию. Размер бумаги А4, поля по 25 мм. Текст в 1 интервал **без переносов, лишних пробелов и абзацных интервалов**, шрифт Times New Roman, 12 пунктов. Файлы рисунков (в градациях серого) в jpg, tif, cdr или иных форматах редакторов Photoshop, Corel Draw с разрешением 300 dpi прилагаются к статье. Рисунки и таблицы: **Рис. 1. Название; Таблица. Название.** Кавычки вида «...». Интервалы – 1,2...1,8 мм или 5–7 шт. Формулы – в MathType, настройка по умолчанию. Нумеруются только те формулы, на которые есть ссылка в тексте.

Курсивом – буквы латинского алфавита, кроме входящих в имена собственные, обозначения стандартных математических функций и химических элементов ($U_{пр.}$, Φ_i , но Al_2O_3 , $\cos\alpha_i$, $\max$, $\lg$, «BASF»). Векторы – полужирным курсивом. Список литературы – по ГОСТ Р 7.0.5-2008 (см. пример). Литература – по упоминанию: [1, 2], [2. С. 245], [3–7].

УДК 621.37 (Пример оформления статьи)

АНАЛИЗ РАБОТЫ СИСТЕМЫ АВТОМАТИЧЕСКОЙ РЕГУЛИРОВКИ

И.И. Иванов, П.П. Петров*

Томский политехнический университет

*ОАО «Центр», г. Москва

E-mail: ivanov@tpu.ru

Показана возможность расчета ... Установлено, что ... Сделан вывод о том, что ... (Аннотация, 10 кегль).

Ключевые слова (ниже ключевые слова на английском языке):

Усилительный каскад, регулировка тока.

В [1, 2] показано, что усилительный каскад с автоматической регулировкой потребляемого тока (АРИПТ) позволяет получить ...

СПИСОК ЛИТЕРАТУРЫ

1. Фамилия И.О. Название книги. – М.: Издательство, 2013. – 123 с.
2. Название книги / под ред. И.О. Фамилия. – М.: Издательство, 2013. – 123 с.
3. Фамилия И.О. Название статьи // Журнал. – 2013. – Т. 316. – № 1. – С. 71–77.
4. Фамилия И.О. Название диссертации: автореф. дис. ... канд. физ.-мат. наук. – Томск, 2008. – 19 с.
5. Название изобретения: пат. 2000000 Рос. Федерация. № 2009129009/10; заявл. 27.07.13; опубл. 10.10.13, Бюл. № 4. – 3 с.
6. Фамилия И.О. Название статьи // Наименование конференции: Труды VII Междунар. научно-практ. конф. молодых ученых. – Томск, 2013. – Т. 1. – С. 226–228.
7. Фамилия И.О. Название статьи // Наименование ресурса. 2013. URL: <http://www.tpu.ru/html/izvestia.htm> (дата обращения: 25.09.2013).

Поступила 25.01.2013 г.

Сведения об авторах:

Иванов Иван Иванович, 1975 г.р., канд. техн. наук, ст. науч. сотр. кафедры автоматики и компьютерных систем Института кибернетики ТПУ. Р.т. 22-22-22. E-mail: ivanov@tpu.ru. Область научных интересов: анализ...

Редактирование и корректура *А.С. Глазырин*

Компьютерная верстка *О.Ю. Аршинова*

Перевод на англ. язык *С.В. Жаркова*

Подписано к печати 28.05.2013. Формат 60х84/8. Бумага «Снегурочка».

Печать XEROX. Усл. печ. л. 26,52. Уч.-изд. л. 23,99.

Заказ 587-13. Тираж 500 экз.



Национальный исследовательский Томский политехнический университет
Система менеджмента качества
Издательства Томского политехнического университета сертифицирована
NATIONAL QUALITY ASSURANCE по стандарту BS EN ISO 9001:2008



ИЗДАТЕЛЬСТВО  ТПУ. 634050, г. Томск, пр. Ленина, 30. Тел./факс: 8(3822) 563-291, www.tpu.ru, izv@tpu.ru